applied sciences

MDPI

Article

Comprehensive Study of Arabic Satirical Article Classification

Fatmah Assiri [1,†] and Hanen Himdi [2,*,†]

[1] Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia; fyassiri@uj.edu.sa
[2] Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah 21493, Saudi Arabia
* Correspondence: hthimdi@uj.edu.sa
† These authors contributed equally to this work.

Abstract: A well-known issue for social media sites consists of the hazy boundaries between malicious false news and protected speech satire. In addition to the protective measures that lessen the exposure of false material on social media, providers of fake news have started to pose as satire sites in order to escape being delisted. Potentially, this may cause confusion to the readers as satire can sometimes be mistaken for real news, especially when their context or intent is not clearly understood and written in a journalistic format imitating real articles. In this research, we tackle the issue of classifying Arabic satiric articles written in a journalistic format to detect satirical cues that aid in satire classification. To accomplish this, we compiled the first Arabic satirical articles dataset extracted from real-world satirical news platforms. Then, a number of classification models that integrate a variety of feature extraction techniques with machine learning, deep learning, and transformers to detect the provenance of linguistic and semantic cues were investigated, including the first use of the ArabGPt model. Our results indicate that BERT is the best-performing model with F1-score reaching 95%. We also provide an in-depth lexical analysis of the formation of Arabic satirical articles. The lexical analysis provides insights into the satirical nature of the articles in terms of their linguistic word uses. Finally, we developed a free open-source platform that automatically organizes satirical and non-satirical articles in their correct classes from the best-performing model in our study, BERT. In summary, the obtained results found that pretrained models gave promising results in classifying Arabic satirical articles.

Keywords: machine learning; deep learning; BERT; GPT; textual feature; n-grams

Citation: Assiri, F.; Himdi, H. Comprehensive Study of Arabic Satirical Article Classification. *Appl. Sci.* **2023**, *13*, 10616. https://doi.org/10.3390/app131910616

Academic Editor: Cosimo Nardi

Received: 1 August 2023
Revised: 4 September 2023
Accepted: 18 September 2023
Published: 23 September 2023

1. Introduction

Satire, as defined in the Oxford Dictionary, is the use of humor to criticize people. Satirical news, sometimes also called sarcasm in the literature [1], is a fairly recent type of satire that has gained popularity throughout the world. Online satirical news sources, such as The Onion (US), the Beaverton (Canada), or the Daily Mash (UK), present fictionalized mashups of current events with one or more fabricated parts. Thus, the phrases in such publications should not be treated as facts. Instead, it is up to the reader to determine the true meaning of the satirist [2].

Because the literal meanings of the words employed are different from the factual message, satirical news is frequently categorized as a subtype of fake news [3]. Satirists do not intentionally mislead their audiences; instead, they present indicators that enable their readers to recognize the satire, which is a key distinction from the more common usage of the phrase "fake news" [4]. Although satirists give audience members indicators to recognize the satire, there are times when the audience fails to recognize these indications and accepts the satire as factual content [5].

Satirical news is often contrasted with "normal" news [6]. Normal news combines facts, opinion, and entertainment, and it often provides information and critiques of high-profile individuals, problems, and/or events with an aim to amuse [7]. It tends to be more

plain in its meaning and may be simpler to grasp than satire. According to this viewpoint, consequences may stem from contrasting satirical news with mainstream news articles that include comparable or factual information.

According to the literature review of artificial intelligence studies that target satire classification conducted by [1], the techniques developed for Arabic satire detection used datasets in the form of tweets or headlines to train classification models. The written format of tweets or headlines provides a small amount of text compared to articles. Moreover, as tweets may be written informally, satirical material may be effortlessly detected when compared to satirical articles that are written in a formal journalistic manner. Although the possible effects of satirical news articles are potentially affected by a particular audience's acceptance of the humorous content within the articles [8], according to some researchers [9], news consumers who are less attentive to traditional forms of news may find soft news formats, such as satirical news, more appealing. Particularly, this may stir up readers' opinions through sensationalist journalism. Although the existence and magnitude of the effects of these reactions have been disputed, they still bear trace evidence in studies [10].

Satirical content can have ethical implications, especially when it targets marginalized groups. Additionally, it can be used to spread fake information and understate serious situations. Satirical content in the form of news articles is a thin line with fake news compared to satirical content in social media posts [11]. To illustrate the dangers of this, an apology from the fake news website FreedumJunkshun.com in late October was published after facing criticism over a post they published on October 26 that falsely claimed one of the American troops killed in Niger was a deserter. The website made the revelation on a related Facebook group named America's Last Line of Defense. That was a bold step from a PolitiFact-designated "parody" fake news site, and it serves as yet another example of how the line between satire and intentional misinformation is complicated (https://www.poynter.org/fact-checking/2017/a-satirical-fake-news-site-apologized-for-making-a-story-too-real/ (accessed on 1 September 2023)).

Satirical articles are a common form of commentary, but when written in a journalistic format, they can pose a unique danger to readers. Since some satirical articles are written to appear as fact, readers may be fooled into believing that the information contained in the article is true. This can lead to confusion and misinformation, as well as sectarianism and even hate speech if the satire is based on polarizing issues. Furthermore, it can be difficult for readers to distinguish between a serious article and another that is satirical, as subtle jokes and puns may go unnoticed if one is not familiar with the subject matter. As such, it is important for readers to be aware of the potential for satirical articles to be misleading, and to exercise caution when engaging with them. To aid with the aforementioned issue, we aim to develop a model that automatically classifies Arabic satirical articles. Additionally, we investigate the impact of several feature extraction methods on classifying Arabic satirical articles. To the best of our knowledge, this work is the first to present a thorough analysis in terms of Arabic satirical content written in a formal journalistic register. The main contributions of this study are:

- Compile a large Arabic news satirist articles dataset;
- Perform a thorough analysis of the impact of several traditional and innovative feature extraction methods for satirical articles classification;
- Build a satire classification model using machine learning (ML), deep learning(DL), and transformers;
- Perform a detailed linguistic analysis of the formation of satirical articles;
- Create an open-source satire classification platform.

The experiments ran on a dataset of satirical and non-satirical articles that were collected from real-world satirical and non-satirical news platforms. Data are found in GitHub (https://github.com/Noza1234?tab=projects (accessed on 1 September 2023)), and can be made available on reasonable request. Our research contributes to the field of Arabic natural language processing by shedding light on the compilation of Arabic satire classification

models using various feature extraction techniques integrated with machine learning, deep learning, and transformers.

The remainder of the paper is organized as follows. Section 2 gives the background for satirical news, then Section 3 presents related work in the field of satirical news detection. Section 4 describes the dataset, preprocessing operations, and the classification models. Finally, the design of the experiments along with their results is described in Section 5.

2. Background

2.1. Satirical News

Satirical news is defined as the use of humor, irony, exaggeration, or ridicule to expose and criticize certain viewpoints, particularly in the context of contemporary politics and other topical issues (https://www.oxfordreference.com (accessed on 1 September 2023)). Halfway Post news outlet (https://halfwaypost.com/2020/09/11/the-best-political-satire-of-president-donald-trump/ (accessed on 1 September 2023)) displayed several satirical articles making fun of US President Donald Trump. Some headlines read as "Donald Trump questioning the citizenship of Dr. Fauci" and "A Brand New Form of Dementia Just Got Named after Donald Trump". It has been used for centuries as a form of social commentary, with some of the earliest examples coming from ancient Rome. Satirical news has been employed by authors and political commentators throughout history in order to explore and discuss a wide range of topics from political systems to social issues [12]. The purpose of satirical news is twofold: first, it serves to educate its readers by highlighting key issues and points of contention in society; s econd, it is used to provide entertainment and encourage readers to think more critically about the world around them. Ultimately, satirical news provides an effective platform for people to reflect on the current state of affairs and develop meaningful solutions for the future.

Satirical news is an increasingly popular form of media, as it provides an entertaining and lighthearted take on a range of topics. Satirical news can be primarily broken down into three main categories: political, social, and cultural satirical news. Political satirical news focuses on the current events of the political world and the involvement of public figures, often parodying their words and actions, such as "The Colbort Report" (https://www.cc.com/shows/the-colbert-report (accessed on 1 September 2023)), which was hosted by Stephen Colbert from 2005 to 2014, an outspoken conservative presenter, who frequently satirized American politics through satire, parody, and irony. The impact of The Colbert Report's show garnered significant recognition following a recent poll conducted by Pew Research Center (https://www.pewresearch.org/short-reads/2014/12/12/for-some-the-satiric-colbert-report-is-a-trusted-source-of-political-news/ (accessed on 1 September 2023)) to measure its impact among adults on the internet. The findings revealed that almost 25% of males aged 18 to 29 reported obtaining news related to politics and government through The Colbert Report within the week before the poll. On the other hand, social satirical news addresses the social issues that affect society, such as gun control, poverty, and immigration, while using humor to bring attention to these issues. A famous satire television series named "Black Mirror" (https://cherwell.org/2019/01/13/black-mirror-art-as-social-satire/ (accessed on 1 September 2023)) received considerable reviews due to its skilled critique of the intricate dynamics between human civilization and technology. Every episode of the series centers around a distinctively disconcerting facet of technology, including a wide array of subjects, such as surveillance and mass media. Finally, cultural satirical news draws attention to the cultural norms of society and utilizes humor to comment on the customs and values of a population. An example of a more dangerous outcome of cultural satire is found in the story of the renowned Greek playwright Aristophanes. He was recognized as one of the earliest satirists in recorded history. Within the context of their theatrical works, the playwright engaged in satirical depictions of religious figures, politicians, and philosophers, employing a combination of comedy and sarcasm to show the cultural differences in the layouts of society. The theatrical production titled "The Clouds" employed satire as a means of ridiculing the esteemed

philosopher Socrates. The authorities in Athens; however, responded to this comedic portrayal with an excessive degree of gravity, potentially influencing their subsequent determination to carry out the execution of Socrates (https://literaryterms.net/satire/ (accessed on 1 September 2023)). Overall, satirical news is a great way to bring attention to important topics in a more entertaining and humorous manner [13].

Generally, satirical news can have a variety of effects on its audience. Positive effects include drawing attention to social issues, making people laugh, and challenging people's beliefs and opinions [14]. However, a negative impact can be used to spread misinformation to manipulate public opinion, and to create confusion. Additionally, satirical news can be used to create a false sense of security, to distract from real issues, and to spread hate and bigotry.

Although satirical news is a form of entertainment that is typically used to make fun of politics or current events, it carries with it a unique set of challenges. One such challenge is the potential for misinterpretation of jokes and satire, which includes the difficulty of distinguishing factual information from satire. As a result, satirical news is often accompanied by a disclaimer informing readers of the intended comedic nature of the piece. An example of the satirical effect of being misunderstood as factual was in 2014 when a satirical news site published a story claiming that an Ebola outbreak had occurred in Atlanta. The story was shared widely on social media and led to many people panic-buying supplies and even evacuating the city, despite there being no actual Ebola outbreak. This is a clear example of how satirical news can have real-world consequences [15]. In addition, it may address sensitive or controversial topics, such as politics, religion, or social issues, which may cause individuals damage. To avoid offending or alienating certain individuals or groups, it is necessary to use humor with care when addressing these subjects. Balancing humor while addressing these topics requires a delicate approach to avoid offending or alienating certain individuals or groups. Satirical pieces should aim to challenge perspectives and initiate dialogue without crossing the line into offensive territory. In light of the prior challenges, it is difficult to produce an original and creative satire. Satire thrives on novelty and originality, but it confronts the ongoing challenge of finding new angles, fresh perspectives, and inventive comedic techniques to keep its content engaging and relevant. To maintain audience interest, it is crucial to strike a balance between consistency and innovation. These challenges help maintain the satirical content in a somewhat fixed template that provokes humor. Analysts may make use of the satirical content found in the satirical articles and define them as cues or features that detect satire.

2.2. Feature Extraction Techniques

The extraction of features is the process of extracting the characteristics and information that best represent the data. Using the proper feature extraction technique could facilitate feature selection and dimensionality reduction, and improve the efficacy of the machine learning or deep learning model applied to the classification process [16,17]. According to the classification model that is being utilized, there are three techniques that can be applied.

2.2.1. Traditional Techniques

- **N-grams**: a string of n-syllables that are contiguous. By collecting the most common n-grams rather than the whole corpus, this method might be used to obtain a more accurate categorization [18]. The n-grams approach was used with several standard algorithmic combinations to identify satire within the Arabic language [19]. Moreover, the n-grams method was used with other methods such as TF-IDF to achieve higher prediction results in classification tasks [20]. N-grams can be used in language processing to analyze patterns and relationships between words or phrases in a text. This method can provide useful insights into the satirical content found as forms of exaggeration or humorous words, i.e., the more humorous words found, the more likely the article would be classified as satiric.

- **TF-IDF**: Term Frequency and Inverse Document Frequency. Here, a term's frequency in a corpus indicates how significant a role it plays in the text [18,21]. It is simple to compute and express word similarity, but the semantic problems that impact the algorithm's overall performance make it ineffective [22]. Satire detection in Arabic was performed using the TF-IDF approach in several studies, such as those that utilized to reflect tweets syntactic structure [23] or linguistic features [24]. Generally, the TF-IDF approach has been proven to be effective in various natural language processing tasks, such as sentiment analysis and text classification [25]. Furthermore, researchers have also explored the use of TF-IDF in combination with machine learning algorithms to improve the accuracy of satirical detection models [26,27]. Since TF-IDF is a numerical statistic used in information retrieval to determine the importance of a word in a document, it is a useful feature to detect the occurrence of satirical cues in a document. As the frequency of satirical elements increases, the probability of the model correctly classifying the piece as satire also increases.
- **Textual features**: addresses lexical, syntactic, and stylistic features. These attributes were useful in detecting satirical contexts in Arabic [28]. For instance, the authors of [29] employed stylistic elements that included quote marks, exclamation points, and questions to classify satirical text. Additionally, other features such as sentiment features, shifters, and contextual features were used for Arabic satire identification [28]. Regarding the shifters, they pick up a variety of linguistic phenomena, such as claims that are inconsistent with reality. They also detect instances of exaggeration, such as when people use the strong descriptive words "huge" and "gigantic". These linguistic phenomena help identify instances of Arabic satire, as they indicate a deviation from the norm or an intentional exaggeration. By analyzing surface/stylistic features, sentiment features, and contextual features, the authors in [28] were able to accurately identify and classify instances of satire in Arabic texts. These findings highlight the importance of considering various linguistic cues and markers when studying irony in different languages. As stated earlier, satirical articles have certain language characteristics, such as exaggeration, humor, and the use of conflicting terminology, which effectively convey the satirist's message. These linguistic cues presented as textual characteristics may be employed to determine the presence of satire within a given content.

2.2.2. Innovative Techniques

- **Word Embeddings**: word embeddings are useful for capturing the semantic connections between words [30]. The raw data used to train the network are fed into a low-dimensional dense vector. Following enough training, the lexicon's semantics are learned, and a map is created by grouping words with similar semantic connotations [18,22]. Moreover, standalone representations that are independent of context are captured via static word embedding. Word and subword embeddings were applied using the word2vec tool, which provides two models for representation: a continuous bag of words and a continuous skip-gram of words [31]. This application was for the purpose of detecting Arabic satire. Moreover, Arabic FastText was used to extract word embeddings [32]. The deep emoji technique was additionally utilized for the extraction of emotion features [23]. Even though the static word embedding method is effective, especially the embeddings trained on large datasets, it does not account for the meaning of a word in various contexts. Due to the possibility that a word in one domain could have a completely different meaning in another, it did not perform well on domain-specific datasets [33]. Contextual word embedding is, therefore, employed to represent the term in accordance with the context in which it appears [22,34]. Technologies such as ElMo, for example, have successfully addressed this problem, although they need larger datasets, whereas domain-specific datasets do not. It is important to note that in non-contextual jobs, such as studying vector spaces, utilizing static word embedding is occasionally preferred. The computational cost of static word embedding is also significantly lower than that of contextual word

embedding [35]. Contextual word embedding has been utilized in a variety of studies, including [22,34], to identify Arabic sarcasm.

Transformers take natural language text as input and generate a prediction for a classification problem. Its architecture uses an encoder–decoder structure, as shown in Figure 1. The encoder takes an input and converts then into a sequence of continuous representations that are fed to a decoder, which generates a prediction.

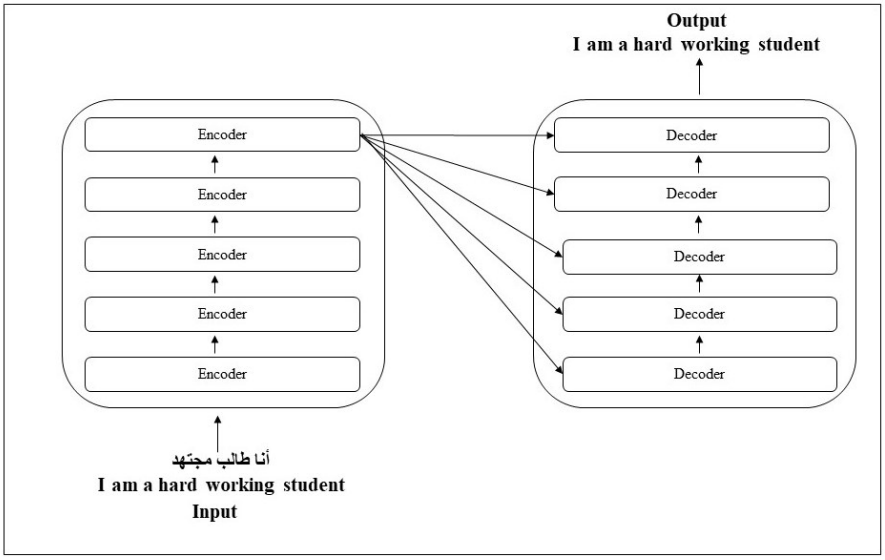


Figure 1. Pretrained transformer model architecture.

According to Vaswani et al. [36], "The Transformer is the first transduction model relying purely on self-attention to calculate representations of its input and output without employing sequence-aligned RNNs or convolution". Using layered self-attention and pointwise, completely linked layers, the transformer conforms with the encoder decoder's overall architectural design. By translating a query and a collection of key-value pairs to an output, the transformer's attention function is calculated. The result is then calculated as a weighted sum of the values, with each value's weight determined by the query's compatibility function with the connected key, where dimension *dk* is of the keys, and dimension *dv* is of the values. Moreover, the transformer performs the attention function in parallel, resulting in *dvdimensional* output values using "multi-head attention" as shown in Equation (1):

$$MultiHead(Q, K, V) = Contact(head_1, ..., head_h)W^O$$
$$, where\ head_i = Attention((QW)_i^Q, (KW)_i^K, (VW)_i^V) \quad (1)$$

Transformers ignore convolutional and recursive layers in favor of multiheaded self-attentional layers as their foundation. They take advantage of the attention mechanism in three different ways. First, it is employed in encoder–decoder attention layers, where the preceding decoder layer generates queries and the encoder output consists of memory keys and values. Second, there are six internal layers in every encoder and decoder. Each of them is made up of two sublayers: positionwise completely linked feedforward networks and multihead self-attention. Transformers are also less expensive in terms of time.

3. Related Work

Classification of Arabic text, especially fake news, has gained much attention in recent years. In the context of satirical text classification, ML and DL models have been investigated with a variety of features, such as textual and contextual features. Recently, pretrained models, such as BERT, were used in this field and gave promising results.

Satirical news classification can be implemented by examining the textual content, to differentiate between satirical and non-satirical content. For that, a classifier was trained on a dataset containing satirical Arabic tweets to detect satirical news was built [28]. It focused on surface features, sentiment features, shifter features, and contextual features. Features were grouped based on prior studies written in non-English languages, and as a result, this classifier reached an accuracy of 72.36%, despite the difficulty of processing Arabic social media texts and the absence of tools to deal with translating words in lexicons.

Some studies have applied classification techniques to detect fake news. A broader perspective that includes Arabic news articles has been adopted [37]. An exploratory analysis was conducted to identify the linguistic properties of satirical Arabic content as a type of fake news. The researchers analyzed satirical articles by searching for specific features: journalistic register (terminology commonly used in journalistic writing), sentiment intensity measures, and subjectivity measures (by analyzing pronouns that denote first-person inflections). Saadany et al. claimed that these lexical features were sufficient to classify satirical Arabic fake news accurately [37]; however, the nature of satirical articles is quite different from news articles.

Machine learning, and more recently deep learning algorithms, have been used to detect satirical text [19,38–40]. A systematic review study conducted by Sarsam et al. [27] using Twitter data found 31 related studies. Algorithms were classified into Adapted Machine learning algorithms (AMLA), such as support vector machine (SVM) and logistic regression (LR), and other customized machine learning algorithms (CMLA), such as a semi-supervised satire identification algorithm and bootstrapping. The study found that AMLA algorithms were widely used in satire detection studies, and SVM was the most efficient one to predict satire on Twitter. Additionally, lexical, pragmatic, frequency, and part-of-speech tagging features improved the performance of these algorithms.

Arabic satire detection has its complications; Abuteir and Elsamani found that the Naive Bayes model using lexical and structural features outperformed random forest and logistic regression with an accuracy of 89.17% for Arabic tweets [21]. Ensemble classifiers were developed to further improve the detection of satirical content [41]; the combination of random forest, logistic regression, and decision tree provided the best performance.

To further improve the detection, BERT was combined with contextual features [42]. The presented approach improved the detection significantly compared to the state of the art. For social media data that are short texts, a hybrid autoencoder-based model, which consists of bidirectional encoder representations from the transformer (BERT), universal sentence encoder (USE), and an unsupervised learning long-short-term-memory-based autoencoder, were proposed [43]. The accuracy reached 92% on one of the used datasets. Other versions of BERT models were used for Arabic satire detection [24,25,34,44], showing promising results.

A recent study that investigated the automation of satire detection from 2010 to 2022 [45] found that deep learning algorithms, such as convolutional neural networks and long short-term memory, performed better than machine learning algorithms. AlHazzani et al. developed an Arabic dataset from Twitters [46]. They applied ML and DL algorithms to study their impact on detecting satirical articles. They found that SVM and BiLSTM provided the best accuracy. Recently, transformer-based architectures have been used, and they performed better than all previous approaches. Another comprehensive study reviewed research works from 2017 to 2022 [1] and pointed to the need for an Arabic satire dataset to be used for such studies, since the few collected datasets are either small or imbalanced. Additionally, the results found that the BERT model is the best-performing model. However, factors can affect these findings, such as corpus size, model parameters, and feature extraction.

As shown in the related work mentioned above, the techniques developed for Arabic satire detection make use of datasets in the form of tweets or headlines, which consist of a small amount of text compared to articles. Additionally, tweets are written informally, which could skew the results because they are more obvious than satirical articles that are

written in a formal journalistic style, making them more difficult to detect. This highlights the novelty of our work.

4. Methodology

Our methodology consists of preprocessing, feature extraction, and the classification model steps as shown in Figure 2. Preprocessing is explained in detail in Section 4.2. Then, features are extracted using n-grams, textual analysis, and word embeddings (Section 4.3). We compiled the first dataset of Arabic satirical articles that is collected as described in Section 4.1. The data are used with ML and DL classification models.

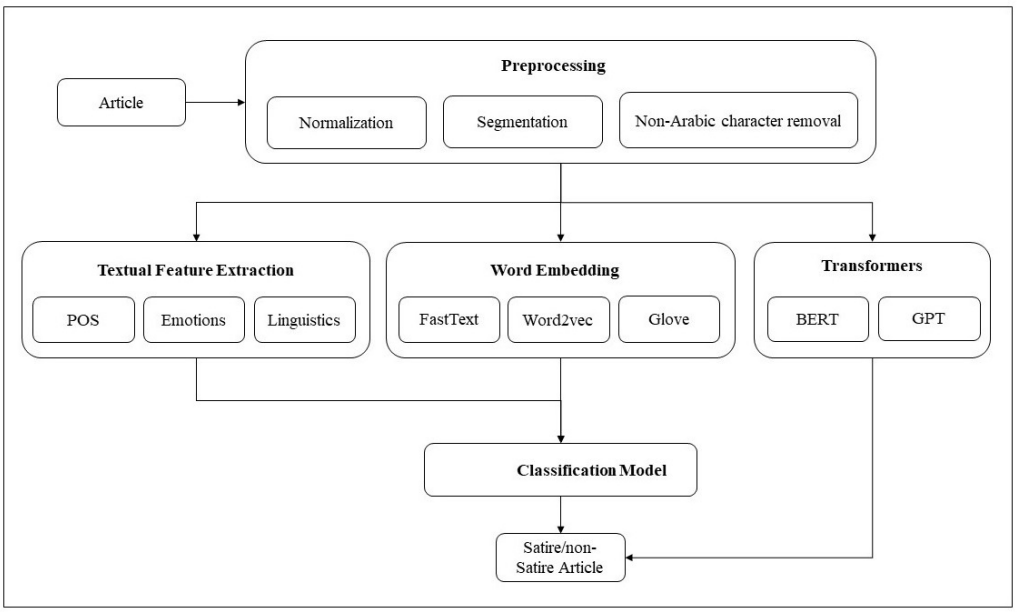


Figure 2. The overall methodology

4.1. Dataset

To investigate the satirical nature of articles in the real world, we compiled a large dataset by manually scrapping through real-world Arabic satirical web platforms. We managed to extract a diverse range of 768 satire from two of the most popular Arabic satire webpages named Alhudood (https://alhudood.net (accessed on 1 September 2023)) and Dkhlak (https://dkhlak.com/ (accessed on 1 September 2023)). These platforms were specifically chosen as they display the satirical articles in a formal journalistic format imitating real news articles. Alhudood has more than 60k Instagram followers and more than 20k Twitter followers, which demonstrates their prominence in the Arabic-speaking world. They explicitly categorize themselves as "satire" platforms, and publish articles weekly. The articles were manually collected from articles that were published from 1 January 2020 to 1 January 2022. We specifically chose to collect satirical articles from explicit real-world satirical news platforms to render the classification model using real-time data. This additionally helps train the model to identify nuances in satirical articles that can be used as markers for satire.

To balance the dataset, we manually collected 768 non-satirical articles from formal newspapers, such as Okaz (https://www.okaz.com.sa/articles (accessed on 1 September 2023)) and Sabq (https://sabq.org/ (accessed on 1 September 2023)). The articles had diverse topics such as politics, economics, religion, and technology. In total, the dataset included 1536 satirical and non-satirical articles. Table 1 shows the statistics of the collected articles; word length is the number of characters in each word and sentence length is the number of words in each sentence. A sample of a satirical article, that is extracted from the satiric platform "alhudood," is shown in Figure 3, which is depicting, in a satiristic manner, a security operation undertaken by police forces that targeted news platforms.

Table 1. Dataset information.

Data	Satire	Non-Satire
No. of Articles	768	768
Avg. no. of sentences	3	3
Avg. no. of characters	453	483
Avg. word length	6.74	6.83
Avg. sentence length	45.52	36.35

بعد أن تناهى إلى مسامعها انتشار تقارير صحفية تحوي أرقاما و حقائق و تتضمن مراجع من شأنها إيصال معلومات صحيحة وتحقيقات قد تكون مهمة، تناقش قضايا تتلمس هموم الشعب، فتحت السلطات المصرية تحقيقا في شبهة وجود موقع إخباري مستقل في مصر. و قال رئيس المخابرات العامة المصرية إن الموقع المشبوه استغل أنه من ضمن الـ ٥٠٠ موقع المحظورين في البلاد "و أن مخبرينا مشغولون بمراقبة ملايين المواطنين وكتابة التقارير والمحتوى الصحفي للمواقع الإعلامية غير المحظورة، لينشر تقريرا يتناول مسائل عائلية تخص سيادة الرئيس السيسي و عائلته مثل جهاز المخابرات و الإعلام و مستقبل البلاد."

Figure 3. Example of satirical text in Arabic.

4.2. Data Preprocessing

Due to the humorous nature of satirical articles, many of the collected articles contained punctuation and non-Arabic characters. Moreover, since satirical articles do not comply with the journalistic genre found in formal news articles, they tend to contain many misplaced Arabic glitches. For that, we performed the following operations:

- Tokenization, which is a method of separating a piece of text into smaller chunks called tokens.
- Normalized elongated word by removing repetition of three or more characters
- Normalized the three Arabic letters: *alef*, *alef maqsoura*, and *ta-marbouta*.
- Removed diacritics and punctuation marks.
- Removed non-Arabic characters

By performing these preprocessing steps, text data are transformed into a clean, consistent, and manageable format that facilitates effective text classification. Preprocessing helps remove noise, standardize the representation of text, reduce dimensionality, and enhance the relevance of features, enabling models to better capture meaningful patterns, and improve overall classification performance. For all of the previous preprocessing steps, we used Tashaphyne Arabic light stemmer (https://pypi.org/project/Tashaphyne/ (accessed on 1 September 2023)), which provides several Arabic NLP tasks including stemming, segmentation, and normalization.

4.3. Feature Extractions

This section provides a thorough explanation of the fundamental feature extraction techniques used to construct the empirical models that were adopted for our tests: *n-grams*, which is a group of n co nsecutive text, that can be words, numbers, symbols, or punctuation, *textual features*, consisting of elements or components of the text, such as nouns, verbs, and pronouns, which are used to create meaning, and *word embeddings*, which convert word into vector representation used in text analysis.

4.3.1. N-Grams

Different ranges of n-grams were investigated, including unigram, bigram, and trigram. Note that all of the n-gram features were normalized by the Term Frequency-Inverse Document Frequency (TF-IDF) method to determine their importance in the dataset. TD-IDF assigns weight to each word in a document based on its term frequency, and its corresponding document frequency. The words with higher weight scores are indications of the most significant ones in the document. By incorporating TF-IDF as a feature representation, text classification models can effectively capture the importance and distribution of

words in documents. TF-IDF highlights discriminative terms, reduces dimensionality, and provides a robust, informative representation that improves the accuracy and performance of text classification models.

4.3.2. Textual Features

A set of textual features found to be useful for classifying Arabic satirical articles [47,48] were selected. The primary reason for selecting these textual features was to examine the impact of each textual feature category on satirical content identification, in terms of the emotional value, word use (linguistics), and word structure (POS). Table 2 summarizes the textual categories extracted from satirical articles. We describe each one in detail as follows:

- **Emotions**: The most well-known list, frequently referred to as "The Big Six", was utilized by Ekman et al. [49] in their investigation into the universal detection of emotion from facial expression. The list contained the most widely acknowledged candidates for fundamental emotions, including joy, sadness, fear, surprise, anger, and disgust. This emotion list was translated to Arabic in the study by Saad [50].
- **Part of speech (POS)**: These refer to nouns, verbs, adverbs, adjectives, prepositions, determiners, pronouns, conjunctions, and proper nouns. We used the Farasa [51] tool to extract each word's POS tag.
- **Linguistics**: Linguistic features 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 recent years, there has been an increasing amount of literature investigating authors' writing styles to identify unique features associated with their writing and to identify certain characteristics [52–56]. The set of linguistic markers investigated in this study, as described in Table 2, are assurance, negations, justification, intensifiers, hedges, illustrations, temporal, spatial, superlative, exceptions, and oppositions. These linguistics were extracted following the approach of Himdi et al. [57].

Table 2. All textual features extracted.

POS		
A. Content Words		
Nouns	Verbs	Adjectives
Adverbs	Proper Nouns	
B. Function Words		
Conjunctions	Prepositions	Pronouns
Particles	Determiners	
Emotion		
Anger	Sad	Fear
Joy	Disgust	Surprise
Linguistic		
Assurance	Negations	Illustration
Intensifier	Hedges	Temporal
Spatial	Exclusion	Opposition
Justification		

4.3.3. Word Embeddings

Word embeddings have been used in different NLP tasks, including, but not limited to, automatic transcription detection, error detection, and speech recognition [58]. It shows an improvement when used with typical NLP tasks as well as other other semantic and syntactic similarity tasks [59]. The following are details of the word embeddings used in this study, which have been found to be effective in related satire classification models [32,60,61].

- **FastText**: is a library developed by Facebook that allows for efficient text classification and representation learning. It is designed to work with language models that are capable of learning from a large corpus of text data. It uses a technique called n-grams, which breaks down a sentence into its component words and then looks at the frequency of the words used to determine the overall meaning of the sentence. This technique allows FastText to accurately identify words and their meanings in any given sentence, no matter how long or short. FastText has seen great success in NLP processing tasks, such as sentiment analysis, text summarization, and entity recognition [62]. Additionally, FastText has pretrained versions for several languages. Since there were only Arabic datasets available for the experiment, FastText's 300-dimension Arabic pretrained vectors were used in this study [63].
- **Word2vec**: is a predictive model that trains by attempting to predict a target word given a context (CBOW method) or by using the context words from the target word to predict the context words (skip-gram method) [64]. It employs trainable embedding weights to map words to their respective embeddings, which are used to assist the model in making predictions. As the loss function for training the model is proportional to the accuracy of the model's predictions, training the model to make more accurate predictions will result in more accurate embeddings. It used a neural network model to generate embeddings for each word [65]. AraVec pretrained embedding on a Twitter dataset employing CBOW and 300 embedding dimensions was used [65].
- **GloVe**: employs matrix factorization techniques applied to the word-context matrix. First, it creates a large matrix of (words × context) co-occurrence information. For example, for each "word" (the rows), it counts how frequently (matrix values) this word appears in a given "context" (the columns) in a large corpus. The number of "contexts" is essentially combinatorial in size, so it would be very large. Therefore, we factorize this matrix to produce a lower-dimensional (word × features) matrix, where each row represents each word as a vector. This is typically achieved by minimizing "reconstruction loss". This loss seeks to identify the lower-dimensional representations that can explain the majority of the variance in high-dimensional data, [66]. For usability in Arabic, there is just one pretrained GloVe embedding currently accessible online, which is a 256-dimensional pretrained GloVe vector word embedding [67].

4.4. Classification Models

4.4.1. Machine Learning

Traditional ML algorithms, such as Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), and Random Forest (RF), have proven to be advantageous in several text classification projects [68–70], particularly Arabic text classification [71,72]. Due to the fact that Naive Bayes is based on Bayes' theorem and the assumption of conditional independence between features, it is an effective classification algorithm [68]. On the other hand, SVM works well with small-to-medium-sized datasets [69], LR provides insights into the relationship between features and class probabilities [70], and RF combines multiple decision trees to make predictions. A description of each algorithm in detail is as follows:

- **Naive Bayes (NB)**: This classifier is composed of a number of algorithms that are based on the Bayes Theorem and assumes independence of the attributes. It is based on estimates, in which the model adjusts its probability table using the training data and predicts new observations by estimating the class probability based on its feature values. The small amount of training data needed by NB results in storage space savings. It also yields quicker results and is not sensitive to missing data [73].
- **Support Vector Machine (SVM)**: is a supervised machine learning model that classifies input data based on dimensional surfaces by finding the maximum separating hyperplane between different classes [74]. SVM offers data analysis for both classification and regression analysis. After that, it chooses a boundary that maximizes the distance between neighboring members of various classes [75]. SVM offers the advantage of resolving overfitting in high-dimensional spaces. As a result, by choosing

a plot from a large selection, it can simulate non-linear acceptable boundaries. It is widely used in text classification projects [76,77].

- **Logistic Regression (LR)**: is a classifier that establishes a link between features and likelihood of the outcome [78]. It is based on the logistic function, an S-shaped curve that maps real value numbers to values between 0 and 1 [79]. LR is reliable for classifying problems [80], and it prevents overfitting.
- **Random Forest (RF)**: this classification algorithm uses a decision tree model that is based on bootstrap aggregation techniques, called bagging [81], which is an ensemble method that combines the predictions from several ML algorithms. Bagging also reduces high variances that can be produced by the algorithm, which is due to its sensitivity to the training data. RF is scalable and robust to outliers.

4.4.2. Deep Learning

CNN is an extension of an artificial neural network (ANN). Similar to ANN with extra layers, such as the convolutional layer and pooling layer, which are used to extract features and downsample the input, thus reducing its computation. On the other hand, Bi-LSTM, which stands for bidirectional long short-term memory networks, consists of two LSM networks, which work in opposite directions. These DL models have been used successfully to classify several Arabic classification tasks [82–84].

- **CNN**: is one of the most popular neural networks that consist of an input layer, hidden layer, and output layer. It uses a convolution layer that is used to transform the input data into an easier-processed form. Additionally, the pooling layer is also used to reduce the input dimensions.
- **Bi-LSTM**: is a bidirectional recurrent neural network (RNN) that takes two long short-term memory (LSTM) neural networks. The first time process the input sequence in the forward direction, and the second time process the input in the backward direction. Thus, improves the model learning and produces better accuracy.
- **CNN & Bi-LSTM**: is a hybrid model that combines CNN and Bi-LSTM. CNN is used to reduce the input dimensions. Then, the output is fed into the LSTM layer. This model uses the convolution layer to extract local features, and the LSTM layer uses the ordering of those features to learn about the text ordering of the input.

4.4.3. Transformers

This section provides a summary of the top text classification transformer models that perform well on texts of various lengths since they already include word embeddings.

- **BERT** (Bidirectional Encoder Representations from Transformers): is a powerful language model transformer that has been shown to achieve state-of-the-art performance on a variety of natural language processing tasks [85]. By examining relationships in sequential data, the neural network that helps the transformer architecture to comprehend context and meaning. This information is the words in a sentence when natural language processing (NLP) is used. Encoder–decoder architecture is used in this system. An input sequence's characteristics are extracted by the encoder on the left side of the architecture, which is then used by the decoder on the right to create the output sequence.
- **GPT** (Generative Pretrained Transformer): is a language model comprising an encoder and a decoder as part of a transformer architecture. It has been applied to NLP tasks. As humans, they generate messages, respond to inquiries, and produce photographs and movies. Antoun et al. [86] proposed the generative pretrained language model AraGPT2. The model utilizes a self-attention mechanism to identify long-term relationships between sequences over time. The model is trained using a collection of texts, the majority of which are written in Modern Standard Arabic (MSA). The AraGPT2 program has been extensively used in NLP projects such as text creation [87]. In contrast to regular transformers, it extends the self-attention block with a second normalizing layer, which makes it unique among transformers. Both BERT and GPT

excel at text classification due to their abilities to capture contextual information. BERT's bidirectional approach allows it to understand word meaning in relation to both preceding and succeeding contexts, while GPT's generative nature enables it to generate coherent text based on the preceding context. These attributes contribute to their effectiveness in text classification tasks, as they provide models with a deeper understanding of language, context, and semantic relationships within texts [88].

5. Experiments

As mentioned above, a thorough examination is implemented in this work for the task of Arabic satire classification. We conducted different experiments to evaluate the performance of each algorithm with several features, as shown in Table 3.

Table 3. Summary of feature extraction techniques.

Technique	Description
N-Grams	Contiguous sequences of n tokens that can be words, characters, or other units extracted from a given text.
Textual Features	Refer to various characteristics or properties of text that can be extracted or analyzed to gain insights into the text
Word Embeddings	Capture semantic relationships between words by representing each word as a point in the embedding space, where similar words are closer to each other.

5.1. Model Compilation

In this section, we review the models used in the experiments. For the ML models, the default parameters were adjusted to ensure equality between all tested models (Table 4). On the other hand, DL parameters had to be fine-tuned to achieve robust results. To compare the performance of each model using various feature extraction strategies equally, we ensured that all the models were equitably adjusted.

Table 4. Machine learning classifier parameters.

Model	Parameters
SVM	batchSize 100 kernel linear
NB	batchSize 100
LR	batchSize 100, maxBoosting-Iterations 500
RF	batchSize 100, bagging with num-Iteraions 100, and number of trees 100

Fine-tuning is performed by adding a fully connected layer on top of the pretrained model. Each model is fine-tuned for the specific task. We unify the essential parameters (Table 5) when creating deep learning models, which are as follows:

- The batch size is the number of examples to be taken into account prior to updating the model's parameters. Batch sizes of 32, 64, and 128 were examined in this study.
- The number of epochs is determined by how frequently the algorithm will execute the training dataset. Here, we experimented with epoch numbers ranging from 1 to 15. To avoid wasting time and storage, we used the stop accuracy method. It allowed us to terminate the training when we reached the highest accuracy to make up for the discrepancy between the loss function and the updating of model parameters. The Adam optimizer was adjusted with a learning rate of 0.001.
- Dropout improves the model's generalization and reduces the likelihood of overfitting. To constrain the weight of layers, the dropout rate was modified to 0.2.
- The classifier is the final layer that converts all the input into predicted classes. Thus, choosing this layer included Conv1D layer, MaxPooling1D layer, and Dense layer with a sigmoid activation function for binary classification.

Table 5. Unified parameters for deep learning classifiers.

Parameter	Value
Batch size	32, 64, and 128
Epochs	range 1 to 15
Dropout Rate	0.2

A detailed description of the compilation of each deep learning and transformer model can be found below.

- **CNN**: The architecture consists of several layers. First, the Conv1D layer, which approximately applies 64 filters with a kernel size of 3 to the input data, and ReLU was added for an applied activation function. Second, the MaxPooling1D layer, which was used to perform max pooling with a pool size of two. Third, the Conv1D layer, which applied 64 filters with a kernel size of three to the output of the previous layer. MaxPooling1D layer was then applied to perform max pooling with a pool size of two. Then, the Flatten layer, which flattens the output of the previous layer, was applied. Finally, the dense layer, which has two neurons with a sigmoid activation function, outputs a probability distribution over the output classes.
- **Bi-LSTM**: its architecture uses the Keras library. The model has a single bidirectional-LSTM layer with 64 units, followed by a dense layer with a sigmoid activation function that outputs two values for binary classification. The loss function used is categorical cross-entropy, and the optimizer was Adam. The model was trained for two epochs; the input shape was 768.1, which is the size of the BERT embeddings after reshaping.
- **CNN & Bi-LSTM**: this model's architecture combines both the CNN and Bi-LSTM layers: A bidirectional LSTM layer with 64 units and a 1D convolutional layer with 64 filters and a kernel size of three extract local patterns from the sequence. Then, the MaxPooling1D layer with pool size two was applied to reduce the dimensionality of the feature maps. The flattening layer converts the 3D tensor output from the previous layer into a 1D tensor. Last, there is the dense layer with two units and the sigmoid activation function, which outputs the probability distribution over the classes.
- **BERT** (Bidirectional Encoder Representations from Transformers): BERT employs bidirectional self-attention transformers to capture both short- and long-term contextual dependencies in the input text. We use AraBERT [86], which has a vocabulary capacity of 64,000 words, 12 attention heads, 12 hidden layers, 768 hidden sizes, a total of 110 M parameters, and 512 maximum sequence lengths. It was trained using a dataset of 3B Arabic words. Specifically, we used the available version "paraphrase-multilingual-mpnet-base-v2 model (https://huggingface.co/ (accessed on 1 September 2023))", which is a pretrained BERT model that is designed to encode multilingual texts into high-quality embeddings. Notably, the Adam optimizer with a learning rate of 1e-4, batch size of 512, and sequence length of 128 was utilized.
- **GPT** (Generative preTrained Transformer): ArAGPT2 is the largest publicly accessible collection of filtered Arabic corpora was used to train the model [89]. The complexity metric, which assesses how effectively a probability model predicts a sample, was used to assess the model. The model is trained on 77 GB of Arabic text. ARAGPT2 is available in four training size variants: base, medium, large, and mega; the smallest model, base, has the same dimensions as ARABERT-base. This enables it to be accessible to a greater number of researchers. Larger model variants (medium, large, and xlarge) provide enhanced performance but are more difficult to fine-tune and computationally costly. The ARAGPT2- detector is based on a pretrained ARAELECTRA model that was refined using a synthetically generated dataset. In this study, we used the ARAGPT2 base model; which has a batch size of 1792, a learning rate of 1.27e-3, LAMB optimizer, 12 heads and 12 layers, and a training size of 135 M.

5.2. Evaluation Metrics

Our experiment evaluated the impact of different feature extraction techniques on a selected set of ML, DL, and transformer models that are well known for their performance in text classification problems. The performance of each classifier was evaluated based on average precision (P), average recall (R), and average F1-score (F1-score). These were computed using the confusion matrix, which was created by applying the classification model. The precision, recall, and F1-score measures were computed using Equations (2), (3), and (4), respectively:

$$\text{Precision (P)} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{TrueNegative}} \tag{2}$$

$$\text{Recall (R)} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \tag{3}$$

$$\text{F1-Measure (F1)} = \frac{2.\text{P.R}}{\text{P} + \text{R}} \tag{4}$$

We chose to calculate the results in the F1-score, as it is used when false negatives and false positives are more significant than true positives and true negatives, as in accuracy. In the majority of real-world classification problems, class distribution is unbalanced [90]; therefore, the F1-score is a more appropriate metric for evaluating our model.

Models were developed using Python version 3.7 using the Tensorflow Keras sequential API. All experiments were carried out on a system with a dual-core Intel Core i7 processor and 11 GB of RAM, running on a Macintosh operating system with a 64-bit processor and access to an NVidia K80 GPU kernel. The dataset included a total of 1536 combined satirical and non-satirical articles, specifically, 768 satirical and 768 non-satirical articles. Furthermore, 70% training and 30% testing ratios were conducted.

5.3. Results and Discussion

Based on data collection, preprocessing, feature extraction, and model training, it was observed that the dataset was well-balanced with a moderate number of samples. The data were preprocessed by scaling and splitting into training and validation sets. Various machine learning models were trained, including SVM, LR, NB, and RF, and their performances were evaluated. Additionally, a CNN, Bi-LSTM, and a hybrid CNN & BiLSTM neural network, along with transformers BERT and GPT, were tested and evaluated.

First, n-grams were used to train the models as shown in Tables 6 and 7. We observes that both ML and DL models performed better with unigram compared to bigram and trigram. Of both models, DL with unigram reached the highest performance. The highest F1-score achieved (94.1%) was obtained with the CNN model.

Then, three textual feature categories were evaluated: emotions, part-of-speech, linguistics, and the combination of them. We used them with ML and DL models to determine the best classification model for satirical articles. The results are shown in Tables 8 and 9. ML models gave promising results; SVM and RF were the best models with linguistics features and with all features combined. The highest F1-score (72.7%) was achieved with RF with all features combined. However, DL surprisingly performed worst with textual features. As the highest F1-score achieved was 64.2% when linguistic features were used with the Bi-LSTM model.

Table 6. results of the machine Learning Algorithms with n-grams [1].

ML		**NB**			**SVM**			**LR**			**RF**	
Class	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Uni	68.0	68.1	68.0	62.2	64.9	62.2	70.0	70.2	69.9	68.9	69.2	68.9
Bi	57.3	60.5	53.4	49.7	50.0	39.4	59.0	62.7	59.0	57.7	61.0	54.0
Tri	49.9	49.8	44.9	49.3	24.6	49.3	49.9	49.2	41.0	49.8	49.6	47.7

[1] Uni = Unigram, Bi = Bigram, Tri = Trigram.

Table 7. Results of the deep learning algorithms with n-grams [1].

DL	CNN			Bi-LSTM			CNN & Bi-LSTM		
Class	P	R	F1	P	R	F1	P	R	F1
Uni	94.4	95.2	94.1	94.2	91.4	93.7	89.1	97.3	93.5
Bi	63.1	99.1	77.4	70.1	91.1	79.0	74.2	99.0	85.3
Tri	70.1	64.2	67.6	53.1	92.7	67.3	99.0	30.2	46.3

[1] Uni = Unigram, Bi = Bigram, Tri = Trigram.

Table 8. Results of machine learning with textual features [1].

ML	NB			SVM			LR			RF		
Class	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Emo	41.3	45.2	46.5	55.6	55.3	55.3	50.8	45.6	48.8	59.2	59.3	59.2
POS	59.9	59.6	59.9	67.6	67.6	67.6	62.9	63.1	63.0	60.4	60.4	60.3
Ling	61.3	60.2	58.7	60.1	62.4	62.1	63.1	62.9	62.8	62.5	62.5	62.5
Comb	67.5	67.5	68.9	67.0	67.0	67.0	63.7	63.8	64.1	72.7	73.5	72.6

[1] Emo = Emotions, POS = Part of speech, Ling = Linguistics, Comb = POS + Emotions + Linguistics.

Table 9. Results of deep learning with textual features [1].

DL	CNN			Bi-LSTM			CNN & B-LSTM		
Class	P	R	F1	P	R	F1	P	R	F1
Emo	50.2	43.8	42.1	41.5	37.1	34.9	53.2	45.7	43.6
POS	62.3	55.2	54.8	58.7	46.7	42.0	59.4	54.3	54.4
Ling	61.1	51.4	49.6	64.9	63.8	64.2	64.8	51.4	48.1
Comb	60.9	57.1	47.5	39.7	37.1	27.8	63.1	60.0	60.5

[1] Emo = Emotions, POS = Part of Speech, Ling = Linguistics, Comb = POS + Emotions + Linguistics.

Word embeddings, Word2vec, FastText, and GloVe were used to train ML and DL models. Notably, Word2Vec and GloVe performed better than FastText, as shown in Tables 10 and 11. The best-performing ML model was SVM with GloVe embeddings, which reached an F1-score of 91%. The highest F1-score was achieved with CNN & Bi-LSTM when W2Vec embedding was used, reaching 94%. Under the same word embeddings, we found that DL models and ML models performed very similarly.

Table 10. Results of machine learning algorithms with word embeddings [1].

ML	NB			SVM			LR			RF		
Class	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Fast	76.0	78.0	79.0	81.0	80.0	82.0	88.0	87.0	89.0	85.0	85.6	87.0
W2V	82.0	83.0	81.0	82.0	84.0	82.0	78.0	79.0	79.0	91.0	87.5	89.0
GloVe	86.0	91.0	88.0	89.0	95.0	91.0	89.0	92.0	90.0	86.0	92.0	87.5

[1] Fast = FastText, W2V= Word2Vec.

Table 11. Results of deep learning algorithms with word embeddings [1].

DL	CNN			Bi-LSTM			CNN & Bi-LSTM		
Class	P	R	F1	P	R	F1	P	R	F1
Fast	89.8	89.8	89.8	86.2	86.2	86.2	84.2.8	84.2	84.2
W2V	86.1	96.0	91.0	83.1	89.0	86.0	92.0	96.0	94.0
GloVe	86.0	91.0	89.0	73.0	95.0	83.0	96.0	90.0	91.0

[1] Fast = FastText, W2V= Word2Vec.

Finally, BERT and GPT transformers were utilized, since they are among the most prevalent classification models in NLP currently. BERT performed better than GPT with an F1-score of 90%.

Generally, the results unveil a number of interesting findings of the benefits of feature extraction methods on Arabic satire classification. It is evident that combining word embeddings with ML or DL models does not result in an impressive performance, as it was consistently outperformed by unigram-based learned models. The reason for this is that simple and direct averaging in word embeddings does not take into account the order of the words, which is captured more precisely by n-gram models.

In addition, we noticed that GloVe and Word2Vec was the most effective word embedding for feature extraction when combined with ML and DL models. However, FasText produced comparable results, leading us to conclude that there is no specific form of word embedding that tops all others. This may be because AracVec, GloVe, and FastText were all trained on Arabic data from Twitter, Wikipedia, and multiple Arabic corpora that contain the plurality of formal Modern Arabic words, and are well suited to satirical articles written in a formal codified form. Although BERT outperformed GPT in terms of transformers, we utilized the basic model of GPT2, which may have provided superior performance if the superior model version had been utilized. Across all tests, the transformers BERT and GPT achieved high results of 95% and 85%, respectively.

For textual features-built models, the best model reached 72.6%, which is in line with the study by Kourai et al. [28], who achieved an F1-score of 73% with ML models. As for DL models, surprisingly, we found poor performance, i.e., 60%, which is in line with [91], which used sentiment lexicons to classify satirical content reaching a score of 46%. The approach of analyzing the textual content either as extracting textual features or sentiment features gave poor performances. This could be because satirical articles are highly dependent on the audience the articles are delivered to in terms of culture, gender, and context. Consequently, the textual content of satirical articles varies depending on their audience, resulting in diverse textual features that might have not been extracted in the feature extraction phase.

Notably, from our comprehensive study, the highest F1-score (95.0%) was achieved with the BERT model. However, the second highest F1-score was achieved with the CNN model when the unigram was applied, which affirmed the efficacy of n-grams in the classification of Arabic satirical articles. This could be interpreted as the greater the n-grams length is, the less accurate the classification becomes. In other words, smaller n-grams tend to emphasize one unit of the text at a time, compared to higher n-grams, which take more than one unit at a time. This finding was also supported in other studies of text classification tasks [92]. Results of BERT and GPT models in Table 12.

Table 12. Results of BERT and GPT models.

BERT			GPT		
P	**R**	**F1**	**P**	**R**	**F1**
94.0	97.0	95.0	82.0	89.1	85.0

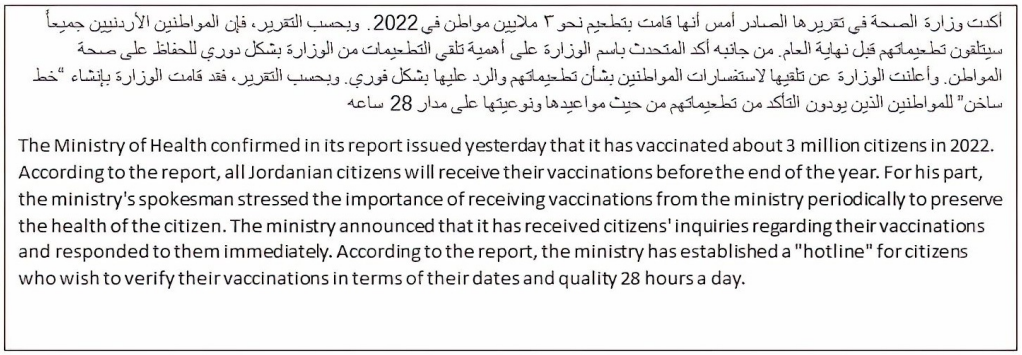
5.4. Error Analysis

In this section, we analyze the misclassified satirical articles of the best Arabic satire classification model. The best model was the BERT model. When further analyzing the articles that were correctly predicted as true positives, we found that 1010 out of 1073 articles in the testing set were correctly predicted. Notably, we found explicit satirical content in these articles, portrayed as adjectives and intensifiers.

However, 63 satirical articles were wrongly classified as non-satire, containing many journalistic registers imitating formal news articles, and had a satirical sense ingrained within the content, thereby presenting implicit satire. Implicit satire is challenging to unravel, since satirical and non-satirical articles share formal terms used in news articles. However, these terms have differing meanings depending on the article's overall tone. For example, the term "highness" occurred frequently in both non-satirical and satirical articles. In the former, it refers to royal figures, and in the latter, it is used to refer to objects of higher

superiors with a mocking tone, such as "his Highness the beloved cat". Figure 4 shows an example of such a misclassified satirical article, which is a form of implicit satire. It is written in a formal journalistic form; however, it contains one phrase "hotline", mocking the lack of services.

Due to BERT's contextualized embeddings and fine-tuning process, satirical content, even when written in a journalistic manner, was effectively analyzed and identified, and thus, classified. This unique feature found in BERT's bidirectional nature and contextualized word embeddings aided in understanding the context and capturing the subtleties present in satirical texts. However, the low accuracy of textual features could be explained by the lack of satirical content in the form of emotions or linguistics that could be used to train the model. This is understandable given that the articles were written in a journalistic style with minimal satirical content.


أكدت وزارة الصحة في تقريرها الصادر أمس أنها قامت بتطعيم نحو ٣ ملايين مواطن في 2022. وبحسب التقرير، فإن المواطنين الأردنيين جميعاً سيتلقون تطعيماتهم قبل نهاية العام. من جانبه أكد المتحدث باسم الوزارة على أهمية تلقي التطعيمات من الوزارة بشكل دوري للحفاظ على صحة المواطن. وأعلنت الوزارة عن تلقيها لاستفسارات المواطنين بشأن تطعيماتهم والرد عليها بشكل فوري. وبحسب التقرير، فقد قامت الوزارة بإنشاء "خط ساخن" للمواطنين الذين يودون التأكد من تطعيماتهم من حيث مواعيدها ونوعيتها على مدار 28 ساعه

The Ministry of Health confirmed in its report issued yesterday that it has vaccinated about 3 million citizens in 2022. According to the report, all Jordanian citizens will receive their vaccinations before the end of the year. For his part, the ministry's spokesman stressed the importance of receiving vaccinations from the ministry periodically to preserve the health of the citizen. The ministry announced that it has received citizens' inquiries regarding their vaccinations and responded to them immediately. According to the report, the ministry has established a "hotline" for citizens who wish to verify their vaccinations in terms of their dates and quality 28 hours a day.

Figure 4. Sample of misclassified satirical articles.

5.5. Limitations

The dataset contained 768 satirical and 768 non-satirical articles, which is considered a moderate number to pertain to a broader analysis. Additionally, the lack of accessible Arabic satire platforms was a challenging factor that contributed to the relatively small number of articles collected in the dataset. Because deep learning models require a large amount of training data, this shortcoming may have impacted their performance. Moreover, the presence of implicit satirical content in the articles caused a challenge, as there are no satiric indicators that could be captured.

5.6. Satire Lexical Density

To investigate the influence of each textual feature category on the formation of satirical articles, we computed the lexical density of each textual feature. This textual analysis step provides insight into the nature of Arabic satirical articles. The lexical densities of all textual feature sets of POS, emotion, and linguistics are presented in Table 13. The lexical density of each feature was computed using Equation (5):

$$\text{Lexical Density (L)} = \frac{(\text{Total number of occurrences of each feature in a class}) \times 100}{\text{Total number of words in the whole class}} \quad (5)$$

According to Table 13, in comparison to non-satirical articles, satirical articles showed an increase in the majority of textual feature categories, focusing on the POS feature sets for nouns, verbs, prepositions, adverbs, proper nouns, and conjunctions. This might be because satirical articles frequently included "made-up" events, which required the creation of proper nouns to support them. Conjunctions and prepositions were necessary for the formation of an artifice article. This result is consistent with that of Rubin et al. [93], who found that shallow syntax POS was a strong indicator of satire.

We found an increase in most satirical articles' textual feature categories compared to non-satire, with attention to nouns, verbs, prepositions, adverbs, proper nouns, and conjunctions in the POS feature sets. This might be due to the fact that satirical articles were fused with many "made-up" events, which involved crafting proper nouns to aid these made-up events. Additionally, prepositions and conjunctions were required for an artifice article. This finding is in line with the finding of [93], where shallow syntax (POS) was highly indicative of the presence of satire.

Table 13. Satire_Nonsatire dataset—lexical densities.

Class	Satire	Non-Satire
Nouns	28.6	29.4
Verbs	5.57	6.65
Prepositions	9.72	9.99
Determiners	16.5	15.5
Interjections	0.01	0.04
Adverbs	0.22	0.29
Adjectives	6.30	5.77
Conjunctions	5.06	5.9
Proper nouns	4.6	5.2
Pronouns	6.10	2.33
Anger	0.08	0.096
Sadness	0.04	0.033
Fear	0.06	0.05
Joy	0.12	0.133
Disgust	0.003	0.015
Surprise	0.02	0.032
Assurances	0.04	0.07
Negations	0.14	0.277
Illustrations	0.06	0.05
Intensifiers	0.03	0.14
Hedges	0.03	0.05
Justifications	0.09	0.04
Temporal	0.08	0.08
Spatial	0.07	0.08
Exclusive	0.018	0.02
Oppositions	0.03	0.18

Satirical articles also used fewer adverbs and determiners. The emphasis in these articles was on the creation of events rather than fictitious adjectives, which may be because they covered a wide range of fictitious topics. Determiners are frequently employed in non-satire to refer to well-known individuals by their appropriate titles. For instance, the Arabic term for a prince is "Al-Amir". Nevertheless, some satirical articles mockingly referred to the same prince as "our friend", which caused a decrease in the use of determiners. On the other hand, satire articles contained fewer adjectives and determiners. This may be due to the fact that these articles covered a variety of fictitious topics, so the emphasis was on the construction of events rather than fictitious adjectives. In non-satire, determiners are typically used to refer to prominent figures by their proper titles.

Similar to how positivity, fear, and sadness were less frequently utilized in satirical articles than anger, joy, disgust, surprise, and other negative emotions. The humorous mocking aspect of the satirical articles, which may have been used instead of anger and emotional phrases to change the reader's perspective toward a hateful topic, is depicted by these emotions, as one could expect. Our results are consistent with those of [93], which discovered that negative semantic orientations enhanced the model's performance to 83%.

In a similar manner, anger, joy, disgust, surprise, and negativity were more commonly used in satire articles compared to positivity, fear, and sadness. As anticipated, these emotions are geared to portray the humorous mocking nature of the satire articles, which may have included angry emotional terms to shift the reader's perspective toward a hateful topic.

Our finding is in line with [93] which found that negative semantic orientations improved the model's performance to reach 83%.

Intensifiers, superlatives, oppositions, negations, and hedges all appeared to be on the rise. The results of [28], which identified exaggeration words in terms of intensifiers and opposition as indicating satire content, are compatible with these subfeatures. Surprisingly, satire articles used justification and illustration phrases less frequently. It is conceivable that this is because these types of articles are typically used for criticism and create mockery with humorous elements [94]. This means that because the terms are obviously satirical, no justification is needed to persuade the reader that the substance is legitimate. Comparatively, it is not required to use illustration terms because the articles are based on fabricated events. The "call to action" elements in the satire articles would be distinctive in that they would include imperative verbs, intensifiers, and superlatives.

We observed an increase in intensifiers, superlatives, oppositions, negations, and hedges. These subfeatures are consistent with the findings of [28], which identified exaggeration words in terms of intensifiers and opposition as identifying satirical content. Surprisingly, justification and illustration terms were used less frequently in satirical articles. It is conceivable that this is because these types of articles are typically aimed at criticism and create mockery with humorous elements [94]. This indicates that no justification terms are required to convince the reader that the content is reasonable, since they are clearly satirical. Comparatively, since the articles are based on fictitious events, it is not necessary to include illustration terms. Uniquely, the "call to action" components in the satirical articles would consist of imperative verbs, intensifiers, and superlatives, as this analysis demonstrates.

6. Model Development

To make use of our best-performing model compiled with BERT, we developed a Python script to deploy the model as a free platform. We used Django to deploy our model as a back-end and designed a user-friendly interface as the front end (Figure 5). The main feature of this platform is to provide the user with a satirical classification of mixed articles. The function works by inputting a folder of unorganized articles (satire and non-satire) and the model outputs a folder that includes two text documents labeled "satire" and "non-satire", with each article placed in its predicted class (i.e., a folder), as shown in Figure 6. The model's code and platform can be found on GitHub (https://github.com/Noza1234?tab=projects (accessed on 1 September 2023)).

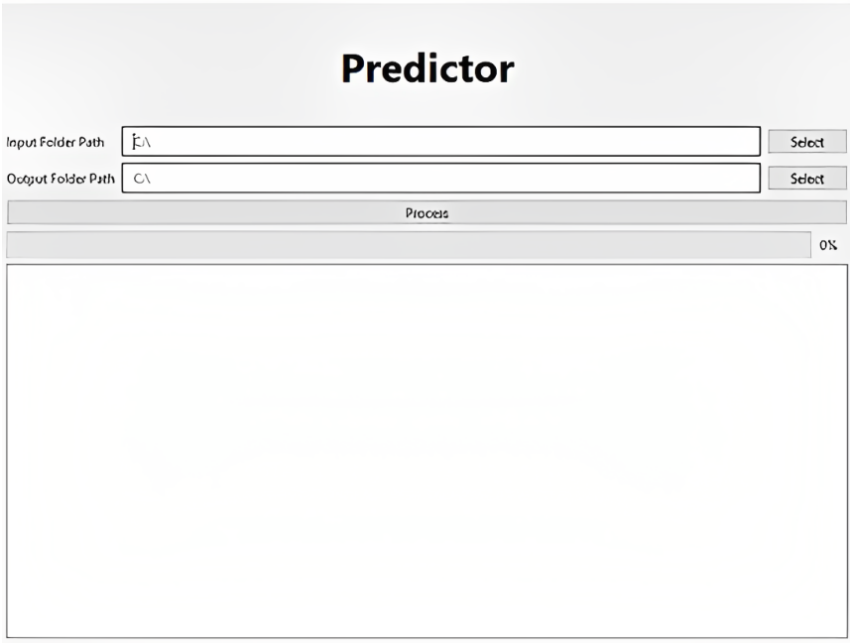


Figure 5. Developed model.

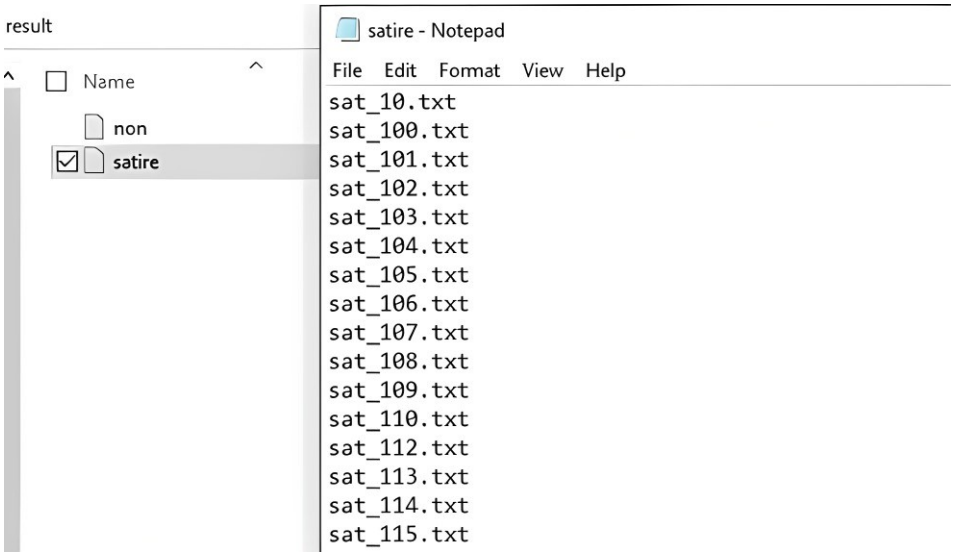


Figure 6. Output of the developed model.

7. Conclusions and Future Work

Arabic satire classification has become an influential topic with the rise of social media. We conducted a comprehensive empirical study to compile the Arabic satirical articles classification model using several feature extraction methods that included n-grams, textual features, and word embedding models with several ML, DL, and transformers. While there have been satire classification projects that investigated textual content found in social media, our study is the first that focuses on Arabic satirical articles written in a formal journalistic manner. To dispel any biased content, we collected the satirical articles from real-world satirical news platforms that explicitly follow a journalistic format. Though the dataset used in this study is relatively moderate, several findings were well-founded. We found that BERT reached an F1-score of 95%, outperforming other traditional feature extraction methods with ML and DL algorithms. On the other hand, traditional methods such as n-grams, specifically unigrams, combined with CNN reached high F1-scores of 94.1%. This indicates the validity of the use of traditional methods closely performing with new methods such as transformers. Additionally, the use of unigrams outperformed bigrams and trigrams, which we provided a justification for such insights. We also investigated and detailed the use of ArabicGPt in the classification of Arabic satirical articles, and the challenge of detecting implicit satirical content was reported. An in-depth lexical analysis that supports the understanding of satirical content, which may help researchers build models that solve the detection of implicit satire, was reported. Finally, a fully developed model is provided freely to classify unorganized files into satirical and non-satirical textual files.

Our study broadens up the tasks of Arabic satire classification, which include satirical content written in a journalistic format, and offers methods for building models that differentiate satirical articles from non-satirical articles. This work can be integrated with social media platforms to guide the reader to satirical content, which can fight against the distribution of false information. For future work, we will alter the values of certain factors when constructing the models, and investigate their performance. It is also possible to combine the probability scores of several deep learning architectures with conventional classifiers to increase performance. We also plan to set forward a larger and more varied dataset to investigate the performance of several hybrid deep learning models and compare them to the results from ChatGPT.

Author Contributions: Conceptualization, F.A. and H.H.; methodology, F.A. and H.H.; software, H.H.; validation, F.A, and H.H.; analysis, F.A.; investigation, H.H.; resources, F.A and H.H.; data curation, H.H.; writing—original draft preparation, F.A.; writing—review and editing, H.H.; visual-

ization, H.H.; supervision, F.A.; project administration, F.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rahma, A.; Azab, S.S.; Mohammed, A. A Comprehensive Review on Arabic Sarcasm Detection: Approaches, Challenges and Future Trends. *IEEE Access* **2023**, *11*, 18261–18280. [CrossRef]
2. Baumgartner, J.C.; Morris, J.S. One "nation," under Stephen? The effects of the Colbert Report on American youth. *J. Broadcast. Electron. Media* **2008**, *52*, 622–643. [CrossRef]
3. Stones, S.; Glazzard, J.; Muzio, M.R. *Selected Topics in Child and Adolescent Mental Health*; BoD-Books on Demand: Norderstedt, Germany, 2020.
4. Egelhofer, J.L.; Lecheler, S. Fake news as a two-dimensional phenomenon: A framework and research agenda. *Ann. Int. Commun. Assoc.* **2019**, *43*, 97–116. [CrossRef]
5. Bowyer, B.T.; Kahne, J.E.; Middaugh, E. Youth comprehension of political messages in YouTube videos. *New Media Soc.* **2017**, *19*, 522–541. [CrossRef]
6. Baym, G.; Jones, J.P. News parody in global perspective: Politics, power, and resistance. *Pop. Commun.* **2012**, *10*, 2–13. [CrossRef]
7. Young, D.G.; Tisinger, R.M. Dispelling late-night myths: News consumption among late-night comedy viewers and the predictors of exposure to various late-night shows. *Harv. Int. J. Press/Politics* **2006**, *11*, 113–134. [CrossRef]
8. O'Keefe, P.A.; Horberg, E.; Plante, I. The multifaceted role of interest in motivation and engagement. In *The Science of Interest*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 49–67.
9. Baum, M.A. Soft news and political knowledge: Evidence of absence or absence of evidence? *Political Commun.* **2003**, *20*, 173–190. [CrossRef]
10. del Pilar Salas-Zárate, M.; Paredes-Valverde, M.A.; Rodriguez-García, M.Á.; Valencia-García, R.; Alor-Hernández, G. Automatic detection of satire in Twitter: A psycholinguistic-based approach. *Knowl.-Based Syst.* **2017**, *128*, 20–33. [CrossRef]
11. Gupta, A.; Kumaraguru, P.; Castillo, C.; Meier, P. Tweetcred: A real-time web-based system for assessing credibility of content on twitter. *arXiv* **2014**, arXiv:1405.5490.
12. Lichtheim, M. *Ancient Egyptian Literature*; Univ of California Press: Berkeley, CA, USA, 2019.
13. Peifer, J.; Lee, T. Satire and journalism. In *Oxford Research Encyclopedia of Communication*; Oxford University Press: Oxford, UK, 2019.
14. Young, D.G. Can satire and irony constitute misinformation. In *Misinformation and Mass Audiences*; University of Texas Press: Austin, TX, USA, 2018; pp. 124–139.
15. Cockerell, I. Fear, Panic and Fake News Spread after Ebola Outbreak in Uganda. 2022. Available online: https://www.codastory.com/newsletters/ebola-disinformation-uganda/ (accessed on 15 April 2023).
16. Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. In Proceedings of the 2014 Science and Information Conference, London, UK, 27–29 August 2014; pp. 372–378.
17. Velliangiri, S.; Alagumuthukrishnan, S. A review of dimensionality reduction techniques for efficient computation. *Procedia Comput. Sci.* **2019**, *165*, 104–111. [CrossRef]
18. Mehta, A.; Parekh, Y.; Karamchandani, S. Performance evaluation of machine learning and deep learning techniques for sentiment analysis. In *Information Systems Design and Intelligent Applications: Proceedings of Fourth International Conference INDIA 2017*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 463–471.
19. Allaith, A.; Shahbaz, M.; Alkoli, M. Neural Network Approach for Irony Detection from Arabic Text on Social Media. In Proceedings of the FIRE (Working Notes), Kolkata, India, 12–15 December 2019; pp. 445–450.
20. Nayel, H.; Amer, E.; Allam, A.; Abdallah, H. Machine learning-based model for sentiment and sarcasm detection. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 386–389.
21. Abuteir, M.M.; Elsamani, E. Automatic Sarcasm Detection in Arabic Text: A Supervised Classification Approach. *Int. J. New Technol. Res.* **2021**, *7*, 1–11.
22. Elgabry, H.; Attia, S.; Abdel-Rahman, A.; Abdel-Ate, A.; Girgis, S. A contextual word embedding for Arabic sarcasm detection with random forests. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 340–344.
23. Kanwar, N.; Mundotiya, R.K.; Agarwal, M.; Singh, C. Emotion based voted classifier for Arabic irony tweet identification. In Proceedings of the FIRE (Working Notes), Kolkata, India, 12–15 December 2019; pp. 426–432.
24. Abuzayed, A.; Al-Khalifa, H. Sarcasm and sentiment detection in Arabic tweets using BERT-based models and data augmentation. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 312–317.

25. Wadhawan, A. Arabert and farasa segmentation based approach for sarcasm and sentiment detection in arabic tweets. *arXiv* **2021**, arXiv:2103.01679.
26. Hengle, A.; Kshirsagar, A.; Desai, S.; Marathe, M. Combining Context-Free and Contextualized Representations for Arabic Sarcasm Detection and Sentiment Identification. *arXiv* **2021**, arXiv:2103.05683.
27. Sarsam, S.M.; Al-Samarraie, H.; Alzahrani, A.I.; Wright, B. Sarcasm detection using machine learning algorithms in Twitter: A systematic review. *Int. J. Mark. Res.* **2020**, *62*, 578–598. [CrossRef]
28. Karoui, J.; Zitoune, F.B.; Moriceau, V. Soukhria: Towards an irony detection system for arabic in social media. *Procedia Comput. Sci.* **2017**, *117*, 161–168. [CrossRef]
29. Al-Ghadhban, D.; Alnkhilan, E.; Tatwany, L.; Alrazgan, M. Arabic sarcasm detection in Twitter. In Proceedings of the 2017 International Conference on Engineering & MIS (ICEMIS), IEEE, Monastir, Tunisia, 8–10 May 2017; pp. 1–7.
30. Gupta, M.; Bakliwal, A.; Agarwal, S.; Mehndiratta, P. A comparative study of spam SMS detection using machine learning classifiers. In Proceedings of the 2018 Eleventh International Conference on Contemporary Computing (IC3), IEEE, Noida, India, 2–4 August 2018; pp. 1–7.
31. Moudjari, L.; Akli-Astouati, K. An Embedding-based Approach for Irony Detection in Arabic tweets. In Proceedings of the FIRE (Working Notes), Kolkata, India, 12–15 December 2019; pp. 409–415.
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
33. Zhou, W.; Bloem, J. Comparing Contextual and Static Word Embeddings with Small Data. In Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021), Dusseldorf, Germany, 6–9 September 2021; pp. 253–259.
34. Alharbi, A.I.; Lee, M. Multi-task learning using a combination of contextualised and static word embeddings for arabic sarcasm detection and sentiment analysis. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 318–322.
35. Gupta, P.; Jaggi, M. Obtaining better static word embeddings using contextual embedding models. *arXiv* **2021**, arXiv:2106.04302.
36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
37. Saadany, H.; Mohamed, E.; Orasan, C. Fake or real? A study of Arabic satirical fake news. *arXiv* **2020**, arXiv:2011.00452.
38. Farha, I.A.; Magdy, W. Mazajak: An online Arabic sentiment analyser. In Proceedings of the Fourth Arabic Natural Language Processing Workshop, Florence, Italy, 1 August 2019; pp. 192–198.
39. Naski, M.; Messaoudi, A.; Haddad, H.; BenHajhmida, M.; Fourati, C.; Mabrouk, A.B.E. iCompass at shared task on sarcasm and sentiment detection in Arabic. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 381–385.
40. Farha, I.A.; Zaghouani, W.; Magdy, W. Overview of the wanlp 2021 shared task on sarcasm and sentiment detection in arabic. In Proceedings of the Sxth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 296–305.
41. Godara, J.; Batra, I.; Aron, R.; Shabaz, M. Ensemble classification approach for sarcasm detection. *Behav. Neurol.* **2021**, *2021*, 9731519. [CrossRef]
42. Babanejad, N.; Davoudi, H.; An, A.; Papagelis, M. Affective and contextual embedding for sarcasm detection. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 225–243.
43. Sharma, D.K.; Singh, B.; Agarwal, S.; Kim, H.; Sharma, R. Sarcasm detection over social media platforms using hybrid auto-encoder-based model. *Electronics* **2022**, *11*, 2844. [CrossRef]
44. Israeli, A.; Nahum, Y.; Fine, S.; Bar, K. The idc system for sentiment classification and sarcasm detection in Arabic. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Kiev, Ukraine, 19 April 2021; pp. 370–375.
45. Băroiu, A.C.; Trăușan-Matu, Ș. Automatic Sarcasm Detection: Systematic Literature Review. *Information* **2022**, *13*, 399. [CrossRef]
46. AlMazrua, H.; AlHazzani, N.; AlDawod, A.; AlAwlaqi, L.; AlReshoudi, N.; Al-Khalifa, H.; AlDhubayi, L. Sa '7r: A Saudi Dialect Irony Dataset. In Proceedings of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, Marseille, France, 20–25 June 2022; pp. 60–70.
47. Yang, F.; Mukherjee, A.; Dragut, E. Satirical news detection and analysis using attention mechanism and linguistic features. *arXiv* **2017**, arXiv:1709.01189.
48. Rendalkar, S.; Chandankhede, C. Sarcasm detection of online comments using emotion detection. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (Icirca), IEEE, Coimbatore, India, 11–12 July 2018; pp. 1244–1249.
49. Ekman, P.; Sorenson, E.R.; Friesen, W.V. Pan-cultural elements in facial displays of emotion. *Science* **1969**, *164*, 86–88. [CrossRef] [PubMed]
50. Saad, M. Mining Documents and Sentiments in Cross-lingual Context. Ph.D. Thesis, Université de Lorraine, Lorraine, France, 2015.
51. Abdelali, A.; Darwish, K.; Durrani, N.; Mubarak, H. Farasa: A fast and furious segmenter for arabic. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, San Diego, CA, USA, 12–17 July 2016; pp. 11–16.
52. Alsmearat, K.; Al-Ayyoub, M.; Al-Shalabi, R.; Kanaan, G. Author gender identification from Arabic text. *J. Inf. Secur. Appl.* **2017**, *35*, 85–95. [CrossRef]

53. Alwajeeh, A.; Al-Ayyoub, M.; Hmeidi, I. On authorship authentication of arabic articles. In Proceedings of the 2014 5th International Conference on Information and Communication Systems (ICICS), IEEE, Irbid, Jordan, 1–3 April 2014; pp. 1–6.
54. Burgoon, J.K.; Blair, J.P.; Qin, T.; Nunamaker, J.F. Detecting deception through linguistic analysis. In Proceedings of the International Conference on Intelligence and Security Informatics, San Antonio, TX, USA, 2–3 November 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 91–101.
55. Gröndahl, T.; Asokan, N. Text analysis in adversarial settings: Does deception leave a stylistic trace? *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–36. [CrossRef]
56. Hajja, M.; Yahya, A.; Yahya, A. Authorship attribution of arabic articles. In Proceedings of the International Conference on Arabic Language Processing, Nancy, France, 16–17 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 194–208.
57. Himdi, H.; Weir, G.; Assiri, F.; Al-Barhamtoshy, H. Arabic fake news detection based on textual analysis. *Arab. J. Sci. Eng.* **2022**, *47*, 10453–10469. [CrossRef]
58. Ghannay, S.; Esteve, Y.; Camelin, N.; Dutrey, C.; Santiago, F.; Adda-Decker, M. Combining continuous word representation and prosodic features for asr error prediction. In Proceedings of the Statistical Language and Speech Processing: Third International Conference, SLSP 2015, Proceedings 3, Budapest, Hungary, 24–26 November 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 84–95.
59. Ghannay, S.; Favre, B.; Esteve, Y.; Camelin, N. Word embedding evaluation and combination. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portoroz, Slovenia, 23–28 May 2016; pp. 300–305.
60. Naseem, U.; Razzak, I.; Eklund, P.; Musial, K. Towards improved deep contextual embedding for the identification of irony and sarcasm. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, Glasgow, UK, 19–24 July 2020; pp. 1–7.
61. Ranasinghe, T.; Saadany, H.; Plum, A.; Mandhari, S.; Mohamed, E.; Orasan, C.; Mitkov, R. *RGCL at IDAT: Deep Learning Models for Irony Detection in Arabic Language*; University of Wolverhampton: Wolverhampton, UK, 2019.
62. Joulin, A.; Grave, E.; Bojanowski, P.; Douze, M.; Jégou, H.; Mikolov, T. Fasttext. zip: Compressing text classification models. *arXiv* **2016**, arXiv:1612.03651.
63. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
64. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
65. Soliman, A.B.; Eissa, K.; El-Beltagy, S.R. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Comput. Sci.* **2017**, *117*, 256–265. [CrossRef]
66. Hindocha, E.; Yazhiny, V.; Arunkumar, A.; Boobalan, P. Short-text Semantic Similarity using GloVe word embedding. *Int. Res. J. Eng. Technol.* **2019**, *6*, 553–558.
67. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
68. Shah, K.; Patel, H.; Sanghvi, D.; Shah, M. A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augment. Hum. Res.* **2020**, *5*, 1–16. [CrossRef]
69. Chen, H.; Wu, L.; Chen, J.; Lu, W.; Ding, J. A comparative study of automated legal text classification using random forests and deep learning. *Inf. Process. Manag.* **2022**, *59*, 102798. [CrossRef]
70. Pranckevičius, T.; Marcinkevičius, V. Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Balt. J. Mod. Comput.* **2017**, *5*, 221. [CrossRef]
71. Omar, A.; Mahmoud, T.M.; Abd-El-Hafeez, T.; Mahfouz, A. Multi-label arabic text classification in online social networks. *Inf. Syst.* **2021**, *100*, 101785. [CrossRef]
72. Al Qadi, L.; El Rifai, H.; Obaid, S.; Elnagar, A. Arabic text classification of news articles using classical supervised classifiers. In Proceedings of the 2019 2nd International Conference on New Trends In Computing Sciences (ICTCS), IEEE, Amman, Jordan, 9–11 October 2019; pp. 1–6.
73. Osisanwo, F.; Akinsola, J.; Awodele, O.; Hinmikaiye, J.; Olakanmi, O.; Akinjobi, J. Supervised machine learning algorithms: Classification and comparison. *Int. J. Comput. Trends Technol. (IJCTT)* **2017**, *48*, 128–138.
74. Vijayan, V.K.; Bindu, K.; Parameswaran, L. A comprehensive study of text classification algorithms. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, Manipal, India, 13–16 September 2017; pp. 1109–1113.
75. Xie, J.; Su, B.; Li, C.; Lin, K.; Li, H.; Hu, Y.; Kong, G. A review of modeling methods for predicting in-hospital mortality of patients in intensive care unit. *J. Emerg. Crit. Care Med.* **2017**, *1*, 1–10. [CrossRef]
76. George, J.; Skariah, S.M.; Xavier, T.A. Role of contextual features in fake news detection: A review. In Proceedings of the 2020 international conference on innovative trends in information technology (ICITIIT), IEEE, Kottayam, India, 13–14 February 2020; pp. 1–6.
77. Shaji, A.; Binu, S.; Nair, A.M.; George, J. Fraud Detection in Credit Card Transaction Using ANN and SVM. In Proceedings of the International Conference on Ubiquitous Communications and Network Computing, Bangalore, India, 8–10 February 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 187–197.
78. Pramanik, P.K.D.; Pal, S.; Mukhopadhyay, M.; Singh, S.P. Big Data classification: Techniques and tools. In *Applications of Big Data in Healthcare*; Khanna, A., Gupta, D., Dey, N., Eds.; Academic Press: Cambridge, MA, USA, 2021; pp. 1–43.

79. Learning, M. Machine Learning Plus. 2021. Available online: https://www.machinelearningplus.com/ (accessed on 1 September 2023).
80. Grover, K. Advantages and Disadvantages of Logistic Regression. 2022. Available online: https://iq.opengenus.org/advantages-and-disadvantages-of-logistic-regression/ (accessed on 1 September 2023).
81. Genuer, R.; Poggi, J.M.; Tuleau-Malot, C.; Villa-Vialaneix, N. Random forests for big data. *Big Data Res.* **2017**, *9*, 28–46. [CrossRef]
82. Razali, M.S.; Halin, A.A.; Chow, Y.W.; Norowi, N.M.; Doraisamy, S. Context-Driven Satire Detection with Deep Learning. *IEEE Access* **2022**, *10*, 78780–78787. [CrossRef]
83. Zhang, M.; Zhang, Y.; Fu, G. Tweet sarcasm detection using deep neural network. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, Osaka, Japan, 1–16 December 2016; pp. 2449–2460.
84. Venkatesh, B.; Vishwas, H. Real time sarcasm detection on twitter using ensemble methods. In Proceedings of the 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA) IEEE, Coimbatore, India, 2–4 September 2021; pp. 1292–1297.
85. Kenton, J.D.M.W.C.; Toutanova, L.K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the Proceedings of naacL-HLT, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, p. 2.
86. Antoun, W.; Baly, F.; Hajj, H. Arabert: Transformer-based model for arabic language understanding. *arXiv* **2020**, arXiv:2003.00104.
87. Alnabrisi, I.; Saad, M. *Detect Arabic Fake News Through Deep Learning Models and Transformers*; Available at SSRN 4341610; SSRN: Rochester, NY, USA, 2023.
88. Rehana, H.; Çam, N.B.; Basmaci, M.; He, Y.; Özgür, A.; Hur, J. Evaluation of GPT and BERT-based models on identifying protein–protein interactions in biomedical text. *arXiv* **2023**, arXiv:2303.17728.
89. Antoun, W.; Baly, F.; Hajj, H. AraGPT2: Pre-trained transformer for Arabic language generation. *arXiv* **2020**, arXiv:2012.15520.
90. Cer, D.M.; De Marneffe, M.C.; Jurafsky, D.; Manning, C.D. Parsing to Stanford Dependencies: Trade-offs between Speed and Accuracy. In Proceedings of the LREC, Floriana, Malta, 19–21 May 2010.
91. Abu Farha, I.; Magdy, W. From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, Marseille, France, 12 May 2020; pp. 32–39.
92. Braga, I.A. Evaluation of stopwords removal on the statistical approach for automatic term extraction. In Proceedings of the 2009 Seventh Brazilian Symposium in Information and Human Language Technology, IEEE, Sao Carlos, Brazil, 8–11 September 2009; pp. 142–149.
93. Rubin, V.L.; Chen, Y.; Conroy, N.K. Deception detection for news: Three types of fakes. *Proc. Assoc. Inf. Sci. Technol.* **2015**, *52*, 1–4. [CrossRef]
94. Ermida, I. News satire in the press: Linguistic construction of humour inspoof news articles. In *Language and Humour in the Media*; Cambridge Scholars Publishing: Newcastle upon Tyne, UK, 2012; p. 185.