IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 June 2024, accepted 1 July 2024, date of publication 8 July 2024, date of current version 17 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3424531

RESEARCH ARTICLE

Unveiling Deception in Arabic: Optimization of Deceptive Text Detection Across Formal and Informal Genres

FATIMAH ALHAYAN[1], HANEN T. HIMDI[2], AND BASMA ALHARBI[2], (Member, IEEE)
[1]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[2]Computer Science and Artificial Intelligence Department, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia

Corresponding author: Fatimah Alhayan (fnalhayan@pnu.edu.sa)

This work was supported by Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia, through the Princess Nourah Bint Abdulrahman University Researchers Supporting Project, under Grant PNURSP2024R719.

ABSTRACT In recent years, social media has significantly influenced how we share information and exchange messages. However, a significant issue arises from the fast dissemination of deceptive information portrayed as legitimate, which may seriously affect both people and society. Identifying unmonitored 'deceptive text' has become a crucial concern in mainstream media due to its potentially damaging impact. Although there have been recent studies that have developed AI models capable of identifying deceptive text in other languages, there is a scarcity of research focused on detecting detective text specifically in the Arabic language. This paper presents a novel Arabic deceptive text detection dataset constructed from publicly available resources. The dataset offers a unique distinction between formal and informal text genres, reflecting the diverse communication styles encountered in real-world deceptive language. We evaluate the performance of various machine learning (ML), deep learning (DL), and transformer-based models on this dataset for classifying text as deceptive or non-deceptive. The study investigates the impact of incorporating additional textual features including morphological features, psycholinguistic features, and sociolinguistic features alongside the raw text data. Our findings demonstrate that the AraBERTv2 model, after fine-tuning the Arabic dataset and incorporating textual features, achieves the best classification performance. This research contributes a valuable resource for Arabic deceptive text analysis and highlights the effectiveness of fine-tuned AraBERTv2 models with enriched features for such tasks.

INDEX TERMS Deceptive Arabic text, machine learning, deep learning, transformer, Arabic text classification.

I. INTRODUCTION

The proliferation of online platforms introduces a challenge stemming from diverse content quality. Put simply, the term ''false information'' is commonly employed to describe low-quality content [1], which encompasses any written or audio-visual content with the potential to deceive, confuse, or misinform individuals making online decisions [2]. Deception, as a concept, refers to the deliberate communication of messages and information with the intent to generate a false conclusion [3]. In textual context, deception involves presenting information in a way that conceals or distorts the truth by using carefully crafted language and persuasive techniques, leading readers to form incorrect interpretations or conclusions. Deceptive texts can take various forms, such as fake news, fake reviews, review spam, or phishing emails. Fake news involves a news article, like a report, editorial, or expose, intentionally being deceptive [4]. Deceptive reviews on products or services are intentionally crafted to seem real, aiming to benefit businesses by boosting their finances and reputation. These reviews come in different styles, languages, content, and lengths, making it challenging for people to spot deceptive reviews on their own [5]. Review spam includes reviews that go from merely irritating

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

self-promotions or unrelated announcements to purposely dishonest product reviews aiming to deceive consumers [6]. Additionally, phishing emails aim to deceive or trick the recipient into sharing sensitive information. Various methods are used to persuade email recipients, like clicking a link to a fake website, providing information directly, or downloading an email attachment that contains harmful software [7].

Studies suggest that humans can only detect deception with an accuracy rate of 54%, emphasizing the need for machine learning to automatically classify texts [8]. The detection of deceptive text has been explored in various forms, including news, reviews, and spam. Numerous studies have focused on text analysis for detecting deceptive text in different languages. In contrast to previous research that predominantly focused on identifying deceptive text in Arabic within specific genres, this study adopts a more comprehensive approach. We aim to thoroughly explore deceptive text across genres, including news articles, tweets, messages, reviews, and emails. The aim of this research is to develop innovative AI solutions capable of effectively detecting deceptive Arabic text in any given genre. Building on the foundation laid by prior studies, which highlighted the significance of deceptive cues identified as textual features for constructing machine learning models that detect deceptive text [9], [10], our research extends this exploration to modern learning techniques, such as transformers. We also specifically focus on evaluating the effectiveness of deceptive textual characteristics identified in previous research and assess their impact when integrated into different models trained to identify deceptive text across genres. The proposed research holds substantial significance on multiple levels: it addresses the urgent challenge of combating deceptive Arabic text while contributing to the advancement of Arabic language processing. By comprehensively addressing deceptive Arabic text across diverse genres, this research has the potential to significantly improve the quality of online Arabic content and enhance user safety.

This study's contributions are:

- Develop an Arabic deception detection model capable of identifying deception from diverse text genres, including formal genres like news articles and informal ones like tweets and reviews.
- Comprehensive comparison and evaluation of various models, including Machine Learning (ML), Deep Learning (DL), and transformer models, utilizing distinct features for enhanced Arabic deception detection.
- Introduction of the first Arabic phishing email dataset, translated from real English phishing emails for comprehensive analysis.
- Provide a thorough analysis of the challenges associated with detecting deceptive text in Arabic.

The rest of this study is organized as follows. Section II provides a comprehensive summary of related work. Section III provides an overview of the problem description. Section IV describes the employed methodology in details. Sections V, VI and VII present the experimental results, error analysis, and discussions and limitations. Finally, the concluding remarks are presented in Section VIII.

II. RELATED WORK

In the realm of identifying deceptive opinions, particularly false reviews in Arabic, the study in [11] extends beyond the use of lexical features (e.g., emotionalism, reflexivity, number of positive words, etc). It introduces innovative semantic features inspired by the analysis of discourse parse and rhetoric relations in Arabic. This involves acknowledging the significance of phrase units in the Arabic language and grammatical studies, which leads to the selection of commonly used unit markers and relations for computing the proposed features. These features, combined with lexical features obtained from a translated dataset (English to Arabic). The study utilizes a semi-supervised Support Vector Machine (SVM) for the classification of reviews, yielding noteworthy results with an 86% and 93% F-measure and accuracy, respectively. However, only classical ML classifiers are used for classification, as well as the reviews dataset used was translated from English.

A comprehensive overview of the Arabic AI Evaluation (ArAIEval) shared task, which is part of the first ArabicNLP 2023 conference held in conjunction with the Empirical Methods in Natural Language Processing Conference, conducted by [12]. Two task's primary objectives are identified. The first task is to detect persuasion techniques in diverse content, including tweets and paragraphs from news articles. This comprises two sub-tasks: binary classification to determine if a text snippet contains any persuasion technique, and multi-label classification to identify the propaganda techniques present. The dataset utilized for the task includes 156 tweets sourced from Arabic news accounts on Twitter and news articles from the AraFacts dataset [13], which features claims verified by Arabic fact-checking websites.

The other task targeted tweets and was organized into two sub-tasks as well, The first is to identify whether a tweet is disinformation or not, and multiclass classification to classify the disinformation tweets as hate-speech, offensive, rumor, or spam. The dataset used for this task includes 20K tweets related to COVID-19.

Notably, the majority of participating teams' proposed models for both tasks relied on fine-tuning transformer models like AraBERT and MARBERT. The highest Micro F1 for the binary classification of the first primary task was achieved at. 76 by [12] and at. 90 for the multi-label classification task in the second primary task by [14].

The team [12] took the first place for the first primary task which is to identify a multi-genre (a tweet or news paragraph) and whether it contains persuasive content. The study adjusted the MARBERT model through multitasking, comprising a primary binary classification task to detect persuasive techniques in text overall, and an additional task concentrating on categorizing texts by type (tweet or news). The additional task aimed to enhance the primary task by enabling the learning of distinct lexical and syntactic features

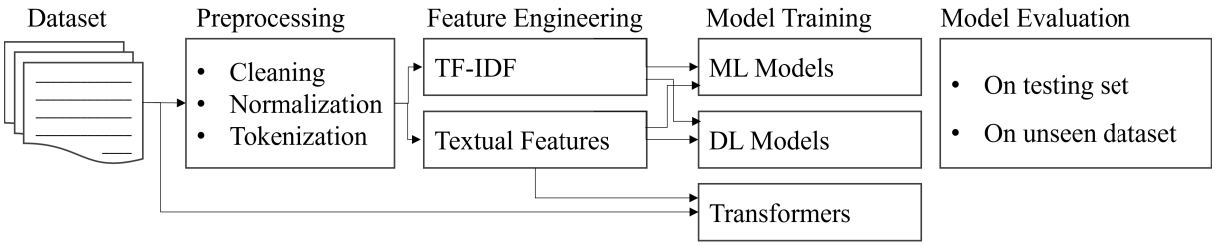

FIGURE 1. Arabic text deception detection framework, where ML stands for machine learning models, and DL stands for deep learning models.

related to persuasive content in tweets or news. Due to the imbalanced dataset, focal loss was utilized to optimize both tasks. The proposed model achieved the top rank on the leaderboard during the evaluation of the test set.

The other team [14] applied extensive preprocessing to tackle issues such as code-switching and the inclusion of emojis in tweets. The segments of the tweets that were not in Arabic were translated automatically into Arabic. Instead of eliminating emojis and hashtags, they were transformed into descriptive Arabic text to maintain the sentiment expressed in the tweets. The team explores large language models, particularly AraBERTCovid19 [15] with fine-tuned hyperparameters to address the computational expense of large models (AraBERT), as well as explores a soft-voting ensemble for the binary classification task (misinformative or not) and multi-class classification task (identify specific types of disinformation within a tweet encompassing hate speech, offensive language, rumors, and spam) disinformation classification tasks. As a result, the team reports a successful integration of meticulous preprocessing and hyperparameter-optimized AraBERT models, resulting in a first-place performance in both binary and multiclass disinformation classification tasks. For the binary task AraBERT model slightly outperformed the ensemble model, whereas the ensemble model outperformed the multi-classification model. This may be explained by the fact that multi-class problems are inherently complicated and need to better capture more subtle relationships in the data.

The study by [16] manually annotated a substantial dataset comprising 40,000 tweets in Arabic, encompassing both deleted and non-deleted tweets categorized into fine-grained disinformative classes, including hate speech, offensive, rumor, and spam. The study aimed to develop classification models for predicting the likelihood of a tweet being deleted before posting and identifying potential reasons for deletion.

The proposed models consisted of a binary classification model to distinguish deleted from non-deleted tweets, another binary model for classifying disinformative versus non-disinformative tweets, and a multiclass model for fine-grained classification of disinformation labels. The approach incorporated both classical algorithms, such as Random Forest (RF) and Support Vector Machines (SVM), and DL algorithms utilizing pre-trained Transformer models, specifically AraBERT and XLM-R.

The experimentation revealed that the optimal settings for each model achieved noteworthy F1 scores of 0.902, 0.895, and 0.752, respectively. These results underscore the efficacy of the models in predicting tweet deletions and classifying disinformative content. Notably, the identification of deleted tweets has the potential to contribute to the development of annotated datasets for misinformative and disinformative categories, representing a significant advancement in the detection of disinformation on social media.

III. PROBLEM DEFINITION

This work tackles the challenge of identifying deception in Arabic text. We approach this problem in two ways: First, by building a model that can classify Arabic text as either deceptive or non-deceptive. Second, by investigating the impact of incorporating additional textual features into the detection process. In essence, this classification task involves feeding the model Arabic text, the model then analyzes this data and predicts a binary outcome. Formally, our problem is defined as follows:

A. PRELIMINARIES

Let $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n\}$ denotes a set of n documents, and each document $\mathbf{d}_i$ is represented as an ordered sequence of words (a.k.a., terms), denoted as $\mathbf{d}_i = (t_i^1, t_i^2, \ldots, t_i^m, \ldots)$, where $|\mathbf{d}_i| \in \mathbb{N}$.

Let $\mathcal{Y} = \{y_\mathbf{d}\} \forall \mathbf{d} \in \mathcal{D}$ be the set of ground truth labels for each document, where $y_{\mathbf{d}_i} = 0$ means document $\mathbf{d}_i$ is a deceptive document, and $y_{\mathbf{d}_i} = 1$ means document $\mathbf{d}_i$ is a non-deceptive document.

B. PROBLEM DEFINITION (TASK 1)

Given a pair of documents and labels $(\mathcal{D}, \mathcal{Y})$, our objective is to learn a function f that maps a document $\mathbf{d}$ to a label y. This is given by the formula in Eq. 1.

$$y = f(\mathbf{d}) \tag{1}$$

In this case, the function f represents a supervised binary classifier. f can be a machine learning model, a deep learning model or a transformer-based model. Details of the selected models are presented Section IV. Different subsets of the dataset $(\mathcal{D}, \mathcal{Y})$ are used to train/validate and test the performance of the model f.

TABLE 1. Overview of selected datasets.

Dataset	Genre	Source	Labelling Technique	Count / Classes
ArabicFakeNews [17]	Formal Articles	Real news: well known news agencies Fake news : Fact Checking Platforms	Fact checking	6000 Real, 2016 Fake
Arabic Fake News [9]	Formal Articles	Real news : well known news agencies Fake news : crowdsourcing	Fact checking	549 Real, 549 Fake
ArCOVID-19 Rumors [18]	Informal Tweets	Twitter	Manual annotation	1831 True , 1753 False
Arabic Translated Amazon Reviews [19]	Informal Reviews	Amazon reviews	reviews extraction	20k Real , 14k Fake
Spam and Ham [20]	Informal Tweets	Ham: well known news agencies Spam: keyword extraction	Hybrid (Manual and keyword extraction)	11.2k Ham , 2.2k Spam

C. PROBLEM DEFINITION (TASK 2)

Given a pair of documents and labels $(\mathcal{D}, \mathcal{Y})$, our objective is to investigate the impact of textual features learned from the raw dataset, $\mathcal{D}$. Thus, our objective in this task is to learn a function f that maps features learned from document $\mathbf{d}$, $textualFeatures(\mathbf{d})$, to a label y. This is given by the formula in Eq. 2.

$$y = f(textualFeatures(\mathbf{d})) \tag{2}$$

Similarly to Task 1, f can be a machine learning model, a deep learning model or a transformer-based model. The representation of the model's input (i.e., feature engineering) varies according to the selected model. Details of model selection for f, and feature engineering are provided next in Section IV.

IV. METHODOLOGY

To address the problem defined in Section III, we employ the pipeline presented in Fig. 1. The proposed pipeline is composed of five main components: dataset (Section IV-A), data pre-processing (Section IV-B), feature engineering (Section IV-C), model selection and training (Section IV-D), and finally model evaluation (Section IV-E). Each component is described in details in the following sections.

A. DATASET

In compliance with legal and ethical standards, this study uses only publicly available data designated for research purposes. It utilized a comprehensive Arabic dataset collected from various open-source repositories, previously employed by other scholars/articles with a primary focus on detecting deception. We considered both formal and non-formal text genres in the collection in order to train and test various models. This deliberate inclusion aims to provide a diverse range of linguistic styles, vocabulary, and to ensure inclusiveness of cultural and dialectical variations for the models to learn from.

For formal text genre, the datasets included were from works of [9] and [17]. The formal dataset included fake news articles that were published or generated on Twitter (which is currently named X), from accounts of fact-checking platforms, which are Akeed,[1] Misbar,[2] and Fatabyyano.[3] The latter included fake news articles that were human-generated and modified real articles to produce fake articles. Both datasets contained formal language imitating those found in a journalistic text genre, whereas the real articles in both datasets were collected from widely known news agencies such as Aljazzera,[4] Okaz,[5] Russia Today Arabic,[6] and Sabq.[7]

For the informal text genre, we used what has been used in previous research to detect fake text which are rumors tweets, reviews, and spam tweets. The rumors tweets dataset was from the study of Houari et al. [18], which included rumors about COVID-19 detected on Twitter. The fake reviews dataset was the work of Alharthi et al. [19], which was an Arabic-translated version of Amazon Review Data[8] and it included original customer reviews and computer-generated fake reviews. The Arabic spam tweets dataset was a presented work of [20], which included a variety of spam and ham tweets displayed on Twitter. The ham tweets were extracted from well-known news platforms such as Al Arabiya,[9] Al Hadath,[10] and Sky News Arabia[11]. Table 1 displays an overview of the selected datasets, and Table 2 shows a sample of deceptive text (a.k.a., spam, fake, false) and non-deceptive text (a.k.a, real, true, ham).

Upon combining all the datasets, a significant imbalance between the two classes would arise. To overcome the imbalance of the gathered datasets, we selected the first 500 articles

[1] https://www.jmi.edu.jo/en/akeed-duke-reporters
[2] https://misbar.com/en/about-us
[3] https://fatabyyano.net/
[4] https://www.aljazeera.com/
[5] www.okaz.com.sa
[6] https://arabic.rt.com/news/
[7] www.sabq.org
[8] https://www.kaggle.com/datasets/rogate16/amazon-reviews-2018-full-dataset
[9] https://www.alarabiya.net/
[10] https://www.alhadath.net/News
[11] https://www.skynewsarabia.com

TABLE 2. Sample dataset for deceptive and non-deceptive classes.

Category	Text
Deceptive (a.k.a., spam, fake, false)	ايلتس نصدر شهادات ايلتس رسميه و معتمده صادره من دون الحاجه لاختبار الشهاده رسميه لها و تكون مسجله في موقع نظاميه و تستطيع ان تاكد منها في الموقع بيدك فقط تواصل معنا عبر الخاص
Non-Deceptive (a.k.a, real, true, ham)	لقد توقفت عن العمل ولكنها تجاوزت الإطار الزمني لشهر واحد . الكثير من المال مقابل شهر آخر.

from each dataset to form this dataset. This method ensured an inclusive and varied collection of articles for our research, capturing a broad spectrum of genres while maintaining a manageable size for thorough examination. First, since the least number of articles were from the deceptive-formal genre in [9] and [17], we merged all of the available articles in the dataset, which included 2500 deceptive formal articles. Second, for the non-deceptive-formal genre, we selected the first 2000, and 500 articles from the same previous datasets, respectively. Third, for the informal genre for both classes, deceptive or non-deceptive, we selected the first 834 articles compatible in length across all [18], [19], and [20] datasets for both classes, to ensure balance with the formal genre. Throughout this process, 2,500 text statements were labeled as 'deceptive' and 2500 text statements labeled as 'non-deceptive', and diversely written in formal and informal text genres.

Furthermore, we have developed an Arabic phishing email dataset to evaluate the performance of the optimized trained models on an unseen dataset. We selected phishing emails because they demonstrate a deceptive behavior akin to that observed in the previously created datasets. Additionally, there is limited availability of Arabic phishing email datasets, despite their significant impact on individuals.[12] We relied on translating an open-source English phishing emails dataset compiled by [21] to Arabic, and balancing it with non-phishing emails from the Enron project published publicly in Carnegie Mellon University's School of Computer Science.[13] Both datasets included labels such as the date of the email and email ID. Since these labels are irrelevant to our work, we only extracted the ''context'' label, which contains the textual content of the email. We employed Marian MT,[14] a neural machine translation tool, to translate the emails. This tool has been found effective in similar translation projects [22], [23]. A total of 100 phishing and 100 non-phishing emails were incorporated into the development of the Arabic phishing email dataset. Phishing emails were labeled as 'deceptive' and non-phishing emails were labeled as 'non-deceptive'. The dataset can be freely accessed in Github[9].[15]

[12]https://gulfbusiness.com/saudi-arabia-led-gcc-in-number-of-phishing-attacks-in-q2-kaspersky-report/

[13]http://www.cs.cmu.edu/enron/

[14]https://marian-nmt.github.io/

TABLE 3. Statistical summary of training, testing, validation datasets.

Text Genre	Deceptive	Non-Deceptive	Average No. of Words
Formal	2500	2500	171.3
Non-Formal	2500	2500	34.7
Unseen dataset (Emails)	100 (Phishing)	100 (Non-Phishing)	202.4

Table 3 shows the statistics for used datasets. Overall, the inclusion of multiple text genres from different sources ensures that our study accounts for cultural and dialectical diversity, mitigating potential biases in the analysis of deception detection.

The decision to focus on protecting user privacy, ethical concerns, and the need to adhere to stringent data protection regulations, we selected datasets that were publicly accessible or provided upon request by their original authors. The datasets did not include any personally identifiable information about the contributors. In this study, the datasets only included the textual content and class labels for the purpose of this study.

B. DATA PRE-PROCESSING

Text pre-processing stands as a vital phase in natural language processing (NLP), encompassing the cleansing and transformation of unstructured text data to prepare it for classification using selected models. The dataset utilized in this study includes various genres, including formal and informal texts such as tweets, often presented in an unstructured form. It also may exhibit diverse word variations for a single term and it may contain emojis, hashtags, and nonalphabetic characters. Consequently, applying feature extractions and text classification to such data can yield poor results. To address these challenges, the following common Arabic text pre-processing steps were conducted using Tasaheel tool[16] [24]:

- **Cleaning**: This involves the removal of diacritics, punctuation marks, and non-alphabetic characters such as emojis, hashtags, emails, and web page links.
- **Normalization**: This involved elongating words by removing the repetition of three or more characters normalized the three Arabic letters: Alef (أ), Alef maqsoura (ى) and ta-marbouta (ة).
- **Tokenization**: This involves separating a piece of text into smaller chunks called tokens. It is important

[15]https://github.com/Hanen-Tarik

[16]Tasaheel is an Arabic NLP toolkit that provides several NLP tasks including pre-processing and automotive textual analysis.

TABLE 4. Comparison of text data before and after pre-processing.

Task	Before Preprocessing	After Preprocessing
Tokenization	الملابس جميله و ناعمه و تتحمل الغسيل عدة مرات	ال + ملبس جميل + ه + و + ناعم + ه + و + ت + تحمل + ال + غسيل + عدة + مرات
Normalization	الملابس جميله و ناعمه و تتحمل الغسيل عدة مرات	الملابس جميلة و ناعمة و تتحمل الغسيل عدة مرات
Cleaning	طلبت 10 وحدات منها كانت سيئة للغاية !	طلبت 10 وحدات منها و كانت سيئه للغايه

because unstructured text can be converted into independent words that can be easily analyzed.

Table 4 provides a comparison of text data examples before and after undergoing three pre-processing tasks: tokenization, normalization, and cleaning.

The pre-processing steps had a significant impact on enhancing the data quality. By meticulously cleaning and normalizing the text, we greatly reduced unwanted noise and ensured the model focused on relevant linguistic features. Additionally, tokenization structured the text for analysis, making it easier to extract meaningful patterns and features efficiently.

C. FEATURE ENGINEERING

In order to feed the pre-processed data to selected models, we use the standard TF-IDF representation for unstructured documents, which is further described in Section IV-C1. Additionally, we employ a set of diverse textual features, described in detail in Section IV-C2, with the objective of testing its adequacy in detecting Arabic deceptive text.

The TFDIF method is a prominent approach for extracting features from text and has shown great effectiveness in tasks concerning text classification. They have been proven to be vital in training Arabic text classification models in general, such as emotion classification [25], sentiment analysis [26], and in deception detection in various formats such as spam [27], fake online reviews [28], and phishing emails [5]. As for the textual features, the notion that deception includes an individual who has many emotions, motives, expressions, and behaviors that influence the behavioral signals they send to an outsider, motivated scholars to investigate these vital indicators to detect deception [29]. To study these indicators, they focused on extracting parts of speech [30], emotions [31], sentiment [32], and linguistics [33], [34], [35] from the text, which all proved to be essential in detecting deception.

1) TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY(TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) assigns weights to words based on their relative importance within a document and across the entire corpus. This technique aims to highlight words that are more distinctive and informative while moderating common words that carry less meaning.

Mathematically, TF-IDF for a term t in a document $\mathbf{d}$ is calculated as:

$$\text{TF-IDF}(t, \mathbf{d}) = \text{TF}(t, \mathbf{d}) \times \text{IDF}(t) \tag{3}$$

where $\text{TF}(t, \mathbf{d})$ is the term frequency of t in $\mathbf{d}$, representing the raw count of t's occurrences in document $\mathbf{d}$, and $\text{IDF}(t)$ is the inverse document frequency of t, calculated as:

$$\text{IDF}(t) = \log\left(\frac{n}{\text{df}(t)}\right) \tag{4}$$

where n is the total number of documents in the corpus, and $\text{df}(t)$ is the number of documents containing t.

2) TEXTUAL FEATURES

The author's writing style may result in language leakage, some of which may signal the presence of deceptive text [36], such as hedging words [37] and emotion words [38]. This being said, three textual feature sets were composed: morphological, psycholinguistics, and sociolinguistics. These textual feature sets were composed to analyze the text from different writing perspectives. Each of these categories is further explained below. The textual features were extracted using the Tasaheel textual analysis tool [24], and the list of all extracted textual features is available in Table 5. The extracted textual features are represented in a tabular form. These features were used to train selected ML and DL models, as depicted in Fig. 1.

First, **morphological features** are the theoretical study of lexemes, their linguistic structure, and their syntactic connections within a given language. It includes part-of-speech (POS) tags. These POS tags provide information about the grammatical role of each word in the text. Part of Speech is a word's assigned word tags that comply with its role in a sentence. These have been effective in detecting fake news [39], [40], and identifying spammers on Twitter [30], [41]. This textual feature category is capable of producing a comprehensive set of markers to investigate the text, as they are the main blocks used to create statements. Thus, the POS examined were limited to those specific POS that previous studies found useful in deception detection [39], [42], [43], [44], which are nouns, verbs, adverbs, adjectives, and proper nouns, conjunctions, prepositions, pronouns, and proper nouns.

Second, **psycholinguistics features** are primarily interested in exploring linguistic awareness and the cognitive processes involved in the development of thoughts. Linguistics are certain syntactic categories that are too fine-grained to be captured by general POS. Each syntactic unit conforms to a certain linguistic purpose, which is used to build meaningful statements. In some cases, the embedding of these linguistic categories identifies unique features associated with their writing and identifies certain characteristics of the deceiver [35], [45]. In this study, the set of deceptive cues as linguistic

label #	Label	Cleaned Text	nouns	verbs	adverbs	adjectives	pronouns	conjunctio	assertion	exclutive	negiation	Q
1	non_deceptive	سي ان ان تستعد اداره الرئيس	10	2	0	0	2	2	0	0	0	0
2	non_deceptive	حكم يتصدي لكره في طريقه	8	3	0	1	0	2	0	0	1	0
3	non_deceptive	تابعونا علي العربيه عبر برنا	5	1	0	0	2	1	0	0	0	0
4	non_deceptive	خبير بفريق التحقيق في منطه	10	2	0	5	1	6	0	0	1	0
5	non_deceptive	بالوثائق تعرف علي اهم الاتفا	4	1	0	2	2	1	0	0	0	0

FIGURE 2. Sample of some extracted textual features.

TABLE 5. Textual features.

Morphology	Psycholinguistics	Sociolinguitics
Noun	Negators	Religious
Verb	Assurance	Location
Adjective	Hedges	Title
Adverb	Justification	Nationality
Conjunction	Intensifiers	Organization
Prepositions	Quantifiers	Day, Time, Month
Proper Noun	Negative:Positive	Anger, Sad
	Exception	Disgust, Joy
	Opposition	Surprise, Fear

markers investigated are as follows: assurance [46], negations, [47], justification [48], intensifiers [35], hedges [49], illustrations [50] exceptions [34], and oppositions [33], positive and negative terms.

Third, **sociolinguistics features** investigate the correlation between society and language as a sub-field of linguistics. It encompasses the emotional and social elements that individuals engage with. The lying nature of the deceptive text has been found in several studies [31] associated with emotional language. Early studies suggested that liars tend to cover their lies by embedding emotional language within their writings [51]. More specifically, studies by [52] and [53] found that liars tend to use more emotional words to hide their lies. Meanwhile, some studies found a correlation between the use of highly emotional words and fake news [54], [55]. Therefore, six essential human emotions—anger, disgust, fear, sadness, joy, and surprise [51] and eight social elements, were used to analyze the sociolinguistics state of each text. Table 3 displays all twenty-nine textual features examined. Moreover, Figure 2 displays a sample of the textual features extracted for each article.

D. MODEL SELECTION

As mentioned earlier, the study aims to comprehensively evaluate different algorithmic approaches and their effectiveness in detecting deception in Arabic text. We believe that employing a multi-model approach contributes to a deeper understanding of the strengths and limitations of various machine learning methodologies in addressing this complex task. Firstly, we selected various ML models, including Logistic Regression (LR), Random Forests (RF), Decision Trees (DT), and k-nearest Neighbors (KNN), to cover a range of linear, non-linear, and tree-based approaches. Also, these models are recognized for their effectiveness in detecting deception in text across different languages, as supported by existing literatures [11] and [56]. However, we acknowledge that ML models may face challenges in capturing complex patterns and dependencies within text data, particularly in high-dimensional feature spaces, which could impact their accuracy in deception detection.

In addition to ML models, we incorporated DL models such as Long Short-Term Memory (LSTM) networks, Recurrent Neural Networks (RNNs), Gated Recurrent Units (GRUs), and Convolutional Neural Networks (CNNs). DL models are known for their ability to capture intricate patterns and dependencies in sequential data like text and have been used in detecting deception [57], [58]. However, DL algorithms are prone to overfitting with large datasets, computationally expensive training, and difficulty in interpreting model decisions. Furthermore, we employed transformer-based models, AraBERTv2 and Distilbert, due to their state-of-the-art performance in various NLP tasks, including Arabic text deception detection [12], [14], and [59]. These models utilize a self-attention mechanism to capture global dependencies, handle long-range dependencies, and are robust to sequence length variations. However, they require significant computational resources for training, and their complex architecture may pose challenges in interpretation.

In summary, our model selection process was guided by the goal of comprehensively evaluating different algorithmic approaches while considering their effectiveness, computational requirements, and interpretability in detecting deception in Arabic text. The following section provides an overview of the different models used in the study experiments.

1) MACHINE LEARNING (ML) MODELS

Four ML classification models—specifically Logistic Regression (LR), Decision Trees (DT), Random Forests (RF), and k-nearest Neighbors (KNN) were utilized.

1) **Decision Trees:** A decision tree serves as a supervised learning algorithm utilized for classification and regression tasks [60]. It adeptly discerns various

ways of dividing datasets under changing conditions. It strategically selects the best attribute, placing it at the tree's root, and subsequently partitions the training into subsets based on dataset feature values. The impact of nodes within the tree becomes more pronounced when they are closely positioned.

2) **Random Forests:** Random Forest is an ensemble learning technique consisting of numerous decision trees [61]. It utilizes a blend of bootstrap aggregating (bagging) and feature randomization to improve the accuracy of predictions. The individual decision trees are trained on random subsets of the data, and their results are combined to formulate predictions. This ensemble methodology enhances the overall performance of the model and reduces the
3) **Logistic Regression:** Logistic Regression is a supervised ML algorithm [62]. It operates on the principle of probability, constraining its cost function to values of 0 or 1. LR is recognized for its cost-effectiveness, although its efficacy diminishes with high-dimensional datasets. Additionally, its performance varies depending on whether the data is linear or non-linear. LR tends to excel when data separability is high, positively impacting its overall performance.
4) **k-Nearest Neighbors:** It is one of the simplest classification methods. Classification in kNN relies on a majority vote from the nearest training instances. The distance metric, k distance, is computed using common arithmetic measures [63].

2) DEEP LEARNING (DL) MODELS

Four DL models, namely Long Short-Term Memory (LSTM) networks, Recurrent Neural Networks (RNNs), Gated Recurrent Units (GRUs), and Convolutional Neural Networks (CNNs), have also been employed.

1) **Recurrent Neural Networks:** Are specifically designed for learning sequences of data, primarily employed in the classification of textual data. The learning occurs at hidden recurrent nodes, relying on information from their preceding layers of nodes.
2) **Long Short-Term Memory:** is a type of RNN that captures sequential relationships among words within a sentence [64]. Given that textual information can be treated as time-series data, the order of words holds a pivotal role in shaping the meaning of sentences. The LSTM cell comprises four crucial components: the forget gate, output gate, input gate, and update gate. The forget gate determines what to discard from the previous memory units, the input gate decides what information to incorporate into the neuron, the update gate modifies the cell, and the output gate produces the new long-term memory.
3) **Gated Recurrent Units:** is a simplified variant of the LSTM, offering reduced training time and enhanced network performance [65]. The functionality of a GRU cell closely resembles that of an LSTM cell, yet the former combines the forget gate and input gate into a unified update gate, utilizing a single hidden state. Additionally, GRU merges the cell state and hidden state into a singular state, resulting in half the number of gates (update and reset gates) compared to LSTM.
4) **Convolutional Neural Networks:** is structured with a series of interconnected layers, where the output from one layer serves as input for the next. These layers include a convolutional layer, a pooling layer, and a fully connected layer, as outlined in the study. The initial layer, the convolution layer, incorporates multiple filters, 16 for different regions, each with 2 filters, to extract sentence features. These filters convolve the input, generating feature maps of variable lengths. The subsequent layer, max pooling, captures essential features from the prior maps. The last layer is a dense (fully connected) layer utilizing the sigmoid function as an activation function [66]. This layer produces the network's output, indicating whether the input sentence is positive or negative [67], [68]

3) TRANSFORMERS-BASED MODELS

This study incorporates two transformer-based models, namely AraBERTv2 and Distilbert, which achieve the best results in several NLP tasks. The transformer's architecture employs a self-attention mechanism, enabling the inclusion of information from any input token in subsequent layers of the network. Transformer-based models undergo pre-training on large volumes of text and are later fine-tuned for specific tasks using smaller datasets.

1) **AraBERTv2:** Bidirectional Encoder Representations from Transformers (Bert) is a DL-based NLP framework aiming to capture text contextual relations bi-directionally [69]. The bi-directional relation allows the Pre-Trained Embedding to train on the provided text's left and right context to better understand its variation. Furthermore, it continues learning by applying unsupervised learning on unlabeled text, allowing it to improve its performance and making it suitable for applications like Google Search. The model utilizes the 'asafaya/bert-base-arabic' pre-trained transformer model to tokenize Arabic text data (a.k.a., AraBERTv2). This choice ensures an efficient representation of linguistic nuances in the language.
2) **Distilbert:** is a lightweight variant of BERT that uses minimal resources to generate comparable results to the BERT model [70]. It uses the distillation method, creating small models to mimic the process and representation of the larger BERT model. Each smaller model learns from BERT and updates its weight using the following parameters: Distillation loss, similarity loss, and masked language model mask. The lightweight nature of Distilbert makes it suitable for resource-constrained devices.

The previous base transformers follow a standard sequence classification architecture. They include embedding for

TABLE 6. Transformer-based models and details.

Models	Checkpoint Name	Total Parameters	Size	Training Corpus
AraBERTv2	'asafaya/bert-base-arabic	135, 194, 882.00	515.73 MB	Oscar
Distilbert	distilbert-base-multilingual-cased	135, 326, 210.00	516.23 MB	Wikipedia

words, positions, and token types. The encoder comprises multiple layers, each containing a self-attention mechanism and feedforward layers. A pooling layer is employed to generate a fixed-size vector for the final classification. To leverage pre-trained language understanding while adapting to the specific classification task, the transformers' parameters are frozen. Only the classifier layers are fine-tuned during training. The details of these transformers, including checkpoint names, total parameters, size, and the training corpus associated with each, are detailed in Table 6.

With the objective to take full advantage of the features generated by the basic word embedding and textual features integrated with the best-performing transformer, we develop a novel-enhanced AraBERTv2 model that is trained on these features. In addition to processing text with AraBERTv2, the enhanced model incorporates additional linguistic features using a feature extractor. This feature extractor consists of a linear layer as a fully connected multi-layer neural network, followed by a ReLU activation function, processing linguistic features before concatenating them with the AraBERTv2 output. The classifier layer takes the combined output of the AraBERTv2 model and the feature extractor, using this representation for the final classification. Similar to the base AraBERTv2 transformer, AraBERTv2 parameters are frozen during training, allowing the model to benefit from pre-trained language understanding while adapting to the task at hand. This model has approximately 2.85 million trainable parameters, reflecting the additional parameters introduced by the feature extractor and modified classifier layer. In general, we fine-tune the model in accordance with embedding these features. Specifically, the base model, the dropout layer, and the classification layer are then jointly trained on the input text embedding and the supplementary features against the supplied target class (deceptive or non-deceptive), as shown in Figure 3.

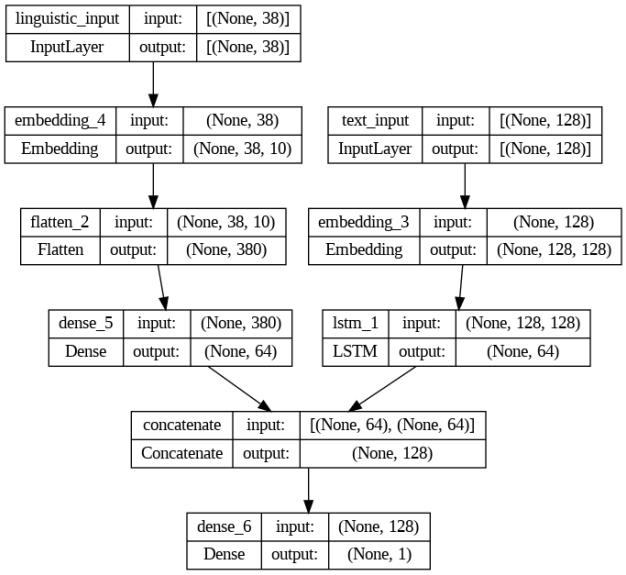

FIGURE 3. Enhanced transformer model architecture.

E. MODEL EVALUATION

Various metrics are used to evaluate the performance of all compiled classification models. Common evaluation metrics [71] were selected, as described below:

- **Precision:** It quantifies the precision of positive predictions, representing the proportion of correctly predicted positive cases out of all positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

- **Recall:** It evaluates the model's proficiency in identifying all relevant instances. It measures the proportion of true positive predictions among all actual positive cases.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

- **F1-Score:** It represents a harmonic mean of precision and recall, providing a balanced assessment of these two metrics. Particularly beneficial when seeking a single metric combining precision and recall, it is calculated as:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

- **Accuracy:** It is the ratio of accurate predictions to the total number of predictions, which provides a comprehensive evaluation of the model's performance.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

where TP, TN, FP and FN stand for True Positive, True Negative, False Positive, and False Negative, respectively.

- **AUC-ROC**: The AUC-ROC curve, also known as the Area Under the Receiver Operating Characteristic curve, is a visual depiction that shows how well a binary classification model performs at different classification levels. In machine learning, it is standard to use this method to evaluate a model's capacity to differentiate between two classes, generally referred to as the positive class and the negative class.

V. EXPERIMENTS

A. EXPERIMENTAL SETTINGS

In this section, we present the experiments for the compiled models adopting ML, DL, and transformers. Extensive

TABLE 7. Hyperparameter tuning for ML models.

Model	Hyperparameter Tuning
LR	C': 0.1, 'penalty': 'l1', 'solver': 'saga', random_state=50
DT	criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 2, 'random_state' = 100
RF	max_depth': 15, 'n_estimators': 100, 'random_state' =50
KNN	n_neighbors': 11

TABLE 8. Parameters and configurations for DL models.

Model	Trainable Parameters	
LSTM	74602	batch_size=100, epochs = 10, loss = Categorical_Crossentropy, optimizer = Adam, Embedding = 500, 120
RNN	63152	
GRU	69502	
CNN	72254	

experiments for Arabic deception detection were carried out to show the impact of using TF-IDF, textual features, pre-trained transformers, and an enhanced AraBERTv2 by fine-tuning the model and employing the deceptive specific textual features. This study utilizes essential feature engineering approaches, namely TFIDF and textual features, in ML and DL models. These strategies are not used in transformer-based models. Transformers have a dual purpose, operating as both feature extractors and classifiers. Manual feature engineering is unneeded for transformer models.

However, specifically for ML, it is necessary to use comprehensive feature engineering methods in order to extract important and relevant features from the text. The transformers' models use pre-trained contextual word embeddings with attention methods to preserve semantic information. They take tokenized special text as input. The TFIDF and textual characteristics are unsuitable for representing semantic information, which results in the loss of semantic information during the creation of transformers. To achieve this, appropriate tokenizers were applied for the transformers.

Table 7 details hyperparameter tuning for the ML models employed, while Table 8 provides a detailed breakdown of the parameters assigned for the DL models. Moreover, Table 9 presents the hyperparameters of the advanced pretrained transformer-based models. The majority of parameters reside within the pre-trained models which represent the core architecture. We used HuggingFace's Transformers in [72] to leverage the pre-trained models.

The experiments have been implemented using a server endowed with an Intel(R) Xeon(R) E5-2670 CPU, NVIDIA(R) GeForce GTX 1080 GPU with 8 GB video memory, and 64 GB RAM. The models were utilized by

TABLE 9. Parameters and configurations for transformers.

Parameters	Value
Epochs	10
Dataset Batch Size	8
Training Batch Size	300
Initial learning rate	5.00e-05
Optimizer	Adam
Loss	SparseCategoricalCrossentropy

Google Colab[17] running on Python 3.0. The organized dataset included 5k deceptive and 5k non-deceptive Arabic textual statements. To test and evaluate each compiled model, it was split into 80% (8000 statements) training, 10% (1000 statements) validation, and 10% (1000 statements)testing. These computational settings generated averages of samples per second, steps per second, and overall runtime for all the developed models, which by 397.03, 54.1, and 0.054, respectively.

B. EXPERIMENTAL RESULTS

In this section, the performance of ML, DL, and transformer models trained on diverse feature sets in identifying deceptive or non-deceptive Arabic text is evaluated. Tables 10, 11, and 12 offers a comprehensive summary of the classification metrics, including precision, recall, F1-score, and accuracy, for both deceptive and non-deceptive classes.

1) ML MODELS RESULTS

Table 10 displays the performance of the four ML models in classifying the text as deceptive or non-deceptive, utilizing various features. When employing TF-IDF, RF obtained an accuracy of 92%, whereas LR, DT, and KNN achieved an overall accuracy of 91%,83%,and 91%, respectively. LR and DT showed a similar accuracy of 84% when using textual features. RF also performed well with an 90% accuracy, providing balanced precision, recall, and F1-scores for both classes. Notably, TF-IDF features consistently achieved the best results across all ML models and metrics. This suggests that, in this context, the choice of features, particularly TF-IDF, significantly contributed to the overall model performance.

2) DL MODELS RESULTS

As shown in Table 11, the performance of the DL models across different features reveals varying performances in categorizing text as deceptive and non-deceptive. When the DL models employed TF-IDF features for text classification, they showed low performance. Specifically, the LSTM and RNN models demonstrated unusual behavior, achieving a precision score of less than 50% for deceptive cases but a corresponding similar score for recall, indicating a failure

[17] https://colab.google/

TABLE 10. Performance of ML models.

TF-IDF						
Model	**Class**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC-ROC**
RF	Deceptive	0.89	0.96	0.93	0.92	
	Non-Deceptive	0.95	0.87	0.91		0.93
LR	Deceptive	0.92	0.95	0.94	0.91	
	Non-Deceptive	0.95	0.91	0.90		0.91
DT	Deceptive	0.81	0.86	0.81	0.82	
	Non-Deceptive	0.85	0.80	0.83		0.83
KNN	Deceptive	0.92	0.90	0.93	0.91	
	Non-Deceptive	0.94	0.91	0.92		.92
Textual Features						
Model	**Class**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC-ROC**
RF	Deceptive	0.91	0.92	0.90	0.90	
	Non-Deceptive	0.84	0.91	0.87		0.90
LR	Deceptive	0.88	0.81	0.85	0.84	
	Non-Deceptive	0.81	0.88	0.84		0.85
DT	Deceptive	0.86	0.83	0.85	0.84	
	Non-Deceptive	0.82	0.86	0.84		0.86
KNN	Deceptive	0.88	0.83	0.87	0.87	
	Non-Deceptive	0.83	0.85	0.88		0.88

TABLE 11. Performance metrics for DL models.

TF-IDF						
Model	**Class**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC-ROC**
LSTM	Deceptive	0.42	0.48	0.52	0.44	0.50
	Non-Deceptive	0.41	0.57	0.52		
RNN	Deceptive	0.55	0.57	0.51	0.50	0.55
	Non-Deceptive	0.41	0.71	0.67		
GRU	Deceptive	0.67	0.87	0.71	0.67	0.69
	Non-Deceptive	0.2	0.41	0.34		
CNN	Deceptive	0.41	0.91	0.69	0.61	0.66
	Non-Deceptive	0.51	0.18	0.32		
Textual Features						
Model	**Class**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC-ROC**
LSTM	Deceptive	0.88	0.84	0.85	0.86	0.92
	Non-Deceptive	0.79	0.8	0.79		
RNN	Deceptive	0.74	0.89	0.86	0.89	0.96
	Non-Deceptive	0.85	0.89	0.81		
GRU	Deceptive	0.82	0.73	0.88	0.85	0.93
	Non-Deceptive	0.87	0.78	0.82		
CNN	Deceptive	0.86	0.71	0.81	0.89	0.95
	Non-Deceptive	0.84	0.88	0.85		

to identify any actual deceptive instances. Conversely, the precision for non-deceptive cases was moderate, suggesting some false positives. The GRU model faced challenges with a deceptive precision of 51% and a non-deceptive precision of 41%, while the CNN model struggled with imbalanced performance, resulting in a deceptive precision of 53% and a non-deceptive recall of 47%. The presence of low scores below 50% in precision and recall indicates the limitations of DL models when using TF-IDF features.

Textual features showcased strong performance, particularly with RNN and CNN achieving the highest accuracy at 89%. On the other hand, LSTM and GRU exhibited lower performance, achieving 86% and 85%, accuracy respectively.

3) TRANSFORMER-BASED MODEL RESULTS

Table 12 outlines the performance of AraBERTv2 and Distilbert, two transformer-based models known for their

TABLE 12. Performance metrics for transformer models.

Pre-Trained Embedding						
Model	**Class**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC-ROC**
AraBERTv2	Deceptive	0.91	0.95	0.91	0.93	0.98
	Non-Deceptive	0.96	0.93	0.97		
Distilbert	Deceptive	0.94	0.91	0.92	0.92	0.98
	Non-Deceptive	0.82	0.97	0.86		
Pre-Trained Embedding +Textual Features						
Model	**Class**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**AUC-ROC**
enhanced AraBERTv2	Deceptive	0.92	0.97	0.93	0.95	0.99
	Non-Deceptive	0.93	0.86	0.89		

effectiveness in NLP tasks. The results showed AraBERTv2 achieving 93% accuracy, with Distilbert close behind at 92%. Notably, the enhanced AraBERTv2 model, incorporating pre-trained embeddings and textual features, demonstrated improvement, achieving the highest accuracy of 95%. The model outperformed similar ML models compiled by [9] on a fewer set of textual features, reaching an accuracy of 78%. It also surpasses models developed by [73], achieving an accuracy of 88.8 % by utilizing the fastText pre-trained word embeddings in conjunction with CNN and BiLSTM. It also outperformed a more recent transformed-based model developed by [74], which reached 87%. This indicates the potential of leveraging both transformer-based models integrated with deceptive- -textual features to optimize the task of automating Arabic deception detection.

VI. ERROR ANALYSIS

We conduct a thorough analysis of the performance of the enhanced model, focusing on the misclassified text instances. The analysis found that 22% of them were of formal genre compared to 8% non-formal. Similar to the misclassified phishing emails previously stated, this may be explained as deceptive text written in a formal genre following a unified writing style, to imitate the formal writing style found in ‘‘non-deceptive’’ text, which causes minimum differences between them. For example, as seen in table 13, a non-deceptive article was classified as deceptive (FP) because it contained negative and intensifier terms such as خفض ، منع ، التراجع ، رفض ، خلف الكواليس , similar to those found in deceptive articles. This notion is mostly noted in studies such as [34] and [40], which found that deceivers tend to imitate the language of non-deceivers to present themselves as genuine. Compared to the fact that formal descriptive articles misclassified as non-deceptive (FN), as seen in table 13, contained no significant language difference, thus mistakenly classified. On the other hand, non-formal deceptive text was mostly attainable as it included many intensifiers, justification, hedges, and a variety of adverbs. Some of their content was composed of ‘‘made-up’’ events tailored with supporting adverbs in terms of time and place, to support their ‘‘made-up’’ events. Others, contained ‘‘ eye-catching’’ phrases tailored with intensifiers, to lure the reader to interact with their content. Nonetheless, our optimal

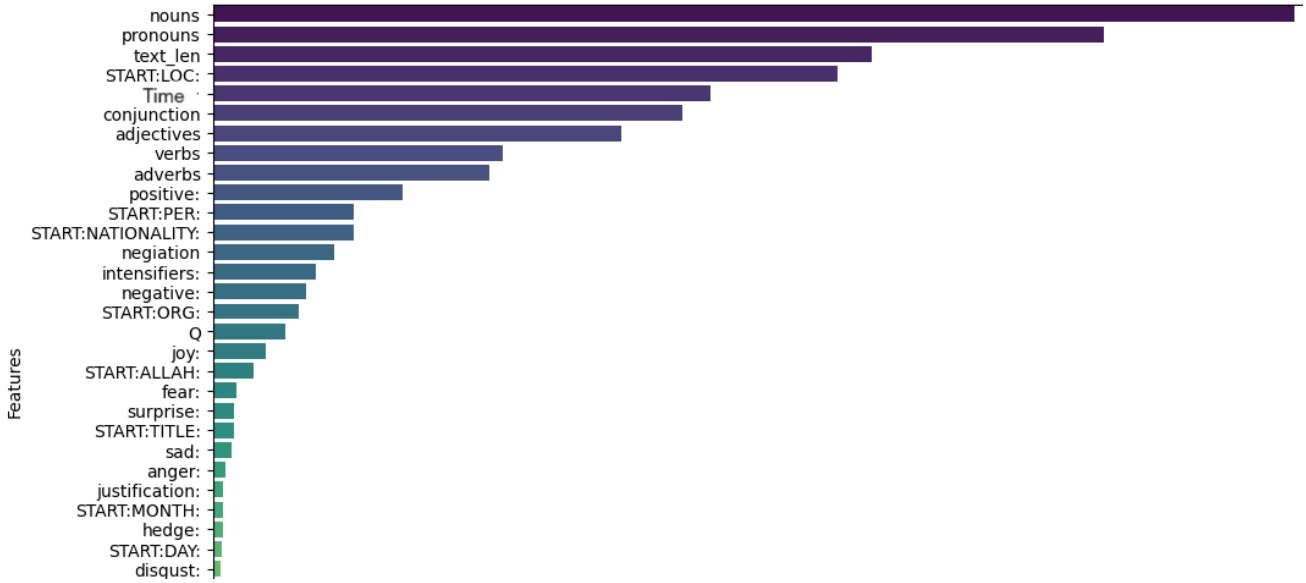

FIGURE 4. Dominant textual features.

model had the ability to detect deceptive text in several genres with respect to the formal and informal writing language. These findings are supported by the most dominant textual features found which are nouns, pronouns, location, time, conjunctions, and adjectives used to weave in the "made up" events, accompanied with details to support the fabrication. In support, we find that some of the dominant textual features are nouns, adjectives, adverbs, intensifiers, and proper nouns. Figure 4 displays them in order of their effect on the models' performance.

A. EXPERIMENTAL RESULTS ON UNSEEN DATASET-PHISHING EMAILS

Aiming to ensure the validation of our model's performance in detecting any form of deceptive text, we conduct this vial experiment. In this experiment, we assess the performance of the enhanced AraBERTv2 model not only against independent and unseen data but also against a new form of deceptive text (phishing email). The phishing emails dataset, composed of 200 emails (100 phishing and 100 non-phishing emails), is used as an input to the model as unseen data. The Phishing Email dataset was subjected to identical pre-processing steps as the main dataset used for training models, followed by the extraction of textual features from it. The results of ML models, RF, DT, KNN, and LR on the unseen dataset were a range of low accuracy from KNN with TF-IDF, 40%, and the highest accuracy in RF with textual features, 59%. Moreover, the accuracy of the DL models was achieved by CNN trained on TF-IDF, 51%, with the lowest attained by RNN trained on textual features, 48%. Furthermore, concerning the transformers, the highest accuracy was achieved by AraBERTv2, 50%, and a poor result by Ditibert, 46%.

TABLE 13. Samples of misclassified articles.

Deceptive articles misclassified as non-deceptive (FP)	تبدأ اليوم بعثة وزارة التضامن الاجتماعي في تصعيد حجاج الجمعيات الاهلية لمشعر عرفات وذلك من خلال ٥٠ باص فقط بنظام الرد الواحد، وطالبت وزارة التضامن الاجتماعي المزيد من الباصات، وهذا أدى الى حنق الحجاج بسبب قلة وسائل النقل. وفي اتصال هاتفي لرئيس البعثة وجهت غادة والي وزيرة التضامن الاجتماعي تحذيرات للبعثة بضرورة تشكيل غرفة عمليات لمتابعة خطة تصعيد الحجاج والمتابعة المستمرة لهم خلال فترة المشاعر، وسرعة توفير العدد المناسب من الباصات
Non-deceptive articles misclassified as deceptive (FN)	اهتمت الصحف العالمية الصادرة اليوم، الثلاثاء، بعدد من القضايا أبرزها مساعى البيت الأبيض لمواجهة أى ركود اقتصادى محتمل، وعودة من جديد فى سوريا والعراق، إلى جانب تطورات بريكست قالت صحيفة نيويورك تايمز الأمريكية إن مسئولى البيت الأبيض بدأوا فى الإعداد لخيارات للمساعدة فى تدعيم الاقتصاد الأمريكى ومنعه من السقوط فى حالة ركود، بما فى ذلك التفكير فى خفض محتمل لضريبة الرواتب وربما التراجع عن بعض التعريفات الجمركية التى أعلنها ترامب، بحسب ما قاتل مصادر مطلعة على المناقشاتوكان ترامب قد أصر على أن الاقتصاد يؤدى أداء مذهلا، ورفض هو ومستشاروه علانية أى إشارة إلى ركود وشيك. لكن خلف الكواليس، فإن فريق ترامب الاقتصادى يضع معا خططا فى خلال ما استم ضعف الاقتصاد.

The performance of the enhanced AraBERTv2 in terms of its prediction of the unseen test data is displayed in Table 14. According to the results, we find that the model achieved 70% F1-score for deceptive and 67% for non-deceptive text. Once more, we discover that non-deceptive

TABLE 14. Performance of enhanced AraBERTv2 on unseen data.

Model	Class	Precision	Recall	F1-score	Accuracy
enhanced AraBERTv2	Deceptive	0.73	0.65	0.70	0.69
	Non-Deceptive	0.70	0.68	0.67	

text could be challenging for the model to classify as they lack deceptive textual cues compared to deceptive text. Though the model's performance was not as high as in the previous trials, we suggest that the nature of phishing emails presented in a formal language imitating the non-phishing (legitimate) emails, caused a challenge for the model to detect useful deceptive cues. Nevertheless, the enhanced AraBERTv2 achieved a 69% accuracy in detecting deceptive text, which demonstrates the beneficial impact of encompassing these textual features in the model's compilation.

VII. DISCUSSION AND LIMITATION

This research aims to propose a methodology for an Arabic deception detection framework capable of discerning various text genres, including formal (e.g., news articles) and non-formal (e.g., tweets and reviews). The goal is to frame this as a binary classification task, employing diverse models that incorporate innovative ML, DL, and transformer models with different features and comparing them. We also investigated the effectiveness of deceptive cues as textual features combined with pre-trained word embeddings for models compiled by transformers. The textual features used to train the assembled ML and DL models achieved high performance, with accuracies over 80% in all ML models and LSTM in DL models. The transformers trained by pre-trained embedding models gave high accuracies, with AraBERTv2 reaching the highest 92%. This may be explained as large-scale text corpora are utilized to train pre-trained transformers with unsupervised or semi-supervised learning objectives. Without the need for explicit textual features, these objectives, such as language modeling or obscured language modeling, motivate the model to acquire meaningful representations directly from the raw text data. Training the transformed-based models on only these features might limit their learning ability to different beneficial features. In support, compiling an enhanced AraBERTv2 model pre-trained embedding and textual features, improved its performance by 2 %, reaching 95%. Detecting deceptive text by employing AI models, like any other natural language processing task, comes with several limitations. Some of these limitations include the challenging issue of detecting formal written text. As seen earlier this type of deceptive text has a similar linguistic pattern as that in the non-deceptive text, making it difficult to differentiate even to humans. Another limitation is the scarcity of available studies that indicate Arabic deceptive linguistic cues that could be used to detect deceptive text.

VIII. CONCLUSION

This study makes a significant contribution to the literature on deceptive text detection in the Arabic language. Specifically, we introduced a unique dataset encompassing formal and informal genres, reflecting the real-world complexities of Arabic communication. Our evaluation explored the effectiveness of various machine learning models, deep-learning and transformer-based models on this dataset. The results of various experiments demonstrated the superiority of the fine-tuned AraBERTv2 model enriched with textual features; including morphological, psycholinguistic, and sociolinguistic features. This finding highlights the importance of incorporating these features alongside raw text data for improved deception detection in the Arabic language.

This research offers a valuable foundation for further exploration. Future studies can leverage our dataset to benchmark future models as well as investigate the generalizability of our findings to other languages. Additionally, research can explore the impact of domain-specific features on deception detection accuracy within the Arabic language. By building upon these findings, we can develop robust and comprehensive tools for combating deceptive information in the ever-evolving digital landscape of Arabic communication.

For access to the Arabic phishing emails dataset and codes for the compiled models used in this study, please visit our GitHub repository.[18]

REFERENCES

[1] F. Alhayan, "Social media information credibility in the context of dementia," Ph.D. dissertation, Dept. Comput. Inf. Sci., Univ. Strathclyde, Glasgow, U.K., 2022.
[2] M. Bastos, S. Walker, and M. Simeone, "The IMPED model: Detecting low-quality information in social media," *Amer. Behav. Scientist*, vol. 65, no. 6, pp. 863–883, May 2021.
[3] D. B. Buller and J. K. Burgoon, "Interpersonal deception theory," *Commun. Theory*, vol. 6, no. 3, pp. 203–242, Aug. 1996.
[4] S. Girgis, E. Amer, and M. Gadallah, "Deep learning algorithms for detecting fake news in online text," in *Proc. 13th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2018, pp. 93–97.
[5] N. Jain, A. Kumar, S. Singh, C. Singh, and S. Tripathi, "Deceptive reviews detection using deep learning techniques," in *Proc. Int. Conf. Appl. Natural Lang. Inf. Syst.*, Salford, U.K. Cham, Switzerland: Springer, Jun. 2019, pp. 79–91.
[6] C. G. Harris, "Detecting deceptive opinion spam using human computation," in *Proc. Workshops 26th AAAI Conf. Artif. Intell.*, 2012, pp. 866–876.
[7] A. Alhogail and A. Alsabih, "Applying machine learning and natural language processing to detect phishing email," *Comput. Secur.*, vol. 110, Nov. 2021, Art. no. 102414.
[8] G. Krishnamurthy, N. Majumder, S. Poria, and E. Cambria, "A deep learning approach for multimodal deception detection," in *Proc. Int. Conf. Comput. Linguistics Intell. Text Process.* Cham, Switzerland: Springer, 2018, pp. 87–96.
[9] H. Himdi, G. Weir, F. Assiri, and H. Al-Barhamtoshy, "Arabic fake news detection based on textual analysis," *Arabian J. Sci. Eng.*, vol. 47, no. 8, pp. 10453–10469, Aug. 2022.
[10] X. Zhang, J. Li, W. Chu, J. Hai, R. Xu, Y. Yang, S. Guan, J. Xu, and P. Cui, "On the out-of-distribution generalization of multimodal large language models," 2024, *arXiv:2402.06599*.
[11] A. Ziani, N. Azizi, D. Schwab, D. Zenakhra, M. Aldwairi, N. Chekkai, N. Zemmal, and M. H. Salah, "Deceptive opinions detection using new proposed Arabic semantic features," *Proc. Comput. Sci.*, vol. 189, pp. 29–36, Jan. 2021.

[18] https://github.com/Hanen-Tarik

[12] M. Hasanain, F. Alam, H. Mubarak, S. Abdaljalil, W. Zaghouani, P. Nakov, G. Da San Martino, and A. Freihat, "ArAIEval shared task: Persuasion techniques and disinformation detection in Arabic text," in *Proc. ArabicNLP*, Singapore, 2023, pp. 483–493.

[13] Z. S. Ali, W. Mansour, T. Elsayed, and A. Al-Ali, "Arafacts: The first large Arabic dataset of naturally occurring claims," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 231–236.

[14] B. Tuck, F. Qachfar, D. Boumber, and R. Verma, "DetectiveRedasers at ArAIEval shared task: Leveraging transformer ensembles for Arabic deception detection," in *Proc. ArabicNLP*, 2023, pp. 494–501.

[15] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools, With Shared Task Offensive Lang. Detection*, H. Al-Khalifa, W. Magdy, K. Darwish, T. Elsayed, and H. Mubarak, Eds. Marseille, France: European Language Resource Association, May 2020, pp. 9–15. [Online]. Available: https://aclanthology.org/2020.osact-1.2

[16] H. Mubarak, S. Abdaljalil, A. Nassar, and F. Alam, "Detecting and identifying the reasons for deleted tweets before they are posted," *Frontiers Artif. Intell.*, vol. 6, pp. 1–10, Sep. 2023, doi: 10.3389/frai.2023.1219767.

[17] I. Alnabrisi and M. Saad, "Detect Arabic fake news through deep learning models and transformers," *Expert Syst. Appl.*, vol. 251, Oct. 2024, Art. no. 123997, doi: 10.1016/j.eswa.2024.123997.

[18] F. Haouari, M. Hasanain, R. Suwaileh, and T. Elsayed, "ArCOV19-rumors: Arabic COVID-19 Twitter dataset for misinformation detection," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, and S. Touileb, Eds. Kyiv, Ukraine: Association for Computational Linguistics, Apr. 2021, pp. 72–81. [Online]. Available: https://aclanthology.org/2021.wanlp-1.8

[19] S. Alharthi, R. Siddiq, and H. Alghamdi, "Detecting Arabic fake reviews in E-commerce platforms using machine and deep learning approaches," *J. King Abdulaziz Univ., Comput. Inf. Technol. Sci.*, vol. 11, no. 1, pp. 27–34, Sep. 2022. [Online]. Available: https://journals.kau.edu.sa/index.php/CITS/article/view/273

[20] S. Kaddoura and S. Henno, "Dataset of Arabic spam and ham tweets," *Data Brief*, vol. 52, Feb. 2024, Art. no. 109904.

[21] S. Chakraborty. (2023). *Phishing Email Detection*. [Online]. Available: https://www.kaggle.com/dsv/6090437

[22] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. F. Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in C++," in *Proc. ACL*, F. Liu and T. Solorio, Eds., Jul. 2018, pp. 116–121.

[23] Y. Wu, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.

[24] H. T. Himdi and F. Y. Assiri, "Tasaheel: An Arabic automative textual analysis tool—All in one," *IEEE Access*, vol. 11, pp. 139979–139992, 2023.

[25] D. E. Cahyani and I. Patasik, "Performance comparison of TF-IDF and Word2 Vec models for emotion text classification," *Bull. Electr. Eng. Informat.*, vol. 10, no. 5, pp. 2780–2788, Oct. 2021.

[26] B. Das and S. Chakraborty, "An improved text sentiment classification model using TF-IDF and next word negation," 2018, *arXiv:1806.06407*.

[27] M. Adnan, M. O. Imam, M. F. Javed, and I. Murtza, "Improving spam email classification accuracy using ensemble techniques: A stacking approach," *Int. J. Inf. Secur.*, vol. 23, no. 1, pp. 505–517, Feb. 2024.

[28] R. Mohawesh, M. Al-Hawawreh, S. Maqsood, and O. Alqudah, "Factitious or fact? Learning textual representations for fake online review detection," *Cluster Comput.*, vol. 27, no. 3, pp. 3307–3322, Jun. 2024.

[29] M. Zuckerman, B. M. DePaulo, and R. Rosenthal, *Verbal and Nonverbal Communication of deception* (Advances in Experimental Social Psychology), vol. 14, L. Berkowitz, Ed., New York, NY, USA: Academic, 1981, pp. 1–59. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S006526010860369X

[30] Z. Alom, B. Carminati, and E. Ferrari, "Detecting spam accounts on Twitter," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2018, pp. 1191–1198.

[31] H. A. Sayyed, S. Rushikesh Sugave, S. Paygude, and B. N Jazdale, "Study and analysis of emotion classification on textual data," in *Proc. 6th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2021, pp. 1128–1132.

[32] S. Salgado and G. Bobba, "News on events and social media: A comparative analysis of Facebook Users' reactions," *Journalism Stud.*, vol. 20, no. 15, pp. 2258–2276, Nov. 2019.

[33] J. Karoui, F. B. Zitoune, and V. Moriceau, "SOUKHRIA: Towards an irony detection system for Arabic in social media," *Proc. Comput. Sci.*, vol. 117, pp. 161–168, Jan. 2017.

[34] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2011, pp. 309–319.

[35] T. Gröndahl and N. Asokan, "Text analysis in adversarial settings: Does deception leave a stylistic trace?" *ACM Comput. Surveys*, vol. 52, no. 3, pp. 1–36, May 2020.

[36] K. Demestichas, K. Remoundou, and E. Adamopoulou, "Food for thought: Fighting fake news and online disinformation," *IT Prof.*, vol. 22, no. 2, pp. 28–34, Mar. 2020.

[37] J. Islam, L. Xiao, and R. E. Mercer, "A lexicon-based approach for detecting hedges in informal text," in *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 3109–3113.

[38] M. Zloteanu, P. Bull, E. G. Krumhuber, and D. C. Richardson, "Veracity judgement, not accuracy: Reconsidering the role of facial expressions, empathy, and emotion recognition training on deception detection," *Quart. J. Experim. Psychol.*, vol. 74, no. 5, pp. 910–927, May 2021.

[39] J. Kapusta and J. Obonya, "Improvement of misleading and fake news classification for flective languages by morphological group analysis," *Informatics*, vol. 7, no. 1, p. 4, Feb. 2020.

[40] V. Perez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," in *Proc. Int. Conf. Comput. Linguist.*, 2018, pp. 3391–3401. [Online]. Available: https://aclanthology.org/C18-1287

[41] R. M. K. Saeed, S. Rady, and T. F. Gharib, "An ensemble approach for spam detection in Arabic opinion texts," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 1, pp. 1407–1416, Jan. 2022.

[42] B. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 11, Mar. 2017, pp. 759–766.

[43] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Inf. Sci.*, vol. 497, pp. 38–55, Sep. 2019.

[44] A. Adha, "Linguistic based cues in detecting deception in Indonesian language use," *Argumentum*, vol. 16, pp. 14–30, Jan. 2020.

[45] M. Hajja, A. Yahya, and A. Yahya, "Authorship attribution of Arabic articles," in *Proc. Int. Conf. Arabic Lang. Process.* Cham, Switzerland: Springer, 2019, pp. 194–208.

[46] S. F. Sabbeh and S. Y. Baatwah, "Arabic news credibility on twitter: An enhanced model using hybrid features," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 8, 2018.

[47] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, "Truth of varying shades: Analyzing language in fake news and political fact-checking," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2931–2937.

[48] G. Gravanis, A. Vakali, K. Diamantaras, and P. Karadais, "Behind the cues: A benchmarking study for fake news detection," *Expert Syst. Appl.*, vol. 128, pp. 201–213, Aug. 2019.

[49] A. Addawood, A. Badawy, K. Lerman, and E. Ferrara, "Linguistic cues to deception: Identifying political trolls on social media," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 13, 2019, pp. 15–25.

[50] K. Lagutina, N. Lagutina, E. Boychuk, I. Vorontsova, E. Shliakhtina, O. Belyaeva, I. Paramonov, and P. G. Demidov, "A survey on stylometric text features," in *Proc. 25th Conf. Open Innov. Assoc. (FRUCT)*, Nov. 2019, pp. 184–195.

[51] R. W. Levenson, P. Ekman, and W. V. Friesen, "Voluntary facial action generates emotion-specific autonomic nervous system activity," *Psychophysiology*, vol. 27, no. 4, pp. 363–384, Jul. 1990.

[52] L. M. Jupe, A. Vrij, S. Leal, and G. Nahari, "Are you for real? Exploring language use and unexpected process questions within the detection of identity deception," *Appl. Cognit. Psychol.*, vol. 32, no. 5, pp. 622–634, Sep. 2018.

[53] J. T. Hancock, L. E. Curry, S. Goorha, and M. Woodworth, "On lying and being lied to: A linguistic analysis of deception in computer-mediated communication," *Discourse Process.*, vol. 45, no. 1, pp. 1–23, Dec. 2007.

[54] J. P. Baptista and A. Gradim, "Understanding fake news consumption: A review," *Social Sci.*, vol. 9, no. 10, p. 185, Oct. 2020.

[55] L. Zhou, J. K. Burgoon, D. Zhang, and J. F. Nunamaker, "Language dominance in interpersonal deception in computer-mediated communication," *Comput. Hum. Behav.*, vol. 20, no. 3, pp. 381–402, May 2004.

[56] A. A. C. Delgado, W. B. Glisson, N. Shashidhar, J. T. McDonald, G. Grispos, and R. Benton, "Detecting deception using machine learning," in *Proc. 54th Hawaii Int. Conf. Syst. Sci.*, 2021, pp. 7122–7131.
[57] H. Saadany, E. Mohamed, and C. Orasan, "Fake or real? A study of Arabic satirical fake news," 2020, *arXiv:2011.00452*.
[58] S. Bajaj, "The pope has a new baby!" in *Fake News Detection Using Deep Learning*. Stanford, CA, USA: Stanford Univ., 2017, pp. 1–8.
[59] M. Al-Yahya, H. Al-Khalifa, H. Al-Baity, D. AlSaeed, and A. Essam, "Arabic fake news detection: Comparative study of neural networks and transformer-based approaches," *Complexity*, vol. 2021, pp. 1–10, Apr. 2021.
[60] P. H. Swain and H. Hauska, "The decision tree classifier: Design and potential," *IEEE Trans. Geosci. Electron.*, vol. GE-15, no. 3, pp. 142–147, Jul. 1977.
[61] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.
[62] G. King and L. Zeng, "Logistic regression in rare events data," *Political Anal.*, vol. 9, no. 2, pp. 137–163, 2001.
[63] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-6, no. 4, pp. 325–327, Apr. 1976.
[64] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[65] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, Dec. 2014. [Online]. Available: http://arxiv.org/abs/1412.3555
[66] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proc. Int. Workshop Artif. Neural Netw.* Malaga-Torremolinos, Spain: Springer, Jun. 1995, pp. 195–201.
[67] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 60, 2012, pp. 84–90.
[68] A. Mohammed and R. Kora, "Deep learning approaches for Arabic sentiment analysis," *Social Netw. Anal. Mining*, vol. 9, no. 1, pp. 1–12, Dec. 2019.
[69] J. D. M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NaaCL-HLT*, vol. 1, 2019, p. 2.
[70] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.
[71] C. E. Metz, "Basic principles of ROC analysis," *Seminars Nucl. Med.*, vol. 8, no. 4, pp. 283–298, Oct. 1978. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001299878800142
[72] A. Safaya, M. Abdullatif, and D. Yuret, "KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media," in *Proc. 14th Workshop Semantic Eval.*, 2020, pp. 2054–2059. [Online]. Available: https://www.aclweb.org/anthology/2020.semeval-1.271
[73] D. Nam, J. Yasmin, and F. Zulkernine, "Effects of pre-trained word embeddings on text-based deception detection," in *Proc. IEEE Int. Conf. Depend., Autonomic Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Aug. 2020, pp. 437–443.
[74] M. Schütz, A. Schindler, M. Siegel, and K. Nazemi, "Automatic fake news detection with pre-trained transformer models," in *Pattern Recognit. ICPR Int. Workshops Challenges*, A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, Eds., Cham, Switzerland: Springer, 2021, pp. 627–641.

FATIMAH ALHAYAN received the Ph.D. degree in computer science from the University of Strathclyde, Scotland, U.K. She is currently an Assistant Professor of computer science with the College of Computer and Information Science, Princess Nourah Bint Abdulrahman University, Saudi Arabia. Her research interests include information credibility, data mining, computational social science, machine learning, and natural language processing (NLP) in both English and Arabic languages.

HANEN T. HIMDI received the Ph.D. degree in computer science from the University of Strathclyde, Scotland, U.K. She is currently an Assistant Professor of computer science and artificial intelligence with the College of Computer Science and Engineering, University of Jeddah, Saudi Arabia. She is also an enthusiastic about pioneering new ideas and developing cutting-edge technologies. Several of her academic articles are devoted to the topic of creating AI models that are useful for the Arabic language. Her research interests include machine learning, natural language processing, textual analysis, deep learning, and the creation of AI models that make use of cutting-edge learning techniques.

BASMA ALHARBI (Member, IEEE) received the B.Sc. degree in computer science from Effat University, Jeddah, Saudi Arabia, in 2008, the M.Sc. degree in computer science from Durham University, Durham, U.K., in 2009, and the Ph.D. degree in computer science from the King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, in 2017. She is currently an Associate Professor with the Computer Science and AI Department, College of Computer Science and Engineering, University of Jeddah, Jeddah.