IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 9 June 2023, accepted 23 June 2023, date of publication 26 June 2023, date of current version 11 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3289715

RESEARCH ARTICLE

Arabic Sentiment Analysis and Sarcasm Detection Using Probabilistic Projections-Based Variational Switch Transformer

S. MUHAMMAD AHMED HASSAN SHAH[1,2], SYED FAIZAN HUSSAIN SHAH[2], ASAD ULLAH[2], ATIF RIZWAN[3], GHADA ATTEIA[4], AND MAALI ALABDULHAFITH[4]

[1]Medical Imaging and Diagnostics Laboratory (MIDL), National Center of Artificial Intelligence (NCAI), COMSATS University Islamabad, Islamabad 63243, Pakistan
[2]Department of Computer Science, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan
[3]Department of Computer Engineering, Jeju National University, Jeju-si 63243, Republic of Korea
[4]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia

Corresponding authors: Ghada Atteia (geatteiaallah@pnu.edu.sa) and Maali Alabdulhafith (mialabdulhafith@pnu.edu.sa)

This work was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R407), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT Text classification is a common task in natural language processing (NLP), where the objective is to assign predefined categories or labels to a given text. Detecting sarcasm and classifying sentiment and dialect in NLP has practical applications, including spam detection, topic classification, and sentiment analysis. However, sarcasm and sentimental expressions, such as irony, humor, or criticism, can be difficult to identify through traditional NLP methods due to their implicit nature. To address this, we propose a Modified Switch Transformer (MST) for detecting sarcasm and classifying sentiment and dialect in Arabic text data. Our approach includes two key contributions: Variational Enmesh Expert's Routing (VE_eR) and Probabilistic Projections (P_ϕ). The switch transformer model incorporates probabilistic projections using a Variational Spatial Gated Unit-MLP to enhance the embedding generation mechanism. This updated mechanism introduces a variational aspect, providing dynamic control over the flow of information in the network, in contrast to the simpler embedding generation phase used in the original switch transformer. Moreover, we incorporate Variational Enmesh Expert's Routing, which utilizes a hierarchical set of Variational experts, where each expert is a small and variational-directed acyclic graph network. The VE_eR routing technique allows the network to dynamically choose which path to take at each layer based on the input, using a set of weights learned during training to determine the best route for a given input. Instead of optimizing route paths deterministically, we utilize Variational Inference and model each route as a random variable from a distribution. Our study evaluates the effectiveness of the Modified Switch Transformer (MST) model on the ArSarcasm Dataset, which includes Arabic language data related to sarcasm, dialect, and sentiments. We compare the performance of our proposed model with existing state-of-the-art models in the literature. The results show that the switch transformer outperforms other models in detecting sarcasm and also performs well in classifying sentiment and dialect.

INDEX TERMS Text classification, Arabic sarcasm detection, Arabic sentiment analysis, switch transformer, probabilistic projections, variational expert routing, NLP, Bayesian inference, deep learning.

I. INTRODUCTION

NLP is a Machine learning (ML) field that aims to make it possible for computers to comprehend and analyze human language. It involves developing models and algorithms that can process and analyze data in natural language, such as speech or text. For tasks including language translation, sentiment analysis, information extraction, classification of texts, and question-answering, NLP techniques use a variety of computational linguistics approaches, statistical

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

models, and neural networks. By utilizing machine learning, NLP closes the communication gap between humans and computational systems by allowing robots to understand and interact with human language. It revolutionizes how we interact with and use language-based data by having several applications in fields including virtual assistants, chatbots, language translation services, content analysis, and sentiment analysis.

Text classification is a common task in NLP, and it involves categorizing a given text into one or multiple predefined categories. ML and deep learning (DL) techniques have been used for text classification with great success recently. ML algorithms like Naive Bayes (NB), Support Vector Machines (SVM), and Random Forests (RF) are commonly used for text classification [1]. These algorithms work well with small to medium-sized datasets and are relatively simple to implement. According to [2], these algorithms demand minimal text data preprocessing, such as stop word removal and stemming. They are designed to learn patterns from the training data, which they then apply to classify new, previously unseen text data.

Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are gaining popularity for text classification, as evidenced by recent studies [3], [4]. In contrast, other methods may not be as effective. These methods are particularly useful when dealing with large-scale datasets and complex relationships between the words and phrases in the text. Deep learning approaches require more text data preprocessing, such as converting the text into numerical representations like word embeddings or sequences of character n-grams. These numerical representations are then used as input to deep learning models, which classify the text by learning the relationships between the words and phrases. To conclude, the selection of a suitable approach for text classification depends on the specific requirements of the task at hand, as both machine learning and deep learning techniques have their own strengths and weaknesses [5], [6].

However, deep learning methods are becoming more widely used due to their ability to handle large-scale and complex text data and their potential for providing improved accuracy compared to traditional machine learning methods. Sarcasm detection, on the other hand, is a subfield of text classification that focuses on identifying instances of sarcasm in text [7]. Sarcasm is a form of irony where the intended meaning is opposite to the expressed meaning, and it can often be difficult for computers to detect sarcasm in text. Sarcasm detection is important because it can help in sentiment analysis, improve the accuracy of recommendation systems, and also help improve the overall quality of human-computer interactions.

The variety and complexity of human language pose several major challenges for the field of NLP. The ambiguity and context sensitivity of language, where words and sentences can have various meanings depending on the surrounding situation, is a significant difficulty. Accurately separating these meanings from one another is still a major problem. Additionally, understanding natural language is difficult since it requires understanding the complexity, idiomatic idioms, sarcasm, and irony common in human speech. Textual comprehension demands sophisticated language models with deep understanding in order to understand the underlying intent and emotions portrayed in the text. Additionally, there are issues with data quality and availability because it can be difficult to find labeled datasets, particularly for languages with limited resources or specific fields.

NLP is important for sentiment analysis because it makes it possible to analyze the sentiments represented in text automatically. The subjective information, views, and attitudes expressed in written language are understood and interpreted using a variety of approaches. NLP is essential to sentiment analysis because it gives machines the ability to comprehend and interpret human language in terms of subjective information, such as views, feelings, and attitudes.

NLP helps in the sentiment analysis of text data by using a variety of methods and techniques. NLP algorithms preprocess the text to make it better suited for analysis by tokenizing it into individual words, eliminating extra words, and normalizing the content. To gather information on sentiment, they also extract relevant textual features, including word frequencies or word embeddings. These algorithms assign sentiment labels based on the extracted features and use the learned patterns and features to predict the sentiment of new, unlabeled material. Sentiment analysis using NLP approaches offers useful information for understanding market trends, customer feedback, social media sentiment, and public opinion, assisting in decision-making and enhancing user experiences.

Text classification and sarcasm detection are typically approached using machine learning algorithms. As noted by [8], supervised machine learning is the most prevalent method for text classification. It involves training the model on a labeled dataset and subsequently using it to categorize new and previously unseen text into predetermined categories. The features used to train the model can be various aspects of the text data, such as word frequency, word n-grams, and sentiment scores. For sarcasm detection, a few challenges need to be addressed. Firstly, sarcasm can be expressed in many different ways, making it difficult to train a model that can recognize all forms of sarcasm [9]. Secondly, the context in which the text was written is often crucial in determining the sarcasm in a statement, and capturing the context is a challenge in itself. Additionally, the tone and style of writing can also affect the detection of sarcasm, making it a complex task. Researchers have proposed several methods to tackle these challenges, including incorporating contextual and semantic features, utilizing meta-data such as user information and other contextual data, and implementing pre-trained language models such as BERT, as highlighted by [10] and [11]. These models have achieved good results on benchmark datasets, but there is still much room for improvement, and much work still needs to be done in this field.

Extensive research has been conducted on sarcasm detection in the English language, as evident from the various datasets and detection systems available [12], [13], [14], [15], [5]. Despite recent efforts, Arabic sarcasm detection remains in its infancy. Before these endeavors, there was an absence of publicly accessible Arabic sarcasm detection datasets, with research on the subject being restricted to an irony detection shared task and a dialectal sarcasm dataset, as discussed in [16]. To address this gap, Alhaqbani et al. created a new dataset called ArSarcasm, which consists of 10,547 tweets, of which 16% are sarcastic, and includes sarcasm and dialect labels. Furthermore, in [17], Ibrahim Abu-Farha et al. scrutinize the annotators' subjectivity in sentiment annotation and assess the efficacy of sentiment analysis systems in identifying sarcastic content.

Subjective language analysis has been a significant research area, particularly sentiment analysis (SA), over the past two decades. SA refers to the process of analyzing and extracting the emotional polarity of a given text. The classification of text into its sentiment class has been the focus of much research, varying based on the level of granularity. SA is a sub-discipline of NLP that has gained momentum due to the increasing prevalence of user-driven platforms like social media websites. While early research on SA concentrated on analyzing sentiment in movie reviews, it has since progressed to encompass diverse domains and areas, including social media analysis and computational social science, as stated in [18]. The Arabic language received less attention in this area until 2010. Arabic SA faces challenges due to the complex morphology of the language and the large variety of dialects.

As SA systems evolved, researchers began examining the inner workings of these systems to comprehend their efficacy and limitations [19]. As reported by [20], SA encounters various obstacles, including handling negation, dependency on the domain, lack of worldly knowledge, and sarcasm. Sarcasm, as noted in [5], refers to a type of verbal irony that conveys disdain or mockery. It is linked with indirectly expressing opinions where the intended meaning differs from the literal meaning. Sarcasm is context-dependent, occurring in shared knowledge between parties, and speakers use sarcasm only if they believe it will be interpreted as such [5]. Sarcasm detection is a vital task in SA, as sarcastic remarks typically convey negative implicit sentiments expressed through positive expressions. The divergence between the expressed and intended sentiments poses a complex obstacle for SA systems, as noted in [21].

Arabic SA confronts notable obstacles due to the intricate morphology and diverse dialects of the Arabic language. As with other languages, detecting sarcasm is a common challenge in Arabic SA, and few studies have addressed both sentiment and sarcasm detection in Arabic. In [22], Abdelrahman Kaseb et al. introduces a novel model architecture for multi-task training encompassing sentiment analysis, sarcasm detection, and dialect prediction. This model achieved state-of-the-art outcomes of 75.98 FPN on sentiment analysis on the ArSarcam-v2 dataset. The dataset's baseline is established by a BiLSTM-based model, which attains an F1-score of 0.46 on the sarcastic category, underscoring the difficulty of detecting sarcasm. ArSarcasm is publicly available for research purposes, and it can be downloaded at no cost by researchers.

Identifying Arabic dialects in written texts is a NLP task that can be performed at three different levels. The first level involves differentiating between Modern Standard Arabic (MSA), Classical Arabic (CA), and dialectical Arabic, as described in (source citation). The second level requires recognizing the specific dialect among the five major Arabic dialects, namely EGY, LEV, NOR, Gulf, and MSA. Finally, the third level involves identifying the dialect at the country level. As referenced in [23], [24], [25], [26], and [27], past studies have explored this task.

NLP and text classification with deep learning are two highly interrelated areas of research that have gained a lot of attention in recent years. However, despite the progress made in these fields, there are still many challenges that need to be overcome to achieve truly human-like performance. Some of the key challenges in NLP and text classification with deep learning are as follows:

Data Representation: NLP tasks require large amounts of data for training deep learning models. This data needs to be carefully preprocessed and represented in a way that is suitable for use with deep learning models. This can be challenging, as text data can contain many different structures and patterns that are difficult to capture using traditional representations.

Data Scarcity: Despite the vast amount of text data available on the Internet, obtaining high-quality, annotated data for NLP tasks can be difficult. This is particularly true for rare or under-represented languages and domains with specialized language use.

Word Embeddings: Word embeddings are a key component of many NLP models, as they provide a dense, continuous representation of words that can capture semantic relationships between words. However, choosing the right word embedding model and ensuring that the embeddings are trained on the right data can be challenging.

Contextualized Word Representations: NLP models often rely on word representations that are trained in a way that ignores context. However, many NLP tasks require word representations that are context-sensitive, and capturing these representations can be challenging.

Model Complexity: Deep learning models for NLP are often very complex, with many layers and parameters. This can make it challenging to interpret the models and understand why they are making certain predictions. It can also make the models difficult to train, as the models may be prone to overfitting the training data.

In text classification, one of the challenges is the imbalanced distribution of classes. In many real-world scenarios, there is a large difference in the number of samples between different classes, which can lead to biased predictions. Additionally, text classification models need to handle the variability and complexity of natural language, which includes dealing with spelling errors, misspellings, synonyms, and other language irregularities. Despite these challenges, NLP and text classification with deep learning are rapidly advancing fields. Many research articles are being done to overcome these challenges and push the boundaries of what is possible with NLP.

The presence of writing style and context poses a significant challenge in NLP. Words and phrases can be understood in a variety of ways depending on the context and writing style, which leads to ambiguity. It becomes crucial to accurately capture and interpret contextual information for disambiguation to address this challenge.

Writing style and context are interrelated in NLP because both are crucial to understanding and interpreting content. The context describes the prior knowledge or the particular setting in which the text is being given. It takes into account elements like prior sentences, the entirety of the document, or the dialogue. On the other hand, writing style describes the individual manner that a writer expresses oneself through language, including choices of vocabulary, syntax, tone, and more. Writing style can convey important contextual information, which explains how context and writing style are related.

This study introduces the MST (Modified Switch Transformer), a modified Switch Transformer that uses the Switch Transformer as its structural backbone. Switch transformers are designed to handle context-specific data by incorporating contextual information during the model's training and inference phases. When working with context-specific data, the switch transformer considers the surrounding information to produce accurate predictions. It does this by identifying long-range dependencies and contextual relationships within the text, enabling it to better comprehend the complexity and context-specific indications related to the task. Based on each input component's significance to the present context, the switch transformer dynamically assigns different weights to each one. It is able to successfully modify its attention mechanism to concentrate on the most relevant details within the context, which enhances its capacity to comprehend and analyze context-specific data. The study's primary contribution is as follows.

1) To control the information flow in a network and better understand the underlying patterns in data, we have developed a Spatial Gated Unit with Probabilistic Projections. The Spatial Gated Unit (SGU) is characterized by stochastic spatial and channel projections, which means that it is modeled using a probability distribution.
2) To improve the routing technique for dynamically adjusting the number of attention heads based on the input sequence length, we use variational Enmesh Expert Routing (VE_eR). The Switch Transformer improves upon previous routing techniques by utilizing variational experts and routes instead of simple MLP experts and deterministic routes. Here, experts are groups of neurons specialized in processing particular regions of the input sequence, and they are dynamically assigned to different regions by the routing mechanism. Meanwhile, routes refer to the process of assigning experts to different input sequence regions. The Switch Transformer's routing mechanism learns to assign experts based on their strengths and the model's needs at each network layer.

The paper is organized as follows: Section II reviews the previous studies that tackled the sarcasm detection problem in the related literature. Section III presents the proposed methodology. The study results are given in Section IV, and section V concludes the findings of this study.

II. LITERATURE REVIEW

The general population has the option to communicate with others through brief text updates on social media platforms like Twitter. This has produced an abundant area of research to look at linguistic expressions used on social media and investigate emotional expressions (sarcasm, humor, offense, etc.).

Sarcasm is characterized as a figurative language style in which an expression is intended to convey the reverse of its literal meaning [28]. Static Word Embedding's (SWE), Contextualized Embedding (MARBERT), and Multi-Task Learning (MTL) models were employed by different researchers to detect sarcasm [29], [30]. Alharbi et al. compare different language-based models and their results after these model training CNN-AraVec gives sarcasm (F1-sarcastic) 0.486 and Sentiment (F-PN) 0.534, MARBERT gives sarcasm (F1-sarcastic) 0.609 and Sentiment (F-PN) 0.702, MTL-CNN gives sarcasm (F1-sarcastic) 0.603 and Sentiment (F-PN) 0.666, MTL-LSTM gives sarcasm (F1-sarcastic) 0.619 and Sentiment (F-PN) 0.698 and MTL-CNN- LSTM gives sarcasm (F1-sarcastic) 0.632, and Sentiment (F-PN) 0.713 [31].

The advent of transformer-based language models in NLP has been a groundbreaking development. These models, including BERT, GPT, and ELECTRA, have led to significant progress in various NLP tasks and allowed researchers to achieve state-of-the-art performance. In a recent study, the effectiveness of 24 such algorithms in detecting Arabic sarcasm and sentiment was evaluated. Based on their research, the models that demonstrate the highest performance are those that utilize a greater number of parameters and are trained solely on Arabic data, including dialectal Arabic, like the newly introduced MARBERT model. During the experiments, several pre-trained models were fine-tuned for sentiment classification and sarcasm detection by adding a fully connected layer on top. The models compared include: Bi-LSTM, [11] mBERT (multilingual BERT based on BERT-base trained on 104 languages) citelan2020empirical, GigaBERT (trained on Arabic news articles and English translations), XLM-RoBERTa (XLM-R) [32], a multilingual extension of RoBERTa with base and large variants) [33], AraBERT (Arabic-specific BERT with different versions trained on different amounts of text and with/without Farasa pre-segmentation) [34], and AraELECTRA (Arabic-specific ELECTRA) [35]. Their result shows that the highest recall, precision and accuracy got from MARBERT 0.714, 0.714, and 0.584, respectively.

A recent study conducted by Muaad et al. aimed to detect instances of misogyny and sarcasm in Arabic text on various social media platforms, including Twitter, Facebook, and Instagram [36]. Their approach utilized seven different NLP classifiers, such as PAC, ARABERT, LRC, DTC, LSVC, RFC, and KNNC, to differentiate between these two topics in Arabic tweets [37]. The researchers employed two Arabic tweet datasets - one for misogyny and another for sarcasm - and proposed two detection scenarios for binary and multiclass classification. Results demonstrated that the AraBERT classifier achieved the highest accuracy in detecting misogyny, with 91.0% accuracy in binary and 89.0% accuracy in multiclass scenarios, while the same classifier achieved 88% accuracy in binary and 77.0% accuracy in multiclass scenarios for detecting sarcasm. These findings suggest that the proposed approach effectively identifies both misogyny and sarcasm in Arabic text and that AraBERT is a powerful deep-learning classifier for this purpose.

Elagbry et al. has proposed a novel approach for detecting Arabic sarcasm, which utilized different Deep Learning and ML techniques [38]. The study aimed to improve the accuracy of sarcasm detection in Arabic, which is difficult because of the complexities of the language. An accuracy of 0.5189 was achieved by utilizing the ArSarcasmV2 dataset with the proposed approach [39]. This result is an improvement compared to previous studies in Arabic sarcasm detection and highlights the effectiveness of the proposed approach.

In their submission to the WANLP ArSarcasm shared-task 2021, Hengle and colleagues present their approach to detecting sarcasm and sentiment in Arabic tweets. Their proposed model utilizes static word vectors and AraBERT representations to classify Arabic text. The results showed that their proposed system achieved superior performance compared to existing approaches and ranked second in sarcasm detection and tenth in sentiment identification. The improved performance was attributed to the combination of contextual information captured by the AraBERT model and the complementary information provided by the word vectors trained on social media texts. The proposed method was evaluated on the ArSarcasm shared-task 2021 and ranked second in the sarcasm detection subtask and tenth in the sentiment identification subtask.

Husain et al. showcased two systems submitted to the Sarcasm Detection and Sentiment Analysis sub-tasks at the Arabic NLP Workshop 2021 (WANLP 6) [40]. The approach is inspired by previous research findings in offensive language studies and the relationship between sentiment analysis, sarcasm detection, and offensive language detection [39]. The approach described in [41] uses transfer learning, which involves fine-tuning a model across various tasks, including detecting offensive language, identifying sarcasm, and analyzing sentiment. By exploring the influence of offensive language on sarcastic language and sentiment, the study aims to enhance the performance of the model. The authors use contextualized word embeddings that are learned from the entire fine-tuning corpus. To achieve transfer learning, the system undergoes joint training on multiple tasks and transfer corpora. Additionally, the system employs subword learning on the combined corpora of offensive language and either sarcasm detection or sentiment analysis. The system's architecture is described in a figure, and two separate submissions were made for each sub-task, trained, and tested independently. The result of Fatemah Husain et al. study with AraBERT model (Accuracy = 0.7607, Recall = 0.6622, Precision = 0.6950, F1-Score = 0.5041) and SalamBERT model (Accuracy = 0.7727, Recall = 0.6807, Precision = 0.7128, F1-Score = 0.5348).

As part of the sixth Arabic NLP workshop's shared task, Abraham and colleagues conducted a research study on detecting Arabic sarcasm and sentiment, which was documented in [42]. The researchers employed both advanced transfer-learning models and traditional machine-learning techniques to develop two distinct algorithms for detecting Arabic sarcasm and sentiment. Despite the growing importance of non-English natural language processing, particularly for Arabic due to the rising use of social media for communication, there has been insufficient research in the area of Arabic sentiment and sarcasm analysis, as highlighted in [43]. The authors of the paper proposed a system pipeline to examine the impact of offensive language linguistic features on both sarcastic language and sentiment content. They evaluated the performance of their sentiment classification and sarcasm detection tasks on a test dataset of 3000 tweets and achieved the 5th rank in sentiment classification and 9th rank in sarcasm detection among 22 and 27 participating teams, respectively. However, the authors noted a significant decline in performance when their models were tested on the test set, with a 4% reduction in sentiment classification accuracy and a 13.6% decrease in sarcasm detection accuracy.

In NLP, sarcasm detection has become a popular research topic. One way to collect data for sarcasm detection is through distant supervision, where specific hashtags or markers are used to indicate sarcasm on social media platforms. For instance, Davidov et al. and Khodak et al. used distant supervision to create datasets from Twitter and

Reddit, respectively. Another approach to data collection is manual labeling, as used by Riloff et al. and Van Hee et al. However, this approach may not capture the intended sarcasm. Oprea and Magdy proposed a new dataset that includes intended sarcasm, which they gathered through an online survey where participants posted sarcastic and non-sarcastic tweets [44], [45], [46], [47], [48], [49].

Since the introduction of the topic by Abbasi et al., Arabic sentiment analysis has been an important research focus. However, researchers later began to concentrate on dialectal Arabic as well [49], [50]. There are currently over fifty datasets available for sentiment analysis, including Elshakankery et al., Kaseb and Farouk, Kiritchenko et al., Rosenthal et al., and Elmadany et al. datasets. As a result, numerous system approaches have been developed for Arabic sentiment analysis, including word embedding with deep learning models, classical machine learning models, and transformer-based models (Abu Farha and Magdy; Alayba et al.; El-Beltagy et al.). According to a comparative study conducted by Abu Farha and Magdy, transformer-based models performed better than both classical machine learning models and word embedding with deep learning models [51], [52], [53], [43], [54].

Dialect prediction is the task of determining the dialect or regional variation of a speaker or text. In one of the early studies on dialect prediction, Eisenstein et al. proposed a method for identifying dialects in Twitter data using a machine learning-based methodology [55].

Pang et al. conducted one of the initial studies on sentiment analysis, where they categorized movie reviews as either positive or negative [18]. Since then, other techniques like lexicon-based, machine learning-based, and hybrid methods have been introduced. More recently, Pang et al. [56] suggested a new graph-based approach for identifying the sentiment of tweets. Sentiment analysis is a widely used method for determining the emotional tone of a text, allowing it to be classified as positive, negative, or neutral.

Al-Ghadhban et al. proposed a classifier model to detect Arabic-sarcasm tweets using Weka and set certain features to classify the tweets as sarcastic. The model was evaluated using recall, precision, and f- score measurements and achieved high performance with results of 0.659, 0.710, and 0.676, respectively [57]. The authors emphasize the importance of detecting sarcasm in tweets for various applications, including improving customer service and product fabrication, and highlight the limited studies on detecting Arabic sarcasm.

Various transformer architectures, including BERT [11], ALBERT [58], and RoBERTa [59], have been presented in the literature. However, in order to get better results for our study, we used a customized switch transformer. In some circumstances, Switch Transformers outperform more traditional transformers like BERT, ALBERT, and RoBERTa. Compared to BERT, Switch Transformers bring about architectural changes and advances. These adjustments, such as adding switch layers and dividing input sequences, enable handling long-range dependencies more effectively and increasing computing efficiency. Switch Transformers can perform better than BERT if these unique architectural benefits match the specifications of a certain task.

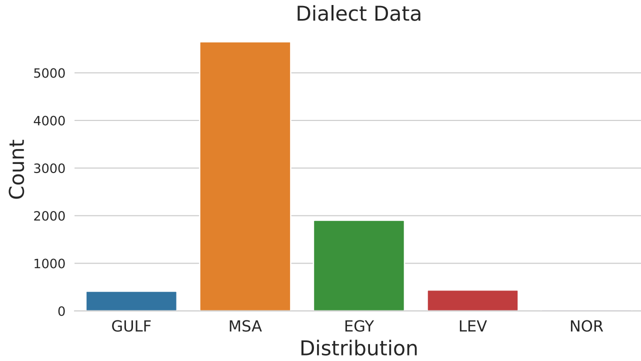

FIGURE 1. **Distribution of tweets on the basis of dialect.**

Switch transformers are better suited to handle long-range dependencies than traditional transformers. They achieve this by combining mechanisms for both global and local attention. Switch Transformers have the potential to be more computationally effective than conventional transformers. Traditional transformers suffer substantial computational costs for longer sequences since each layer must process the whole input sequence. Switch Transformers deal with this problem by dividing tokens and processing each dividing independently. Switch Transformers show potential in few-shot learning situations. Switch Transformers can generalize and learn from a few examples because of their global attention mechanism, which also allows for capturing higher-level contextual information [60].

III. MATERIALS AND METHODS

A. DATASET

The ArSarcasm dataset is a collection of approximately 27,000 Arabic tweets that have been manually annotated for sentiment and sarcasm by Arabic-speaking annotators. This dataset, designed for Arabic sentiment analysis and sarcasm detection tasks, comprises tweets from a diverse range of subjects, including politics, sports, and entertainment. The sentiment annotation contains three categories: positive, negative, and neutral, while the sarcasm annotation comprises two categories: sarcastic and non-sarcastic. The dataset is intended for the development and evaluation of machine learning models for Arabic sentiment analysis and sarcasm detection tasks.

ArSarcasm is a newly introduced dataset designed to detect Arabic sarcasm, including sarcasm and dialect labels, in addition to sentiment analysis labels from previously available datasets such as SemEval 2017 [61] and ASTD [62]. The dataset consists of 10,547 tweets, out of which 1,682 (16%) are sarcastic [17]. The distribution of data regarding dialect, sentiment, and sarcasm is presented in Figures 1, 2, and 3, respectively. A detailed description is provided in Table 1 for further information about the dataset.

TABLE 1. Overview of Characteristics in the Dataset.

Dialect	Non-Sarcastic	Sarcastic	Negative	Neutral	Positive	Total
Egyptian	1584	799	1179	733	471	2383
Gulf	397	122	200	218	101	519
Levantine	433	118	239	178	134	551
Maghrebi	20	12	18	10	4	32
MSA	6431	631	1893	4201	968	7062
Total	8865	1682	3529	5340	1678	10547

Egyptian, Levantine, North African, Gulf, and Modern Standard Arabic are among the primary dialects of Arabic. While having a shared Arabic foundation, these dialects vary significantly in pronunciation, vocabulary, grammar, and cultural influences as a result of historical and regional disparities. Here is a summary of how various dialects differ and are similar:

1) Egyptian (EGY): Due to Egypt's cultural influence on the Arab world, Egyptian Arabic is one of the most well-known dialects and is extensively spoken in Egypt. It has distinct linguistic characteristics and pronunciation, which are frequently characterized by the presence of the "ayn" and "ghayn" sounds. The words and phrases used in Egyptian Arabic were additionally influenced by French, English, and other languages.
2) Levantine (LEV): The Levantine dialect is used in Syria, Lebanon, Jordan, and Palestine, among other places. It has variants according to particular places and cities. Levantine Arabic is significantly influenced by Aramaic, Turkish, French, and English, and has a softer pronunciation than other dialects.
3) North African (NOR): Countries including Morocco, Algeria, Tunisia, and Libya all have Arabic-speaking populations. They share striking parallels with the Tamazight language family and are influenced by native Berber languages. The pronunciation of "qaf" is typically pronounced as "g" in North African languages, while the letter "j" is typically replaced with the sounds "sh" or "ch."
4) Gulf: The countries of the Arabian Peninsula, including Saudi Arabia, the United Arab Emirates, Qatar, Bahrain, Kuwait, and Oman, speak Gulf dialects. One distinctive linguistic trait of Gulf Arabic is the pronunciation of "qaf" as a "g" or "k" sound. It also has a large lexicon that is tied to both marine and desert cultures.
5) Modern Standard Arabic (MSA): MSA is a standardized form of Arabic used in formal contexts, media, literature, and education throughout the Arab world. It is not a spoken dialect. It is used as a universal language and is close to traditional Arabic. Arabic speakers from various places can use MSA because it is more uniform and user-friendly. Its pronunciation, however, is distinct from spoken dialects, and its grammatical structure is more traditional.

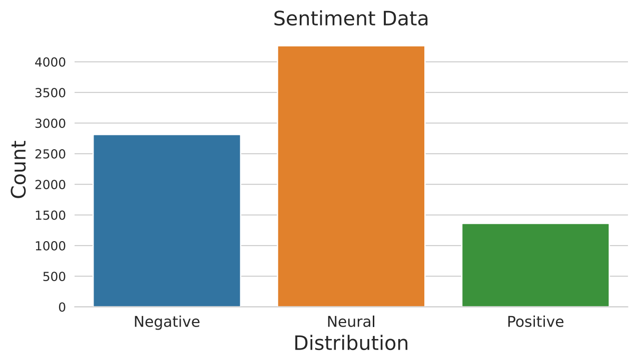

FIGURE 2. Distribution of tweets on the basis of sentiment.

Each accent has distinctive linguistic traits, such as how certain letters or sounds are spoken, different tone patterns, and distinctive regional accents. Each region's vocabulary includes terms and expressions shaped by its historical, cultural, and linguistic background. Vocabulary differs among accents. Verb conjugations, pronoun usage, and sentence patterns are only a few examples of how grammar structures and usage can vary. Additionally, each accent has been modified by cultural influences, historical exchanges, and exposure to other languages, creating a variety of linguistic landscapes within the Arabic language family. Even though Arabic speakers have common origins and can comprehend one another, knowing and accepting these variances is essential for effective communication and an awareness of cultural diversity among the various Arabic-speaking regions.

B. PROPOSED METHODOLOGY

In this research we proposed a novel and modified switch transformer, which use probabilistic projections (Variational spatial gated unit embedding) and Variational Enmesh Experts routing for detecting irony and sarcasm in Arabic text dataset. Figure 4 shows the methodology of the modified switch transformer architecture.

C. SWITCH TRANSFORMER

Switch Transformer is a type of neural network architecture for text classification, which utilizes a dynamic switching mechanism to determine the appropriate number of attention heads to use at each layer [60]. This allows the model to adapt to the varying complexities of different input sequences

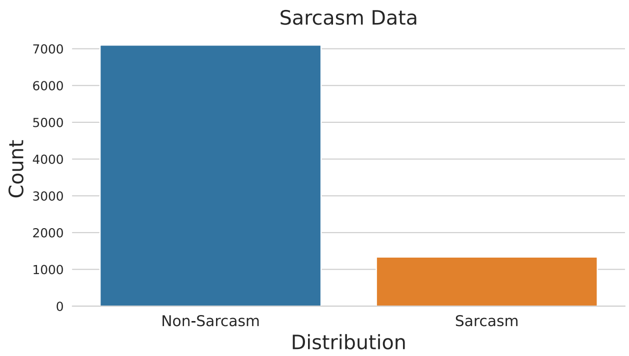

FIGURE 3. Distribution of tweets on the basis of sarcasm.

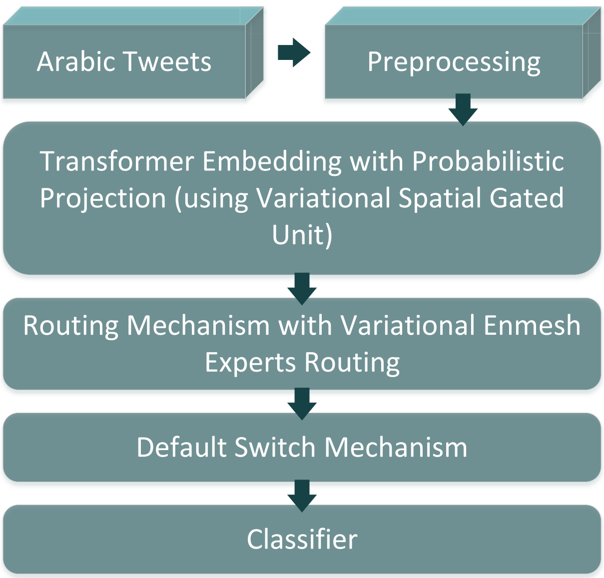

FIGURE 4. Flowchart illustrating the methodology of the proposed architecture, highlighting the key steps of involved in the proposed model.

and improve its performance on text classification tasks. The architecture is inspired by the transformer architecture but with a switch mechanism to dynamically adjust the number of attention heads, which can help improve the computational efficiency and overall performance of the model, as shown in Figure 5.

The routing mechanism in the Switch Transformer is a mechanism for dynamically determining the number of attention heads to use at each layer in the network. It is based on the idea of "routing" information through a set of possible paths or attention heads. The mechanism uses a routing function that takes input from the current layer's hidden representation and outputs a set of routing weights that determine which attention heads should be used. These routing weights are updated dynamically during the forward pass of the network, allowing the model to adapt to the changing complexities of the input sequence and adjust the number of attention heads accordingly. This allows the Switch Transformer to be more flexible and efficient than traditional transformer models, which use a fixed number of attention heads throughout the entire network.

D. BAYESIAN LEARNING (POSTERIOR AND PRIOR DISTRIBUTION)

Bayesian learning is a statistical approach that involves updating our belief about a hypothesis based on new data using Bayes' theorem [63]. It uses the prior probability distribution, P(H), which reflects our initial belief about the hypothesis before seeing the data, and the likelihood function, P(D|H), which is the probability of observing the data given the hypothesis, to calculate the posterior probability distribution, P(H|D), which is the probability of the hypothesis given the data.

Mathematically, the Bayes' theorem can be expressed as [64]:

$$P(H|D) = \frac{P(D|H).P(H)}{P(D)} \tag{1}$$

Bayesian learning is a type of learning that involves using probabilistic models and Bayesian inference to make predictions about new data. In Bayesian learning, we specify a prior distribution that reflects our assumptions about the model parameters prior to observing any data. As we collect new data, we update our beliefs by computing the posterior distribution, which reflects our updated beliefs about the parameters given the data we have observed.

The posterior distribution is computed using Bayes' rule, which states that the probability of a hypothesis (in this case, the parameter values of our model) given the data is proportional to the probability of the data given the hypothesis (i.e., the likelihood) times the prior probability of the hypothesis. In mathematical notation, Bayes' rule can be expressed as:

$$P(hypothesis|data) = \frac{P(data|hypothesis) \, . \, P(hypothesis)}{P(data)} \tag{2}$$

The equation $P(hypothesis|data) \propto P(data|hypothesis) \cdot P(hypothesis)/P(data)$ is a fundamental concept in Bayesian statistics. It expresses how to update our belief (the prior probability) in a hypothesis, given new evidence (the data). The left-hand side, $P(hypothesis|data)$, is called the posterior probability, which we want to calculate after observing the data. The right-hand side is composed of two factors: $P(data|hypothesis)$ is the likelihood function, which quantifies how well the hypothesis explains the data, and $P(hypothesis)$ is the prior probability, which represents our initial belief in the hypothesis before seeing any data. The denominator, $P(data)$, is a normalizing constant that ensures that the posterior probability integrates to 1 over all possible hypotheses. The equation provides a framework for combining prior knowledge with observed data to make probabilistic predictions and decisions.

The prior distribution and posterior distribution are important concepts in Bayesian learning because they allow us

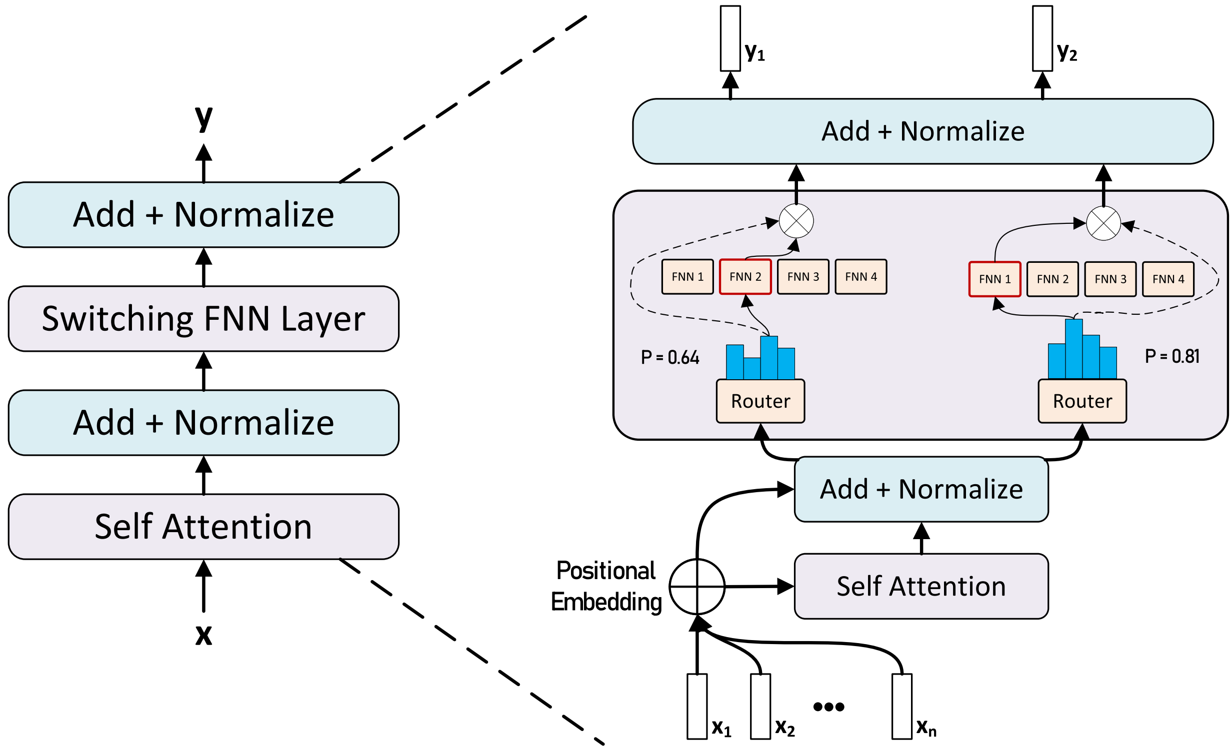

FIGURE 5. Working diagram of a switch transformer for NLP problems.

to incorporate prior knowledge and update our beliefs in a principled way as we observe new data. By starting with a prior distribution reflecting our initial beliefs about the parameters, we can update our beliefs in a way consistent with the data we have observed. This can lead to more accurate and robust models, especially in situations where the amount of data is limited or noisy.

E. PROBABILISTIC PROJECTIONS WITH SGU-MLP

The spatial gated unit refers to a gating mechanism that helps the network control the flow of information and decide which features to use. This gating mechanism is applied at each layer of the MLP, allowing the network to dynamically adjust its internal representation to capture the structure of the input image better [65]. We make this process stochastic by modeling data flow with a probability distribution. In the case of using deterministic neural flow, we use a variational layer.

$$X = \delta(XU) \tag{3}$$

$$Z^{\sim} = \delta(Z) \tag{4}$$

$$Y = ZV \tag{5}$$

The formula uses an activation function, such as GeLU, represented by δ. The linear projections, U and V are variational along the channel dimension and are defined in the same manner as in the FFNs of Transformers. For instance, the BERT base has shapes of 768×3072 and 3072×768, respectively. The diagram omits shortcuts, normalizations, and biases for clarity.

For cross-token interactions to occur, the layer s $(\cdot)$ must incorporate a compression operation along the spatial dimension. A straightforward approach is to use a linear projection.

$$F_{w,b}(Z) = W.Z + b \tag{6}$$

The matrix W, which has the same size as the sequence length, n, and token-specific biases b, is used in the equation. For instance, if the padded input sequence contains 128 tokens, the shape of W will be 128×128. Unlike self-attention, where $W(Z)$ is generated dynamically from Z, the matrix W is not dependent on the input representations. In this study, the layer s $(\cdot)$ is formulated as the result of linear gating, specifically, $s(Z) = Z$.

$$s(Z) = Z \odot F_{w,b}(Z) \tag{7}$$

where $\odot$ denotes the element wise multiplication. Here W weight matrix also drawn from probability distribution. The variational dense layers refer to a type of dense layer in the network that uses a variational dropout technique. The dropout regularization technique involves randomly removing or "dropping out" units from a neural network during training, thereby introducing some degree of uncertainty into the model. This approach can help prevent overfitting, where the model becomes too closely tailored to the training data and performs poorly on new, unseen data. By forcing the network to learn more robust and independent features, dropout regularization can improve the model's generalization performance on new data.

The combination of the spatial gated units and the dense variational layers results in a model that can capture complex features while also being more robust to changes in the input data. The following figure 6 displays the spatial gated unit featuring probabilistic projections.

Using probabilistic projections instead of simple projections has an advantage in that it can model the uncertainty in the weights of the encoding block. This is particularly beneficial in cases where there is limited training data or when the network is required to generalize well to new, unseen data. Additionally, the uncertainty estimates from the variational layer can be employed for various purposes, including active learning and Bayesian optimization.

F. VARIATIONAL ENMESH EXPERT ROUTING (VE_eR)

In the Switch Transformer architecture, EOM (End of Message) Routing is employed as a technique to dynamically regulate the number of attention heads utilized at each layer [60]. This method involves using an EOM symbol to indicate the conclusion of a sequence and then adapting the number of attention heads based on the length of the input sequence.

In the Switch Transformer architecture, the EOM Routing method calculates the routing weight for each attention head, which decides the head's significance in the final representation of the input sequence. The routing weight is computed based on the sequence's hidden state and the layer's present hidden state. The routing function dynamically controls the number of attention heads employed at each layer, which utilizes these weights and adapts the number of attention heads to the length of the input sequence.

At a given layer, the router variable, denoted as W_r, generates logits $h(x) = W_r.x$, which are then normalized using a softmax distribution across the N available experts. The corresponding gate-value for a particular expert, i, is determined by

$$P_i(x) = \frac{h(x)_i}{\sum_j^N e^{h(x)_j}} \tag{8}$$

The token x can be routed by selecting the top-k gate values, where $\mathcal{T}$ denotes the set of indices corresponding to the top-k values. To calculate the layer's output, each expert's computation on the token is linearly weighted by its corresponding gate value and the results are summed.

$$y = \sum_{i\epsilon\mathcal{T}} P_i(x)E_i(x) \tag{9}$$

Although the expert in Routing typically involves an MLP, we have developed a specialized expert that is both variational and entangled. As a result, we refer to this approach as Variational Enmesh Expert Routing. (The term "enmesh" in this context is used to refer to a nonlinear directed acyclic graph network (DAG) that is entangled or interconnected).

Figure 7 shows the variational transformation (also called variational layer) is denoted by Ω symbol. One of the main benefits of variational layer is the ability to model the uncertainty in the weights of the network, which can be especially useful in scenarios where the amount of available training data is limited or when the network is expected to generalize well to new, unseen data. This uncertainty estimation is provided by the variational layer's ability to learn a probability distribution over its weights rather than a single point estimate. The "+" symbol denotes the fusion approach, specifically addition, as illustrated in the diagram. The entire network is a directed acyclic graph, which facilitates the learning of non-linear patterns in the architecture. DAGs in deep learning offer numerous benefits, such as enabling non-linear connections between layers, resulting in more complex and flexible architectures.

Additionally, DAGs support dynamic architectures, making them adaptable to different input types and sizes. Moreover, these models can perform complex (graph) computations, such as attention mechanisms, which have proven effective in various domains, including NLP.

Using the EOM symbol allows the network to dynamically adjust the number of attention heads based on the length of the input sequence. This helps the network to be more computationally efficient and improve its performance on text classification tasks by avoiding over- or under-utilizing attention heads.

The EOM Routing mechanism provides a flexible and efficient way for the Switch Transformer to adapt to the changing complexities of different input sequences and achieve improved performance on text classification tasks.

IV. RESULTS AND DISCUSSION

This research work aims to classify Arabic sentiments and detect Arabic irony and sarcasm in textual data using an advance and modified switch transformer. However, it can also be used for other Text classification tasks. In this section, we perform a comparative analysis between the proposed architecture and different machine learning algorithms (trained on the same data). Table 2 shows the parameters of the proposed architecture.

A. COMPARATIVE ANALYSIS WITH OTHER MACHINE LEARNING ALGORITHMS

ML algorithms use mathematical models to make predictions or decisions by learning from data. There are two types of ML algorithms: supervised and unsupervised. Supervised learning involves training the algorithm on labeled data to identify patterns associated with specific outcomes, while unsupervised learning identifies patterns in unlabeled data without prior knowledge of what the patterns might be. Unsupervised learning is often used for clustering similar data points based on shared characteristics.

In this note, we will discuss six popular supervised machine learning algorithms: Random Forest Classifier, Naive Bayes, Support Vector Classifier (SVC), K-Nearest Neighbors Classifier (KNN), Logistic Regression, and Decision Tree Classifier [66].

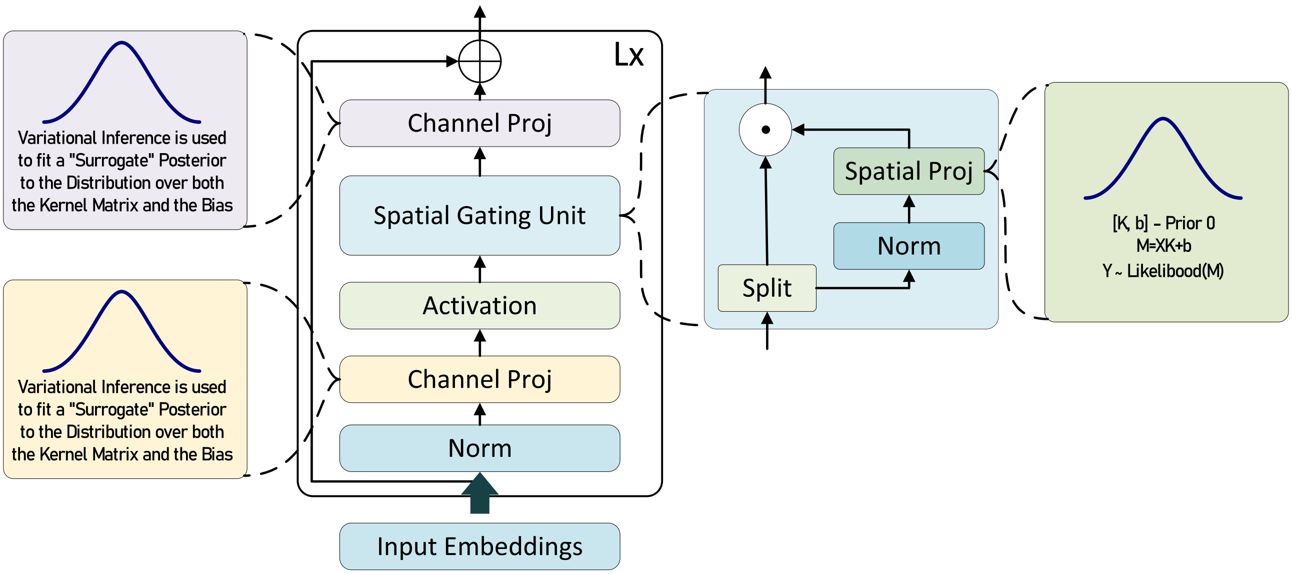

FIGURE 6. **Probabilistic Projection based SGU (Spatial Gated Unit) Multi-layer Perception described through a diagram showcasing the key steps and processes involved. The weights and biases of the prior distribution are represented by K and b respectively. The linear equation with a learnable slope and y-intercept is denoted by the symbol M. Finally, Y represents the likelihood of M, which is the output of the model.**

TABLE 2. **Hyperparameters of the proposed architecture.**

MST: Switch Transformer with Probabilistic Projections & Variational Enmesh Expert's Routing		
S.No	**Detail**	**Quantitative values**
1	Mechanisms in proposed MST	Transformer Embedding's with , Routing and Switch Controlling
2	Size of vocabulary	20000
3	features	200
Optimization Parameters		
4	Epoch	20
5	No of classes	2
6	Batch size	50
7	Learning rate	0.001
8	Optimizer	Adam
9	Trainable Parameters	2,503,369

One of the types of algorithms used in supervised learning is the decision tree, which involves hierarchical if-else conditions to predict the label for a given instance in the data [67]. A more advanced version of the decision tree is the Random Forest Classifier, which utilizes multiple decision trees and is therefore referred to as a "forest" of decision trees.RFC falls under the category of ensemble learning, where each tree in the forest learns from a subset of the data, and the results are aggregated in the end [68].

After training the random forest classifier on an Arabic sarcasm dataset, we achieved an accuracy of 81.66%. This study utilized the SVC algorithm as a linear classifier that distinguishes data into two groups by detecting a hyperplane that optimizes the margin between the classes. The margin is calculated as the distance between the nearest points in the data, known as the support vectors and the hyperplane. SVC can be used for both binary and multi-class problems using the one-vs-one or one-vs-all approaches [69]. The visual summary of SVM is depicted in Figure 8.

We also trained an SVM on the same Arabic sarcasm text dataset and achieved an accuracy performance measure of 83.52%. In addition, we trained several other well-known machine learning classifiers, and their respective results are presented in the Table 3.

To classify sentiments, various deep learning models were employed, and the MST model had the highest recall rate, but very low accuracy and f1-score. Other machine learning (ML) algorithms had accuracy, recall, and f1-scores that were approximately 50%. On the other hand, in the case of dialect classification, our proposed model had an accuracy rate of 67% and a recall rate of 90%. In contrast, other ML algorithms had good accuracy but very low recall. Therefore, our algorithm performed very well compared to others regarding the recall.These ML models are trained

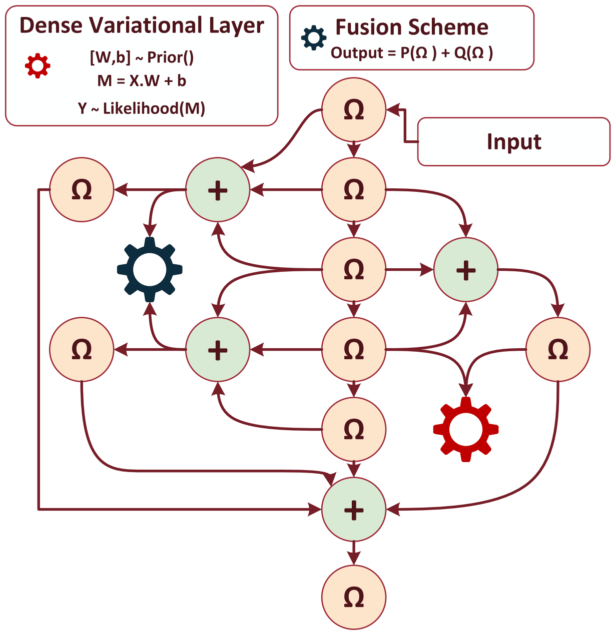

FIGURE 7. A schematic representation of the working principle of Expert of Variational Enmesh Experts Routing (EVE) architecture.

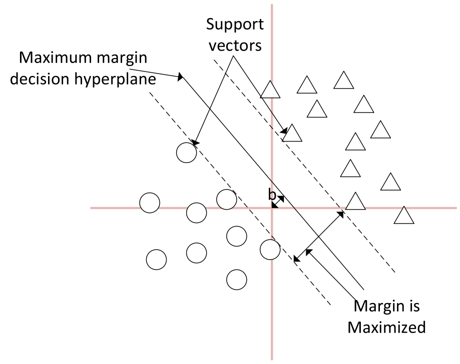

FIGURE 8. Support Vector Machine (Graphical Representation).

alongside the proposed MST for the purpose of conducting a comparative analysis. The outcomes of these ML algorithms in the case of sarcasm detection, dialect classification, and sentiment analysis are presented in tables 3, 4, and 5, respectively.

1) SARCASM DETECTION: COMPARISON BETWEEN PROPOSED MODEL AND OTHER MACHINE LEARNING MODELS

This section of the paper presents a comparative analysis of several machine-learning classifiers for detecting sarcasm in Arabic text. The classifiers that were evaluated in this study include K-Nearest Neighbors Classifier, Random Forest Classifier, Logistic Regression, and Naïve Bayes. The authors compare the performance of each classifier based on various performance metrics such as accuracy, precision, recall, and F1-score, as shown in Table 3. The primary objective of this analysis is to identify the most effective classifier for

TABLE 3. Comparative analysis in context of sarcasm detection.

No	Algorithm	Accuracy	Recall	F1-Score
1	Random Forest Classifier	0.84	0.84	0.78
2	K – Nearest Neighbors	0.8190	0.82	0.76
3	Naïve Bayes	0.6904	0.69	0.71
4	Decision Tree Classifier	0.7447	0.74	0.75
5	Logistic Regression	0.8352	0.84	0.76
6	Support Vector Machine	0.83523	0.84	0.76
7	Proposed MST Algorithm	0.8357	0.8350	0.8329

detecting sarcasm in Arabic text, which can then be utilized to develop more accurate and efficient sarcasm detection models in the future. The insights gained from this study can be valuable for researchers in natural language processing, especially for those focusing on Arabic text analysis.

Table 3 clearly shows that the proposed MST architecture and Random Forest Algorithm perform very well compared to other machine learning algorithms in terms of accuracy, recall, and f1-score. According to Table 3, the proposed MST architecture and the Random Forest Algorithm have outperformed other algorithms in all three metrics - accuracy, recall, and f1-score. However, it is also noted that other algorithms have achieved good accuracy but have low recall and f1-score. This implies that those algorithms are good at predicting true positives but are not very effective at identifying false negatives. The comparison of the f1-score for sarcasm detection is presented in Table 3.

Therefore, it can be concluded that the proposed MST architecture and Random Forest Algorithm are more effective in classifying sentiments as compared to other algorithms, which have shown to be less effective in terms of recall and f1-score.

2) SENTIMENT CLASSIFICATION: COMPARISON BETWEEN PROPOSED MODEL AND OTHER MACHINE LEARNING MODELS

A range of machine learning classifiers such as K-Nearest Neighbors, Random Forest, Logistic Regression, and Naïve Bayes was used to conduct Arabic sentiment classification, and their performance was evaluated. A comparison of the outcomes obtained from each classifier is presented in Table 4. This analysis aims to identify the most effective classifier for Arabic sentiment classification, which can be useful for researchers in natural language processing. The results of this study offer valuable insights into the performance of different classifiers in Arabic sentiment classification, which can aid in the development of more accurate and efficient sentiment classification models in the future.

Table 4 presents a comparison of various machine learning classifiers used for sentiment classification. It shows that the proposed architecture outperforms other algorithms in terms of recall, but it has lower accuracy and precision. One possible reason for this could be the absence of data cleaning processes such as stop-word removal in the training data.

TABLE 4. Comparative analysis in context of sentiment analysis.

No	Algorithm	Accuracy	Recall	F1-Score
1	Random Forest Classifier	0.5566	0.56	0.51
2	K – Nearest Neighbors	0.4704	0.47	0.44
3	Naïve Bayes	0.4528	0.45	0.46
4	Decision Tree Classifier	0.4604	0.46	0.46
5	Logistic Regression	0.5190	0.52	0.42
6	Support Vector Machine	0.5228	0.52	0.42
7	Proposed MST Algorithm	0.52	0.8076	0.5612

TABLE 5. Comparative analysis in context of dialect classification.

No	Algorithm	Accuracy	Recall	F1-Score
1	Random Forest Classifier	0.6671	0.67	0.58
2	K – Nearest Neighbors	0.4704	0.47	0.44
3	Naïve Bayes	0.6033	0.60	0.56
4	Decision Tree Classifier	0.5133	0.51	0.52
5	Logistic Regression	0.6776	0.67	0.54
6	Support Vector Machine	0.6680	0.67	0.54
7	Proposed MST Algorithm	0.6681	0.8967	0.5917

On the other hand, other algorithms achieve only around 50% accuracy, recall, and precision. Additionally, Table 4 also compares the f1-score for sentiment classification.

This comparison highlights the strengths and weaknesses of each algorithm and can provide valuable insights into improving the accuracy and effectiveness of future models. Overall, the proposed architecture appears to have a higher recall rate, but there may be room for improvement in terms of accuracy and precision by incorporating data-cleaning techniques.

3) DIALECT CLASSIFICATION: COMPARISON BETWEEN PROPOSED MODEL AND OTHER MACHINE LEARNING MODELS

This section presents a comparison of different ML classifiers, namely K-Nearest Neighbors Classifier, Random Forest Classifier, Logistic Regression, and Naïve Bayes, for Arabic dialect classification. The performance of each classifier is evaluated using various metrics such as accuracy, recall, and F1-score, and the results are presented in Table 5. This comparison aims to determine the most effective classifier for Arabic dialect classification. The proposed model shows an accuracy of 67% and a recall of 90%, which outperforms the other ML algorithms. However, the precision of the proposed model is not reported. On the other hand, other algorithms achieve high accuracy but very low recall, indicating that they may not be effective in identifying all the dialects in the dataset. Therefore, the proposed model can serve as a baseline for future research on Arabic dialect classification.

Table 5 compares various ML algorithms for Arabic dialect classification. The results indicate that the proposed architecture performs better than other algorithms in terms of recall and F1-score, implying that it is better at accurately identifying Arabic dialects in the dataset. Other classifiers such as SVM, Logistic Regression, and Random Forest

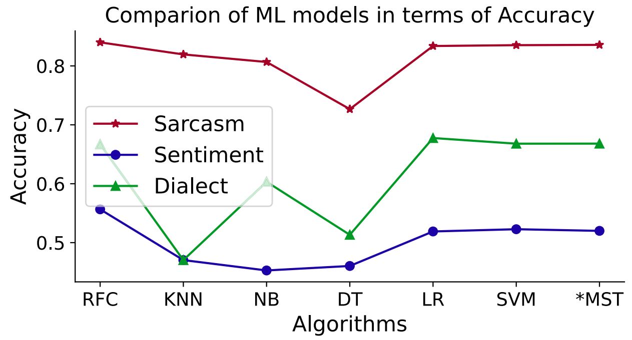

FIGURE 9. Comparison of ML models in terms of Accuracy of Sarcasm Detection, sentiment classification and Dialect Classification.

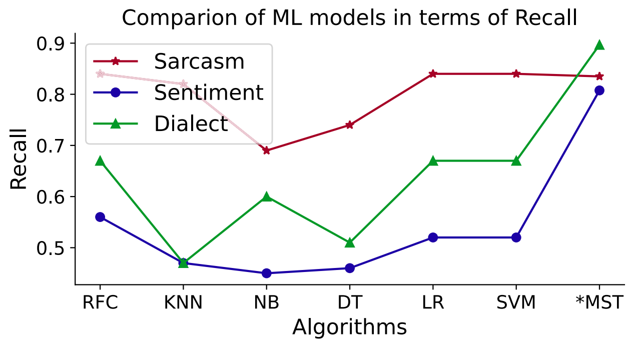

FIGURE 10. Comparison of ML models in terms of Recall of Sarcasm Detection, sentiment classification and Dialect Classification.

Classifier also perform reasonably well, achieving a recall rate of about 67%. However, the precision values for these algorithms are not reported in this comparison, which could affect their overall performance. In general, the proposed architecture appears to be a promising approach for Arabic dialect classification.

The outcomes of the study suggest that the proposed architecture could be a promising method for Arabic sentiment analysis, sarcasm detection, and dialect classification and could potentially become a standard for future research in this area. As depicted in Figure 9, the accuracy of the proposed model outperforms other ML models for all three tasks. Figure 10 and Figure 11 present a comparison of the proposed model with other state-of-the-art ML models in terms of recall and F1-score, respectively, highlighting the importance of the proposed model for detecting sarcasm, analyzing sentiment, and classifying dialects in Arabic text.

B. COMPARISON WITH OTHER STATE OF THE ART ARCHITECTURES

This section aims to compare the proposed MST (Modified Switch Transformer) with other state-of-the-art architectures in the literature. The comparison is based on factors like the algorithms' performance, efficiency, and accuracy. The main objective of conducting a comparative analysis is to provide an unbiased and comprehensive evaluation of

TABLE 6. Sarcasm Classification: Comparative analysis with existing state of the art models.

Author Name	Model Name	Accuracy	Recall	Precision	F1-Score
[70]	QARiB	-	0.690	0.734	0.551
[71]	MARBERT	-	0.714	0.714	0.584
[30]	ARABERT	0.77	-	-	0.75
[37]	LSVC	0.840	0.840	0.820	0.81
[38]	Ara-Bert	0.753	0.670	0.685	0.518
[41]	SalamBERT	0.772	0.680	0.712	0.534
[43]	MARBERT	0.832	0.771	0.724	0.74
Proposed	**MST**	**0.8357**	**0.8350**	**0.8310**	**0.8329**

TABLE 7. Sentiment Classification: Comparative analysis with existing state of the art models.

Author Name	Model Name	Accuracy	Recall	Precision	F1-Score
[70]	Mazajak	0.84	-	-	0.43
[70]	Mazajak	0.64	-	-	0.61
[22]	SAIDS	-	-	-	0.7598
[72]	BOW-LR	-	0.58	-	0.50
[72]	TF-IDF-LR	-	0.60	-	0.52
[73]	SVM	-	-	-	0.60
[72]	BERT	-	0.59	-	0.50
[72]	AraBert	-	0.66	-	0.59
[74]	BiLSTM	0.62	0.61	-	0.63
Proposed	**MST**	**0.5110**	**0.8076**	**0.4302**	**0.6798**

TABLE 8. Dialect Classification: Comparative analysis with existing state of the art models.

Author Name	Model Name	Accuracy	Recall	Precision	F1-Score
[70]	Mazajak	0.84	-	-	0.43
[70]	Mazajak	0.64	-	-	0.61
[22]	SAIDS	-	-	-	0.71
[75]	semi-supervised learning	0.68	-	-	-
[76]	LSTM	0.71	-	-	-
[76]	CNN	0.68	-	-	-
[76]	Bi-LSTM	0.70	-	-	-
[74]	MARBERT	0.67	-	-	-
Proposed	**MST**	**0.6681**	**0.8967**	**0.50**	**0.5917**

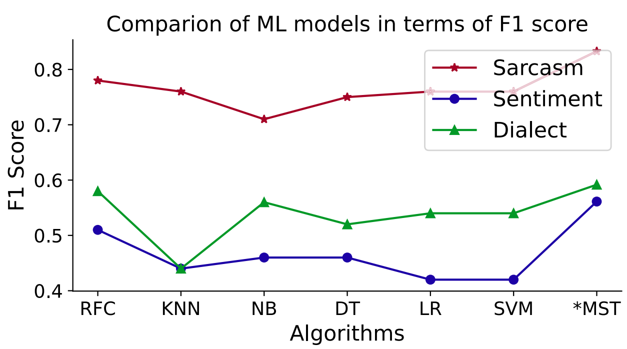

FIGURE 11. Comparison of ML models in terms of F1 score of Sracasm Detection, sentiment classification and dialect classification.

the systems being compared and to determine the best model for a specific problem. It is worth noting that the architectures presented in the tables and compared with the proposed MST are not trained alongside the MST. Instead, these models are derived from existing literature. In this section, we perform a comparison between the proposed MST and state-of-the-art architectures previously conducted by different researchers. By comparing different architectures, researchers can identify the strengths and weaknesses of each model and build on them to improve the performance of future models.

Table 6 compares the performance of the proposed MST architecture with other state-of-the-art architectures for Arabic sarcasm detection. The results show that the proposed model outperforms the other models in terms of accuracy, f1-score, recall, and precision, even without using any data cleaning processes or pre-trained embeddings. This comparison objectively evaluates the different architectures and highlights the effectiveness of the proposed MST architecture for the specific problem of Arabic sarcasm detection. The model achieves good recall for the sentiment classification task but very low accuracy and precision. This task involves classifying text into three types of sentiments - positive, negative, and neutral. The low performance of the proposed model may be due to the absence of a data-cleaning process. Table 7 presents a comparison of different state-of-the-art models for sentiment classification.

The comparison presented in Table 7 aims to identify the best-performing model for sentiment classification and provide guidance for the development of more accurate models in the future. After analyzing the table, it can be concluded that the proposed model achieves good recall, indicating that it effectively identifies instances of the sentiment classes. However, the proposed model has very low accuracy and precision, suggesting that it may misclassify some instances. On the other hand, the performance of the MST model can be improved by adjusting the hyperparameters. The proposed MST model's performance for dialect classification has been evaluated, and the results indicate a high recall of around 90% but a low f1-score. The task involves classifying text into five classes, making it a multi-class classification problem. The table presents a comparison of the proposed MST model's performance with other state-of-the-art models for dialect classification. This objective evaluation of the different models can help identify the best-performing model for dialect classification in Arabic text. Table 8 shows the comparison of MST with different dialect classification models.

The results of the proposed MST model for dialect classification have been analyzed and presented in above Table 8. The analysis shows that the proposed model has a high recall, but a low f1-score. This suggests that the model is good at identifying the correct dialect, but it struggles with precision. However, these findings provide an opportunity for researchers to modify the proposed MST model and optimize the hyperparameters to achieve more precise results in the future.

V. CONCLUSION

Text classification is a challenging problem in NLP and deep learning. In this research, we detect sarcasm and classify sentiment and dialect in the Arabic text dataset. We develop a novel switch transformer architecture (MST) with probabilistic projections and a new type of scheduling called Variational enmesh experts routing. The spatial gated unit with probabilistic projections refers to a gating mechanism that helps the network control the flow of information statistically and decide which features to use. The use of Variational enmesh experts routing allows the network to dynamically adjust the number of attention heads based on the length of the input sequence. The experts in routing mechanism are variational hierarchical experts, which are more non-linear and powerful expert. In this research, we used the ArSarcasm dataset of Arabic tweets to detect sarcasm and classify sentiment and dialect. The proposed model is a modified version of the switch transformer with variational touch. Our MST model achieved a recall of 84% for sarcasm detection, 90% for dialect classification, and 81% for sentiment classification. We performed a comparative analysis using various ML models, including Naive Bayes and Random Forest Classifier. Some future guidelines for improving the architecture include decreasing the parameter count, shortening computing time, and boosting overall performance.

REFERENCES

[1] M. S. Khan, A. Rizwan, M. S. Faisal, T. Ahmad, M. S. Khan, and G. Atteia, "Identification of review helpfulness using novel textual and language-context features," *Mathematics*, vol. 10, no. 18, p. 3260, Sep. 2022.

[2] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, Apr. 2019.

[3] S. Lyu and J. Liu, "Convolutional recurrent neural networks for text classification," *J. Database Manag.*, vol. 32, no. 4, pp. 65–82, Oct. 2021.

[4] T. Ahmad, M. S. Faisal, A. Rizwan, R. Alkanhel, P. W. Khan, and A. Muthanna, "Efficient fake news detection mechanism using enhanced deep learning model," *Appl. Sci.*, vol. 12, no. 3, p. 1743, Feb. 2022.

[5] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, "A survey on text classification: From traditional to deep learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, pp. 1–41, Apr. 2022.

[6] U. Khan, S. Khan, A. Rizwan, G. Atteia, M. M. Jamjoom, and N. A. Samee, "Aggression detection in social media from textual data using deep learning models," *Appl. Sci.*, vol. 12, no. 10, p. 5083, May 2022.

[7] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," *ACM Comput. Surv.*, vol. 50, no. 5, pp. 1–22, Sep. 2018.

[8] F. A. Abdulghani and N. A. Z. Abdullah, "A survey on Arabic text classification using deep and machine learning algorithms," *Iraqi J. Sci.*, pp. 409–419, Jan. 2022.

[9] M. Shrivastava and S. Kumar, "A pragmatic and intelligent model for sarcasm detection in social media text," *Technol. Soc.*, vol. 64, Feb. 2021, Art. no. 101489.

[10] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" in *Proc. China Nat. Conf. Chin. Comput. Linguistics*. Cham, Switzerland: Springer, 2019, pp. 194–206.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[12] G. Abercrombie and D. Hovy, "Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of Twitter conversations," in *Proc. ACL Student Res. Workshop*, 2016, pp. 107–113.

[13] F. Barbieri, F. Ronzano, and H. Saggion, "Italian irony detection in Twitter: A first approach," in *Proc. 1st Italian Conf. Comput. Linguistics*, 2014, pp. 28–32.

[14] E. Filatova, "Irony and sarcasm: Corpus generation and analysis using crowdsourcing," in *Proc. LREC*, 2012, pp. 392–398.

[15] D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1003–1012.

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

[17] I. A. Farha and W. Magdy, "From Arabic sentiment analysis to sarcasm detection: The ArSarcasm dataset," in *Proc. 4th Workshop Open-Source Arabic Corpora Process, Tools, Shared Task Offensive Lang. Detection*, 2020, pp. 32–39.

[18] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," 2002, *arXiv preprint cs/0205070*.

[19] M. Abdul-Mageed, M. Diab, and M. Korayem, "Subjectivity and sentiment analysis of modern standard Arabic," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol.*, 2011, pp. 587–591.

[20] D. M. El-Din Mohamed Hussein, "A survey on sentiment analysis challenges," *J. King Saud Univ.-Eng. Sci.*, vol. 30, no. 4, pp. 330–338, 2018.

[21] M. Bouazizi and T. Otsuki Ohtsuki, "A pattern-based approach for sarcasm detection on Twitter," *IEEE Access*, vol. 4, pp. 5477–5488, 2016.

[22] A. Kaseb and M. Farouk, "SAIDS: A novel approach for sentiment analysis informed of dialect and sarcasm," 2023, *arXiv:2301.02521*.

[23] M. El-Haj, "Habibi—A multi dialect multi national Arabic song lyrics corpus," Lancaster Univ., Tech. Rep., 2020.

[24] S. Khalifa, N. Habash, D. Abdulrahim, and S. Hassan, "A large scale corpus of Gulf Arabic," 2016, *arXiv:1609.02960*.

[25] F. Sadat, F. Kazemi, and A. Farzindar, "Automatic identification of Arabic dialects in social media," in *Proc. 1st Int. Workshop Social Media Retr. Anal.*, Jul. 2014, pp. 35–40.

[26] K. Egan, "Cross lingual Arabic blog alerting (COLABA)," in *Proc. 9th Conf. Assoc. Mach. Transl. Americas, Government MT User Program*, 2010.

[27] M. Abdul-Mageed, C. Zhang, H. Bouamor, and N. Habash, "NADI 2020: The first nuanced Arabic dialect identification shared task," in *Proc. 5th Arabic Natural Lang. Process. Workshop*, 2020, pp. 97–110.

[28] S. Gupta, R. Singh, and V. Singla, "Emoticon and text sarcasm detection in sentiment analysis," in *Proc. 1st Int. Conf. Sustain. Technol. Comput. Intell.* Cham, Switzerland: Springer, 2020, pp. 1–10.

[29] S. S. Birunda and R. K. Devi, "A review on word embedding techniques for text classification," in *Innovative Data Communication Technologies and Application*. Singapore: Springer, 2021, pp. 267–281.

[30] R. Duwairi and M. El-Orfali, "A study of the effects of preprocessing strategies on sentiment analysis for Arabic text," *J. Inf. Sci.*, vol. 40, no. 4, pp. 501–513, Aug. 2014.

[31] I. A. Farha and W. Magdy, "Benchmarking transformer-based language models for Arabic sentiment and sarcasm detection," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 21–31.

[32] M. M. Zain, R. N. Simbolon, H. Sulung, and Z. Anwar, "Analisis sentimen pendapat masyarakat mengenai vaksin COVID-19 pada media sosial Twitter dengan robustly optimized Bert pretraining approach," *Jurnal Komputer Terapan*, vol. 7, pp. 280–289, Dec. 2021.

[33] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2019, *arXiv:1911.02116*.

[34] H. E. Moubtahij, H. Abdelali, and E. B. Tazi, "Arabert transformer model for Arabic comments and reviews analysis," *Int. J. Artif. Intell.*, vol. 11, no. 1, pp. 379–387, 2022.

[35] S. Alrowili and V. Shanker, "ArabicTransformer: Efficient large Arabic language model with funnel transformer and ELECTRA objective," in *Proc. Findings Assoc. Comput. Linguistics*, 2021, pp. 1255–1261.

[36] A. Y. Muaad, H. J. Davanagere, J. V. B. Benifa, A. Alabrah, M. A. N. Saif, D. Pushpa, M. A. Al-Antari, and T. M. Alfakih, "Artificial intelligence-based approach for misogyny and sarcasm detection from Arabic texts," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, Mar. 2022.

[37] M. Lichouri, M. Abbas, B. Benaziz, A. Zitouni, and K. Lounnas, "Preprocessing solutions for detection of sarcasm and sentiment for Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 376–380.

[38] H. Elgabry, S. Attia, A. Abdel-Rahman, A. Abdel-Ate, and S. Girgis, "A contextual word embedding for Arabic sarcasm detection with random forests," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 340–344.

[39] I. A. Farha, W. Zaghouani, and W. Magdy, "Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 296–305.

[40] F. Husain and O. Uzuner, "Leveraging offensive language for sarcasm and sentiment detection in Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 364–369.

[41] B. Haidar, M. Chamoun, and A. Serhrouchni, "Multilingual cyberbullying detection system: Detecting cyberbullying in Arabic content," in *Proc. 1st Cyber Secur. Netw. Conf. (CSNet)*, Oct. 2017, pp. 1–8.

[42] A. Israeli, Y. Nahum, S. Fine, and K. Bar, "The IDC system for sentiment classification and sarcasm detection in Arabic," in *Proc. 6th Arabic Natural Lang. Process. Workshop*, 2021, pp. 370–375.

[43] M. Abdul-Mageed, A. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep bidirectional transformers for Arabic," 2020, *arXiv:2101.01785*.

[44] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcasm in Twitter and Amazon," in *Proc. 14th Conf. Comput. Natural Lang. Learn.*, 2010, pp. 107–116.

[45] O. Tsur, D. Davidov, and A. Rappoport, "ICWSM—A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 4, 2010, pp. 162–169.

[46] M. Khodak, N. Saunshi, and K. Vodrahalli, "A large self-annotated corpus for sarcasm," 2017, *arXiv:1704.05579*.

[47] E. Riloff, A. Qadir, P. Surve, L. D. Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 704–714.

[48] C. Van Hee, E. Lefever, and V. Hoste, "SemEval-2018 task 3: Irony detection in English tweets," in *Proc. 12th Int. Workshop Semantic Eval.*, 2018, pp. 39–50.

[49] I. A. Farha and W. Magdy, "The effect of Arabic dialect familiarity on data annotation," in *Proc. The 7th Arabic Natural Lang. Process. Workshop (WANLP)*, 2022, pp. 399–408.

[50] A. Yafoz and M. Mouhoub, "Sentiment analysis in Arabic social media using deep learning models," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 1855–1860.

[51] A. Mourad and K. Darwish, "Subjectivity and sentiment analysis of modern standard Arabic and Arabic microblogs," in *Proc. 4th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2013, pp. 55–64.

[52] K. Elshakankery, M. Fayek, and M. Farouk, "LASTD: A manually annotated and tested large Arabic sentiment tweets dataset," in *Proc. 5th Int. Conf. Inf. Syst. Data Mining*, May 2021, pp. 62–66.

[53] S. Kiritchenko, S. Mohammad, and M. Salameh, "SemEval-2016 task 7: Determining sentiment intensity of English and Arabic phrases," in *Proc. 10th Int. Workshop Semantic Eval.*, 2016, pp. 42–51.

[54] S. R. El-Beltagy, M. El Kalamawy, and A. Bakr Soliman, "NileTMRG at SemEval-2017 task 4: Arabic sentiment analysis," 2017, *arXiv:1710.08458*.

[55] J. Eisenstein, "Identifying regional dialects in on-line social media," *The Handbook of Dialectology*. Wiley, 2017, pp. 368–383.

[56] J. Pang, Y. Rao, H. Xie, X. Wang, F. L. Wang, T. Wong, and Q. Li, "Fast supervised topic models for short text emotion detection," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 815–828, Feb. 2021.

[57] D. Al-Ghadhban, E. Alnkhilan, L. Tatwany, and M. Alrazgan, "Arabic sarcasm detection in Twitter," in *Proc. Int. Conf. Eng. MIS (ICEMIS)*, May 2017, pp. 1–7.

[58] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALbert: A lite bert for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.

[59] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019, *arXiv:1907.11692*.

[60] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res.*, vol. 23, no. 1, pp. 5232–5270, 2022.

[61] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," 2019, *arXiv:1912.00741*.

[62] M. Nabil, M. Aly, and A. Atiya, "ASTD: Arabic sentiment tweets dataset," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2515–2519.

[63] M. Magris and A. Iosifidis, "Bayesian learning for neural networks: An algorithmic survey," *Artif. Intell. Rev.*, pp. 1–51, Mar. 2023.

[64] L. Vanneschi and S. Silva, "Bayesian learning," in *Lectures on Intelligent Systems*. Berlin, Germany: Springer, 2023, pp. 259–270.

[65] H. Liu, Z. Dai, D. So, and Q. V. Le, "Pay attention to MLPs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 9204–9215.

[66] T. Pranckevicius and V. Marcinkevicius, "Comparison of Naïve Bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification," *Baltic J. Modern Comput.*, vol. 5, no. 2, p. 221, 2017.

[67] X. Luo, "A new text classifier based on random forests," in *Proc. 2nd Int. Conf. Mater. Eng. Inf. Technol. Appl. (MEITA)*, 2017, pp. 290–293.

[68] X. Chen and H. Ishwaran, "Random forests for genomic data analysis," *Genomics*, vol. 99, no. 6, pp. 323–329, Jun. 2012.

[69] M. A. Chandra and S. S. Bedi, "Survey on SVM and their application in image classification," *Int. J. Inf. Technol.*, vol. 13, no. 5, pp. 1–11, Oct. 2021.

[70] I. A. Farha and W. Magdy, "Mazajak: An online Arabic sentiment analyser," in *Proc. 4th Arabic Natural Lang. Process. Workshop*, 2019, pp. 192–198.

[71] B. Talafha, M. E. Za'Ter, S. Suleiman, M. Al-Ayyoub, and M. N. Al-Kabi, "Sarcasm detection and quantification in Arabic tweets," in *Proc. IEEE 33rd Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2021, pp. 1121–1125.

[72] B. Alharbi, H. Alamro, M. Alshehri, Z. Khayyat, M. Kalkatawi, I. I. Jaber, and X. Zhang, "ASAD: A Twitter-based benchmark Arabic sentiment analysis dataset," 2020, *arXiv:2011.00578*.

[73] N. Al-Twairesh, H. Al-Khalifa, A. Al-Salman, and Y. Al-Ohali, "AraSenTi-tweet: A corpus for Arabic sentiment analysis of Saudi tweets," *Proc. Comput. Sci.*, vol. 117, pp. 63–72, Jan. 2017.

[74] I. A. Farha and W. Magdy, "A comparative study of effective approaches for Arabic sentiment analysis," *Inf. Process. Manag.*, vol. 58, no. 2, Mar. 2021, Art. no. 102438.

[75] F. Huang, "Improved Arabic dialect classification with social media data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2118–2126.

[76] L. Lulu and A. Elnagar, "Automatic Arabic dialect classification using deep learning models," *Proc. Comput. Sci.*, vol. 142, pp. 262–269, Jan. 2018.

S. MUHAMMAD AHMED HASSAN SHAH was born in Kamra, Attock, Pakistan, in 2001. He received the B.S. degree in computer science from COMSATS University Islamabad, Attock Campus, Pakistan, in 2022. He was a Research Assistant with the Medical Imaging and Diagnostic Laboratory, National Center of Artificial Intelligence, COMSATS University Islamabad. His research interests include computer vision, medical imaging, NLP, LLMs, quantum deep learning, graph neural networks, evolutionary optimization techniques, self-supervised contrastive learning, representation theory, generative adversarial, and Bayesian learning.

SYED FAIZAN HUSSAIN SHAH received the bachelor's degree in computer science from COMSATS University Islamabad, Attock Campus, in 2023. His research interests include natural language processing (NLP), data analysis, and data science.

ASAD ULLAH was born in Attock, in June 1999. He received the bachelor's degree in computer science from COMSATS University Islamabad, Attock Campus. His research interests include machine learning, deep learning, computer vision, neural networks, and 3D imaging.

ATIF RIZWAN received the B.Sc. degree from the University of the Punjab, Lahore, Pakistan, in 2015, and the M.C.S. and M.S. degrees in computer science from COMSATS University Islamabad, Attock Campus, Attock, Pakistan, in 2018 and 2020, respectively. He is currently a Ph.D. Researcher with the Department of Computer Engineering, Jeju National University, Jeju, Republic of Korea. He has good industry experience in mobile and web application development and testing. His research interests include applied machine learning, data and web mining, analysis, the optimization of core algorithms, federated learning, and the IoT-based applications. He was awarded a fully funded scholarship for the entire duration of his Ph.D. studies.

GHADA ATTEIA received the Ph.D. degree in electrical and computer and geomatics engineering from the Schulich School of Engineering, University of Calgary, Calgary, AB, Canada, in 2015. From 2010 to 2015, she was a Research Assistant with the Geomatics Engineering Department, University of Calgary. From 2017 to 2023, she has been an Assistant Professor with the Information Technology Department, CCIS, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia, where she is currently an Associate Professor. Her research interests include artificial intelligence, machine and deep learning, and new and renewable energy. She was a recipient of the Queen Elizabeth II Doctoral Award, in 2012 and 2013, and the L. R. (Dick) Newby Memorial Doctoral Award from the University of Calgary, in 2014.

MAALI ALABDULHAFITH was born in Saudi Arabia, in 1985. She received the Ph.D. degree in computer science from Dalhousie University, Halifax, Canada, in 2018. In 2014, she joined the College of Computer and Information Science (CCIS), Princess Nourah bint Abdulrahman University, as a Lecturer, where she was promoted to an Assistant Professor, in 2018, and is currently the Director of data management and performance measurement, overlooking and managing the strategy of the college. Her research interests include machine learning, data analytics, emerging wireless technology, and technology applications in health care.