AraSpam: A Multitask Deep Neural Network for Spam Detection in Arabic Twitter

Lulua Alhamdan[1], Ahmed Alsanad[2], Nora Al-Twairesh[3]
Department of Information Systems, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia[1]
College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia[1, 2, 3]

***Abstract*—Twitter has become widely used for disseminating information across the Arab world. It provides diverse communicative and informational needs while serving as a rich data source for a wide range of research. However, the integrity of such data is frequently undermined by the pervasive issue of spam. Existing research proposed the use of spam detection models at multiple levels—the account, tweet, and campaign levels. Many of these models target Uniform Resource Locator (URL)-based spam messages, whereas a significant portion of spam content operates without embedded URLs. Furthermore, spam detection methodologies tailored to the account level often lack the precision required for tweet-level analysis or, conversely, fail to capture broader account-level behavioral patterns. Moreover, studies focusing on Arabic spam have largely been restricted to specific geographical regions or linguistic varieties, such as Arabic dialect (AD) or Modern Standard Arabic (MSA), thereby neglecting the full spectrum of Arabic's linguistic diversity in spam messages. This study aims to address these limitations by proposing AraSpam, a multitask deep neural network that detects both spam messages and profiles using a single model. It was trained using a dataset of tweets written in AD and MSA covering different spamming targets. The text features were extracted using transformer-based models: AraBERT for tweet text and mBERT for profile screen name. The experiment demonstrated 96% accuracy in detecting both spam accounts and tweets with seven different spamming targets. Additionally, the experiments revealed that reducing the number of spam classes resulted in an increase in tweet detection performance and a decrease at the account level.**

***Keywords*—Spam detection; Twitter; multitask deep neural network; transformer-based model**

I. Introduction

Spam is generally referred to as unwanted and unsolicited material. It takes on several forms on different platforms, including email, Wikipedia, e-commerce, and online social networks (OSNs). Recently, artificial intelligence (AI) has enabled multimedia spam and social spambots to generate automatic content [1]. Users who send spam messages are known as spammers. They could be individuals, campaigns, or social bots. They are typically motivated by a clear objective, concentrating on its attainment [2][3].

Twitter is recognized as an important OSN platform for daily activities and is employed by individuals, organizations, and researchers for information dissemination and analysis. It allows users to share messages known as tweets. A tweet is generally limited to 280 characters, including text, Uniform Resource Locators (URLs), multimedia, emojis, hashtags, and mentions. Recently, users who subscribe to premium accounts can share messages that are extended to a maximum of 25,000 characters. Twitter defines spam as "*sharing or posting content in a bulk, duplicative, irrelevant, or unsolicited manner that disrupts people's experience* [4]". The powerful information-sharing capabilities of Twitter, including hashtags, mentions, and retweets, attract spammers who exploit the platform for their activities. Commonly, spam is produced on Twitter through mentions of well-known users' accounts, replies to their tweets, or wildly trending hashtags [5][6].

Different types of spammers with various intents have been observed. On the political side, some campaigns disseminate agendas to influence people's opinions during critical periods of political significance [7][8]. Commercially, in addition to standard advertisement spam, spammers try to influence people's opinions toward specific products or services by writing fake reviews [9][10]. In cybersecurity, Twitter has become a target for attackers seeking to spread malware [11][12]. Additionally, illegal and pornographic materials are also propagated on Twitter [13][14]. Accordingly, these spamming activities are decreasing the quality of Twitter data. The decrease in data quality affects researchers by requiring additional cleaning effort before performing various tasks, including data analysis, training natural language processing (NLP) models, and others [9][15]. Twitter has implemented restrictions to ensure the reliability of its content and to prevent any deceptive practices, thereby increasing its audience's confidence in its authority [4]. However, these regulations and rules lack detection capabilities except for limited violations. Researchers have proposed different detection approaches for spam tweets, profiles, campaigns, and bots. Most rely on supervised machine learning (ML) techniques using single-type or hybrid features, including text, content, account, and graph features [16][17]. The deep learning (DL) approach eliminates the need for feature selection; however, a large labeled dataset is required for improved performance [18]. Even if high accuracy is achieved through these approaches, spammers continue to change their behaviors and challenge the detection process, a problem referred to as spam drift [19]-[21]. Regarding detection level, tweet-level detection lacks the ability to detect spamming behaviors, whereas account-level detection models cannot identify spam tweets generated from legitimate accounts. On the other hand, campaign- and bot-level detection models are unable to detect individual and ordinary spammers' accounts, respectively [16].

In Arabic language, specifically, various studies have revealed that the effectiveness of the features in the detection process is language-dependent and differs between societies [22]-[25]. At the same time, most of the literature is based on the English language [25]-[27]. Among the limited Arabic studies, the majority focused on content in Saudi Arabia [10].

This study aims to enrich Arabic Twitter spam detection by addressing different limitations. First, using a more comprehensive dataset that covers Arabic spam messages, even written in Modern Standard Arabic (MSA) or Arabic dialect (AD). Additionally, in order to eliminate the spam drift problem, the data was collected using different time slots, hashtags, user mentions, and keywords. Furthermore, unlike previous works that automatically assigned the same label to both accounts and tweets, tweets and accounts were labeled independently to highlight instances where legitimate accounts and vice versa occasionally wrote spam messages. Moreover, to reveal the purpose of spamming, spam tweets were categorized based on the most common spamming purposes on Arabic Twitter. Finally, a detection model is proposed to classify both tweets and user profiles within a single model by leveraging the concept of a multitask learning framework. Accordingly, this will help in revealing spam messages generated from legal accounts and vice versa. Furthermore, by leveraging the similarities between the two tasks, we can reduce the processing time.

The rest of this study is structured as follows: Section II presents a review of related work on Twitter spam detection. Section III provides background on the techniques employed in this study and outlines the proposed methodology. Section IV presents and discusses the experimental results. Section V presents the conclusion, summarizing the key findings and highlighting the overall significance of the study.

II. Related Work

Various methods have been proposed for spam detection on Twitter, each implemented through different techniques. Recently, the primary approaches in the literature are ML and DL, applied at different levels, including tweets, accounts, campaigns, and social bots. ML-based studies vary in their choice of features, with some focusing on a single feature type, such as text, content, user, or graph, while others adopt hybrid features. The next section explores studies employing tweet- and account-level detection approaches, with a dedicated section on Arabic spam detection research.

A. *Tweet-Level Detection*

First, Tweet-level detection involves classifying tweet messages into spam or non-spam. Studies on the ML approach most often use content features, as classified by [19], into URL or hashtag features, image or video features, and text-based features. Other studies considered user account features in tweet classification, such as in [28], which proposed a framework based on ML with a combination of user and tweet features that were available for real-time detection, such as the number of characters, digits, hashtags, and followers.

Recent studies have proposed extracting text features using various methods, including term frequency–inverse document frequency (TF–IDF), bag-of-words, n-gram, and Word2Vec. The authors of [18] proposed using Word2Vec to learn message syntax with a multilayer perceptron artificial neural network (MLP) that outperforms other feature-based ML classifiers such as random forest (RF) and naïve bayes (NB). The authors of [29] presented another means of outperforming these methods, using bidirectional long short-term memory networks (BiLSTMs) to extract text features for training supervised ML classifiers.

Instead of depending on feature-based ML classifiers, a variety of studies proposed models based on deep neural networks. Neha proposed using long short-term memory (LSTM) after extracting text features by applying Global Vector (GloVe) word embeddings [30], achieving an accuracy of approximately 95%. Moreover, [31] used the Word2Vec embedding for training an extreme learning machine (ELM) with multiple layers, achieving an accuracy of 88%. The use of transformer-based models for generating embeddings was applied by [32] for detecting spam in a short message service (SMS) dataset. This is accomplished by extracting text features using Generative Pre-trained Transformer 3 (GPT-3) and ensemble classifiers with weighted voting for classification.

B. *Account-Level Detection*

Account-level detection involves classifying a user account as a spammer or a legitimate user. Most studies accomplished it using ML with a combination of account, content, and graph features. The majority used statistical features, such as the ratio between the number of followers and the number of friends, the average number of favorite tweets, tweeting frequency, and account age [33]. In addition to these features, other studies have focused on interaction features in relation to user and content features [34]. Another study [35] analyzed URLs and sharing patterns to detect spam accounts. Detecting spammers' behaviors and extracting useful behavioral features have led researchers to perform network analyses and utilize graph features, such as average neighborhood followers, average neighbor tweets, and betweenness centrality of mentions [36]. Instead of relying on such features, [37] utilized a CNN to combine text features extracted using Word2Vec with common account numerical features, such as account age, number of followers and followings, and others. In various studies, suspicious words were combined with other features to characterize spammer accounts. Some approaches involve checking for the presence of words from a predefined dictionary [23], whereas others extract statistical features based on these words [36]. The authors of [38] utilized additional linguistic features by targeting spammers based on their topics of interest, arguing that information from individuals discussing various topics is less trustworthy than information from those focused on specific areas.

All previous approaches required a labeled dataset for Twitter accounts; however, complete profile datasets were not publicly available due to privacy concerns [39]. Several studies have proposed using the unsupervised approach to mitigate the cost of manual annotation [40]-[42]. Instead of detecting individual human spam accounts, various studies focused on detecting automated spam via social bots [49][50], while others were more interested in identifying groups of spammers forming a campaign [43]-[46].

C. Spam Detection on Arabic Twitter

Several studies have focused on determining the effect of the language used on spam classification. For example, [22] used an ML classifier on two datasets: English and Roman Urdu. They observed that the results improved when a language-dependent model was used. Another study, by the authors of [25], examined the differences between spammers in various social contexts that affect the effectiveness of selected features. They built a model for detecting Arabic spammers and tested it with two datasets—Arabic accounts and English accounts—and concluded that the accuracy decreased in detecting English spammers. The authors of [23] tested the effectiveness of features in English, Arabic, Korean, and Spanish datasets and found that the behavior of spammers and the effectiveness of features differ between these datasets. These findings motivated researchers to consider lingual and social contexts in detection techniques. In Arabic spam tweets, the authors of [26] extracted text features using skip-grams and continuous bag of words (CBOW) to train ML classifiers and reached 87.3% accuracy with skip-grams learned using a corpus collected from Twitter. Another study examined tweet sentiment in conjunction with behavioral features to investigate the likelihood of a tweet being spam [27]. They found that the effectiveness of commenting behavior exceeds that of liking behavior in the classification process, whereas message sentiment is not a useful factor. Other researchers classified spam tweets in the Gulf Arabic dialect using content-based features with ML classifiers and attained 86% accuracy. Moreover, they stated that URL safety is insufficient in detecting Arabic spam and that the safety of the used words contributes more to the detection process [7]. The authors of [47] focused on spam in Saudi Arabia. They generated eight datasets, each collected from specific hashtags, which fall under three topics: national, health, and politics. After preprocessing, N-grams and Word2Vec embeddings were used for text feature extraction. These features were used to train different ML classifiers for the eight datasets separately, showing high performance using RF with N-grams. Moreover, ML classifiers were used in [48] with word embeddings and optimized by an augmentation technique that is based on Aravec's similarity. Then, these embeddings were combined with different features from three categories: content, user, and interaction. Hence, [10] focused on detecting advertisement spam in Arabic tweets by testing two methods: word embeddings with an ML classifier and fine-tuning AraBERT, a pre-trained Arabic language model. The latter achieved high accuracy compared with the former method. Moreover, the n-gram embeddings performed better than the Mazajak embeddings in the classification process. Another study [49] distinguishes between three types of spam: promotional, phishing, and spam. The last two categories are characterized based on the used URL: phishing if the URL directs to a trading website, and spam if it links to external websites. In their model, text features were extracted using Aravec for word embeddings in one model, whereas a character-level convolutional network was used for character embeddings in the other model. They integrated CNN for feature extraction with LSTM for spam classification. Their experiments demonstrated higher classification accuracy using word embeddings compared to character embeddings. In addition to regular text, [50] focused on classifying spam text within images. This was completed using an efficient and accurate scene text detector (EAST) and a convolutional recurrent neural network (CRNN) for text extraction .Then, the detection is performed by conducting comparisons between used words and two lists built by the authors: blacklist and whitelist. This method requires a big dataset in one language to avoid misclassification.

Moving to account-level detection, the authors of [25] classified Arabic spammers with 92.6% accuracy using 14 behavioral and content features selected by information gain (IG) and chi-squared feature selection methods. In addition to their previous finding that classification features are socially dependent, they observed how spammers employed different evasion techniques that may diminish the effectiveness of detection features over time. Similarly, the authors of [51] selected 16 features from a set of content- and user-based features and calculated different statistics for a subset of these features, including (total, minimum, maximum, and average). The rank of these features was determined using mean decrease Gini and subsequently reduced through the application of recursive feature elimination (RFE). Starting with about 47 features, more than 90% accuracy was achieved with 16 selected features. Comparing ML classifiers and deep neural networks (DNNs), the authors of [52] compared the two methods using text features. The features were extracted using N-grams (uni-gram, bi-gram, and char-gram) in the former and GloVe and fastText in the latter. DNN and GloVe outperformed other techniques in ML and DL models, respectively. Adapting the methodology used in [37], the authors of [53] proposed combining text features with metadata from tweets and accounts. Text features were extracted using word embeddings to train the CNN model, whereas a simple neural network was trained using 12 statistical features. The final classification is defined by combining the outputs of the two networks. Their framework achieved 94.27% accuracy in classifying Arabic spammer accounts. In addition to their proposed model for classifying spam tweets, the authors of [49] proposed another model for detecting Arabic spammers. After scraping a subset of tweets from the account, the embeddings were extracted (character level and word level) and used to train a deep learning model (CNN and LSTM), concluding that 10 to 15 tweets are optimal for high-performance classification. On the other hand, the authors of [14] focused on detecting types of spammers, namely those publishing porn content. Accordingly, different text features were tested individually and in combination. These features included username, screen name, user description, or a single tweet message extracted from the account. Text features were extracted from these attributes using n-gram comparison between word-level and character-level. Then, a Support Vector Machine (SVM) was trained using different alternatives of features. Other experiments involve fine-tuning pretrained language models, including Multi-BERT and AraBERT. These experiments revealed that screen names are not informative in the detection model, even with preprocessing and normalizing the used text. Moreover, the performance was highly improved by adding a single tweet with the username and description.

A review of the existing literature reveals several limitations that require further investigation. First, most studies rely on a single criterion for spam collection, such as URLs, hashtags, account suspension, or predefined keywords. These methods restrict the diversity of captured spamming behaviors and limit model generalizability. In addition, datasets are often collected within short time frames and annotated using automated or semi-automated techniques, which increases sample homogeneity and labeling inaccuracies.

Second, prior studies adopt a binary classification model, typically distinguishing only between spam and non-spam messages, focusing on narrow categories such as URLs, advertisements, or pornographic content. Such approaches fail to capture the full spectrum of spamming behaviors, limiting the generalizability and robustness of the resulting models.

Third, previous studies often consider either the tweet level or the account level in isolation, operating under the assumption that spam tweets are produced by spammer accounts and that non-spam tweets are generated only by legitimate accounts. This assumption overlooks more scenarios, such as legitimate accounts posting occasional spam messages or spammer accounts generating legitimate content to evade detection techniques.

Finally, feature-based approaches often require extensive and continuously available user, content or network data, which may not always be accessible due to privacy restrictions, platform limitations, or availability.

III. Materials and Methods

A. Dataset

Data was gathered utilizing the Twitter Application Programming Interface (API) across several time intervals from 2018 to 2023. We employed a comprehensive array of keywords, encompassing user mentions, hashtags, and suspicious terms, to effectively identify diverse spamming targets. The data was manually annotated to classify user profiles as either legitimate or spammer, with spam tweets categorized into seven distinct spamming targets: commercial advertisements, medical advertisements, illegal services, sorcery content, pornographic content, monetary requests, and other irrelevant messages (TABLE I.). The annotation process was conducted manually by two annotators, with an auditor responsible for assigning the final labels. Both annotators are native Arabic speakers with expertise in computer information systems and were provided with detailed annotation guidelines to ensure accurate labeling of spam tweets and accounts. Inter-annotator consistency was evaluated using Cohen's Kappa (κ), yielding a value of $\kappa = 0.92$ at the tweet level and $\kappa = 0.9$ at the account level. Three versions were derived from this dataset as follows:

- DS_Multi_8: This version of tweet labels encompasses all seven categories of spam targets (Fig. 1), as well as the ham class.
- DS_Multi_4: In this version, the seven spam categories have been consolidated into three abstract categories: promoting spam (commercial advertisements, medical advertisements, and illegal services), unethical spam (sorcery content and pornographic content), and irrelevant spam (monetary requests and irrelevant content).
- DS_Binary: This version includes a binary label for tweets, denoting whether they are classified as spam or not.

The dataset comprises more than 15,000 records, with approximately 54% being spam tweets and the remaining 46% being non-spam tweets. As previously mentioned, different keyword types were used to collect spam tweets, with varying effectiveness. A variety of hashtags indicating Arabic country names were used. Of tweets using these hashtags, 59% were ham messages, and 41% were spam messages. The majority of tweets using sports-related hashtags were spam (68% vs. 32% ham). Trending hashtags were particularly targeted by spammers, with only 28% of tweets being ham and 72% being spam messages. User mentions were also analyzed, particularly replies to various Arabic news accounts (@AJABreaking, @AJArabic, @AlArabiya_Brk, @alqabas, @SaudiNews50, and @Sabqorg); 77% of these tweets were ham messages, while 23% tweets were spam messages. Finally, spam words associated with different spam types were examined. Only 5% of these tweets were ham messages, while 95% were spam.

At the account level, the dataset comprised 5,620 unique user accounts. Of these, 55% were identified as legitimate and active during the annotation process, while 36% were classified as active spammers. In addition, 4% of the accounts were suspended, and 5% had been deleted at the time of annotation.

TABLE I. Spam Tweet Categories

Category	Definition
Commercial Advertisement	Tweets promoting products or services other than verified account advertisements.
Illegal Medicine Advertisement	Tweets promoting medical products or services (e.g., weight loss products, sexual products).
Illegal Service	Tweets promoting illegal services, including those prohibited by the government, or inflating OSN metrics (e.g., number of followers, retweets, verification).
Money Request	Tweets asking for monetary help.
Sorcery Service	Tweets promoting sorcery services.
Pornographic Content	Tweets with pornographic content, including text or media.
Irrelevant Content	Tweets using hashtags or mentions unrelated to the tweet topic or tweets with only hashtags, mentions, or media.

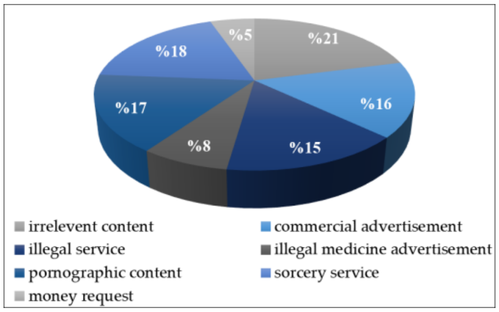

Fig. 1. Spam tweets per spam category.

B. AraSpam: Spam Detection Model Based on Multitask Learning

The AraSpam model is built to detect both spam tweets and accounts in Arabic Twitter. The model begins by extracting a textual representation from the preprocessed tweet text and the profile screen name. The two extracted embeddings were subsequently stacked, supplemented by an oversampling technique, and then input into the multitask deep neural network. Two deep multitask neural network architectures were employed: hard parameter sharing and soft parameter sharing.

1) Text preprocessing: The preceding phase of embedding extraction from tweet text is preprocessing. The text was preprocessed with AraBERT Preprocessor following its default settings:

- Keeping emojis;
- Removing "html" markup;
- Replacing email, URL, and user mentions with specific tokens as follows:
- Replacing URL with [رابط], which means [link];
- Replacing user mention with [مستخدم], which means [user];
- Replacing email with [بريد], which means [email];
- Removing the repetition of more than two non-digit characters, e.g., (سلااام) will be converted to (سلام);
- Stripping Tatweel (-), e.g., (محمـــد) will be converted to (محمد);
- Removing diacritics: Dammah (◌ُ), Tanween Aldam (◌ٌ), Kasra (◌ِ). Tanween Alkasr (◌ٍ), Fatha (◌َ), Tanween Alfath (◌ً), Skoun (◌ْ), and Shaddah (◌ّ);
- Replacing slash (/) with dash (-);
- Mapping Hindi numbers (١ ٢ ٣) to Arabic numbers (1 2 3);
- Inserting white space.

2) Text features extraction: Bidirectional Encoder Representations from Transformers (BERTs) is a transformer-based model that acquires linguistic representations by utilizing extensive text corpora through unsupervised methods such as masked language modeling (MLM) [54]. It can be fine-tuned to solve different NLP tasks, such as sentiment analysis and question answering. The model's input embeddings are the result of integrating three types of embeddings: token embeddings, segmentation embeddings, and position embeddings. Another version of BERT, trained on a Wikipedia corpus comprising 104 languages, is referred to as multilingual BERT (mBERT) [55]. It possesses advantages such as the tokenization method that can handle out-of-vocabulary words and its capacity to capture cross-lingual representation. An Arabic adaptation of the BERT model is AraBERT [56], which is trained using a large Arabic corpus and comes in different versions that differ based on the type and size of the training datasets, as shown in TABLE II. The original AraBERT dataset was crawled from Wikipedia and multiple Arabic corpora, including news articles from various Arab countries, covering a broad spectrum of topics written in Modern Standard Arabic (MSA). The latest versions incorporate tweets in diverse Arabic dialects, including emojis.

Sentence Transformers are a distinct category of transformer-based models designed to provide embeddings for sentences, thereby enabling effective evaluation of their semantics. Unlike traditional transformer models like BERT, which generate contextualized embeddings for individual tokens, Sentence Transformers construct a fixed-length vector representation for entire sentences. Sentence-BERT (SBERT) [57] and Language-agnostic BERT Sentence Embedding (LaBSE) [58] are transformer-based models built to provide embeddings for sentences or text. The last one was expanded to accommodate cross-lingual tasks, having been trained in 109 languages. Another sentence transformer mode supporting multilingual is E5 [59]. It generates dense, high-dimensional embeddings in which semantically similar inputs, regardless of language, are situated closer together in the embedding space. Accordingly, these models are highly effective in capturing semantic similarity, information retrieval, translation, and a range of other applications.

TABLE II. ARABERT MODEL VERSIONS [60]

Model	Pre-Segmentation (Farasa)	Dataset (Sentences/Size/nWords)
AraBERTv1-base	Yes	77M / 23GB / 2.7B
AraBERTv0.1-base	No	77M / 23GB / 2.7B
AraBERTv2-base	Yes	200M / 77GB / 8.6B
AraBERTv2-large	Yes	200M / 77GB / 8.6B
AraBERTv0.2-base	No	200M / 77GB / 8.6B
AraBERTv0.2-large	No	200M / 77GB / 8.6B
AraBERTv0.2-Twitter-base	No	AraBERTv0.2 dataset + 60M Multi-Dialect Tweets
AraBERTv0.2-Twitter-large	No	AraBERTv0.2 dataset + 60M Multi-Dialect Tweets

AraSpam utilizes word embeddings derived from two sources: preprocessed tweet text and user screennames, comparing between word- and sentence-level transformers as follows:

- Word level: extracting word embedding from preprocessed tweet text via AraBERTv0.2-Twitter-large and user screen names through mBERT.
- Sentence level: Three different alternative models were used for extracting sentence-level embeddings for both tweet text and account screen names, including SBERT, LaBSE, and E5.

3) Data resampling: To address the class imbalance problem, augmentation strategies involve either oversampling or undersampling. These strategies primarily aim to mitigate bias in the learning process against the majority class. The undersampling technique involves eliminating instances from the majority class, whereas oversampling approaches aim to augment the samples in the minority class, either randomly or through synthetic methods. One of these is the adaptive synthetic (ADASYN) sampling approach. It aims to account for the more challenging samples that are identified throughout the augmentation process [61].

DS_Multi_8 and DS_Multi_4 exhibit an imbalance among the various spam tweet classifications. The ADASYN augmentation approach was employed to address the dataset imbalance.

4) Multitask deep neural network: A DNN is an ML approach that can process raw data, such as text, and derive the necessary representations for classification tasks. The DNN model comprises an artificial neural network with multiple hidden layers (Fig. 2). Every neuron in a layer takes inputs, assigns weights and bias, processes the sum through an activation function, and transmits the output to the subsequent layer [62]-[64], as in Eq. (1):

$$z = wx + b \tag{1}$$

where, z is the output, w is the weight, x is the input matrix, and b is the bias. The activation function can be one of several functions, mainly Sigmoid [Eq. (2)], Rectified Linear Unit (ReLU) [Eq. (3)], and Softmax [Eq. (4)], for multi-classification problems. The equations of these functions are as follows:

$$\text{Sigmoid: } \sigma(z) = \frac{1}{1+e^{-z}} \tag{2}$$

$$\text{ReLU: } f(z) = \max(0, z) \tag{3}$$

$$\text{Softmax: } \sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{4}$$

Finally, the loss function [Eq. (5)] calculates the difference between the predicted value $(\hat{y}_i)$ and the actual value (y_i) as follows:

$$L = -\sum_i y_i \log(\hat{y}_i) \tag{5}$$

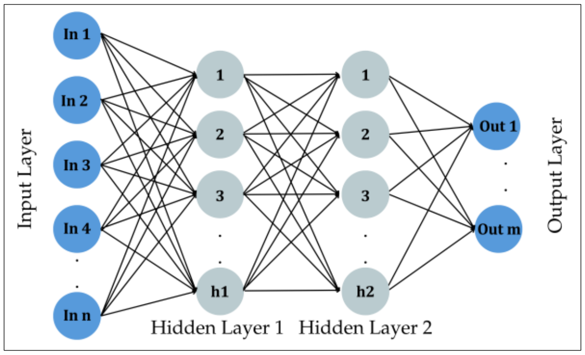

Fig. 2. General DNN structure.

Multitask learning (MTL) is a novel learning paradigm that leverages human learning to enable computers to learn multiple related tasks, thereby enhancing generalization performance [65]. At the beginning of MTL, the idea was motivated by the need to address the data scarcity problems that arise when different tasks have insufficiently labeled data for training. MTL combines these data to increase the model accuracy [66]. Formally, MTL is defined as "Given m learning tasks $\{\mathcal{T}\}_{i=1}^{m}$, where all the tasks or a subset of them are related, multi-task earning aims to learn the m tasks together to improve the learning of a model for each task $\mathcal{T}$i by using the knowledge contained in all or some of the other tasks [66]".

MTL tasks can be homogeneous, meaning all tasks share the same learning type, such as classification, regression, clustering, or others. Conversely, it could be heterogeneous if the learning tasks differ [67]. Joining the learning process between these tasks can be accomplished using different approaches, including feature learning, low-rank decomposition, task relationship learning, and task clustering. These approaches can be applied to tasks involving various learning types, including supervised, semi-supervised, unsupervised, reinforcement, online learning, and multi-view learning [66][68][69]. On the other hand, when using deep MTL, different methods of task concatenation can be employed, such as hard and soft parameter sharing, as shown in Fig. 3. The different tasks share the hidden layers in hard parameter sharing, while a constraint is added to encourage similarities among related parameters.

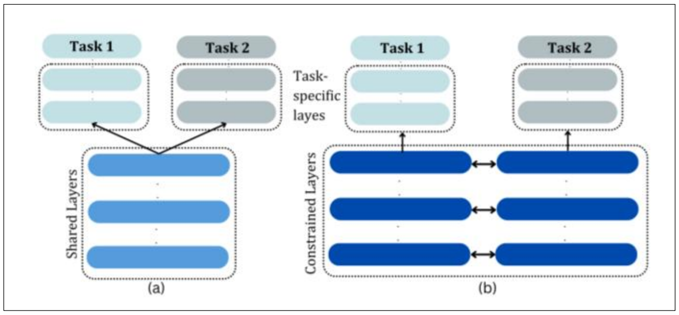

Fig. 3. Deep multitask learning approaches: (a) Hard parameter sharing, and (b) Soft parameter sharing.

a) Hard parameter sharing approach: The input layer functions as an embedding layer, as outlined in the preceding section for the tweet classification task. The identical layer is used for the account classification task after it is combined with the output of the tweet classification task layers. The tasks share certain hidden levels, whereas each task possesses additional task-specific layers, as shown in Fig. 4.

b) Soft parameter sharing approach: This architecture maintains the same input layer as the previous one, but the organization of the hidden layers differ. This method incorporates distinct hidden layers for each task, as illustrated in Fig. 5.

C. Experimental Setup

The experiments were conducted using Keras, an API integrated within the TensorFlow backend for implementing deep learning models. Consequently, the Keras tuner was utilized with 100 epochs to identify the optimal hyperparameters, including the number of units in the dense layer, dropout rate, and learning rate. Moreover, the Adam optimizer, L2 regularization (with a coefficient of 0.01), batch normalization, ReLU activation function, and cross-entropy loss function were used. Furthermore, transformer-based models were imported from the HuggingFace Transformers library. The experiments were conducted using Google Colab, equipped with high RAM and robust GPUs.

D. Performance Measures

The models' performances were measured using accuracy [Eq. (6)], macro F1-score [Eq. (7)], precision [Eq. (8)], and recall [Eq. (9)] as follows:

$$\text{Accuracy} = \frac{\text{True Positives + True Negatives}}{\text{True Positives + True Negatives +False Positives + False Negatives}} \qquad (6)$$

$$\text{F1} - \text{Score} = \frac{\text{False Positives}}{\text{True Positives+False Negatives}}\square \qquad (7)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positive}} \qquad (8)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}} \qquad (9)$$

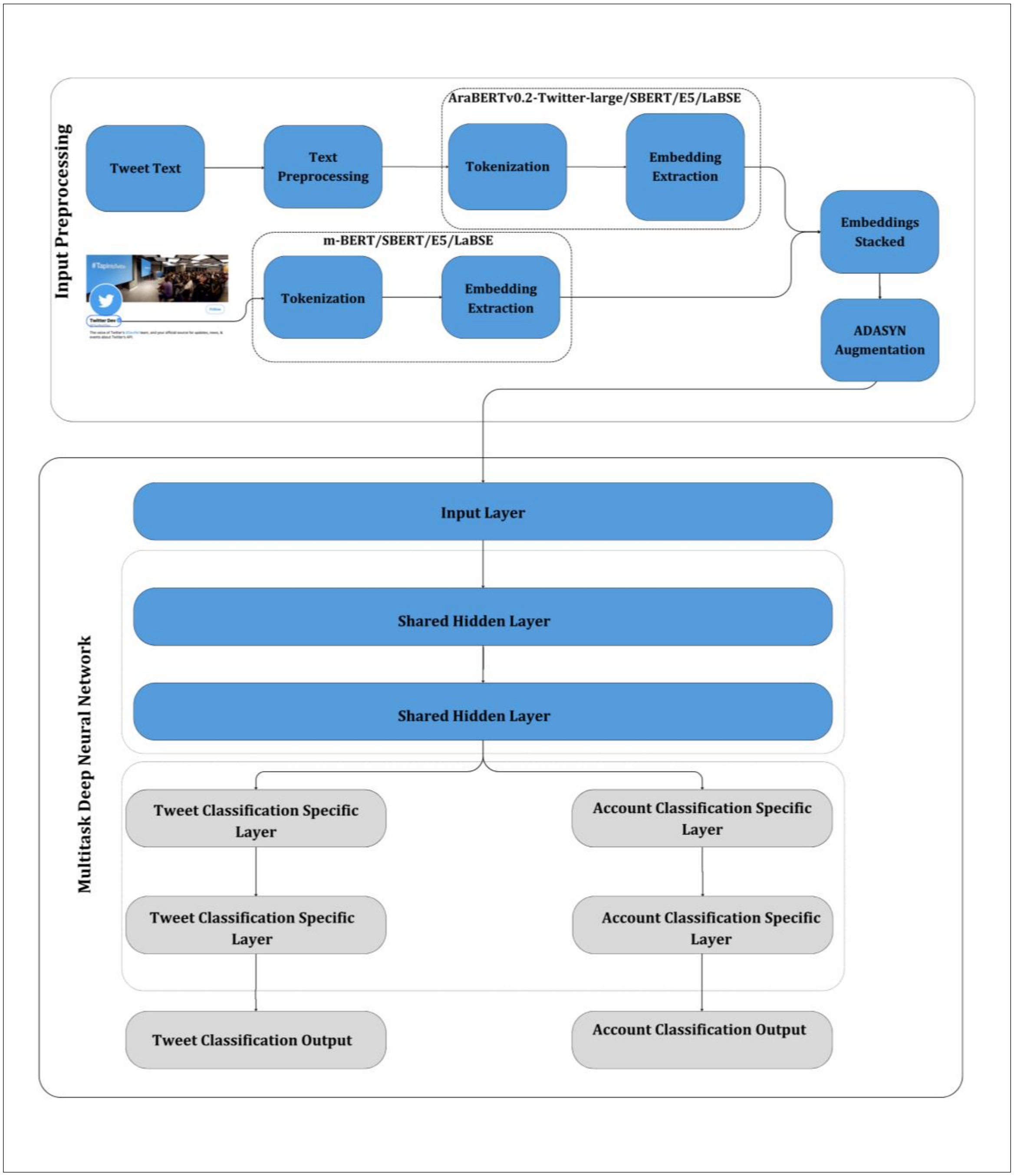
Input Preprocessing
Tweet Text
Text Preprocessing
AraBERTv0.2-Twitter-large/SBERT/E5/LaBSE
Tokenization
Embedding Extraction
m-BERT/SBERT/E5/LaBSE
Tokenization
Embedding Extraction
Embeddings Stacked
ADASYN Augmentation
Multitask Deep Neural Network
Input Layer
Shared Hidden Layer
Shared Hidden Layer
Tweet Classification Specific Layer
Account Classification Specific Layer
Tweet Classification Specific Layer
Account Classification Specific Layer
Tweet Classification Output
Account Classification Output

Fig. 4. AraSpam: Arabic spam detection model based on hard parameter sharing.

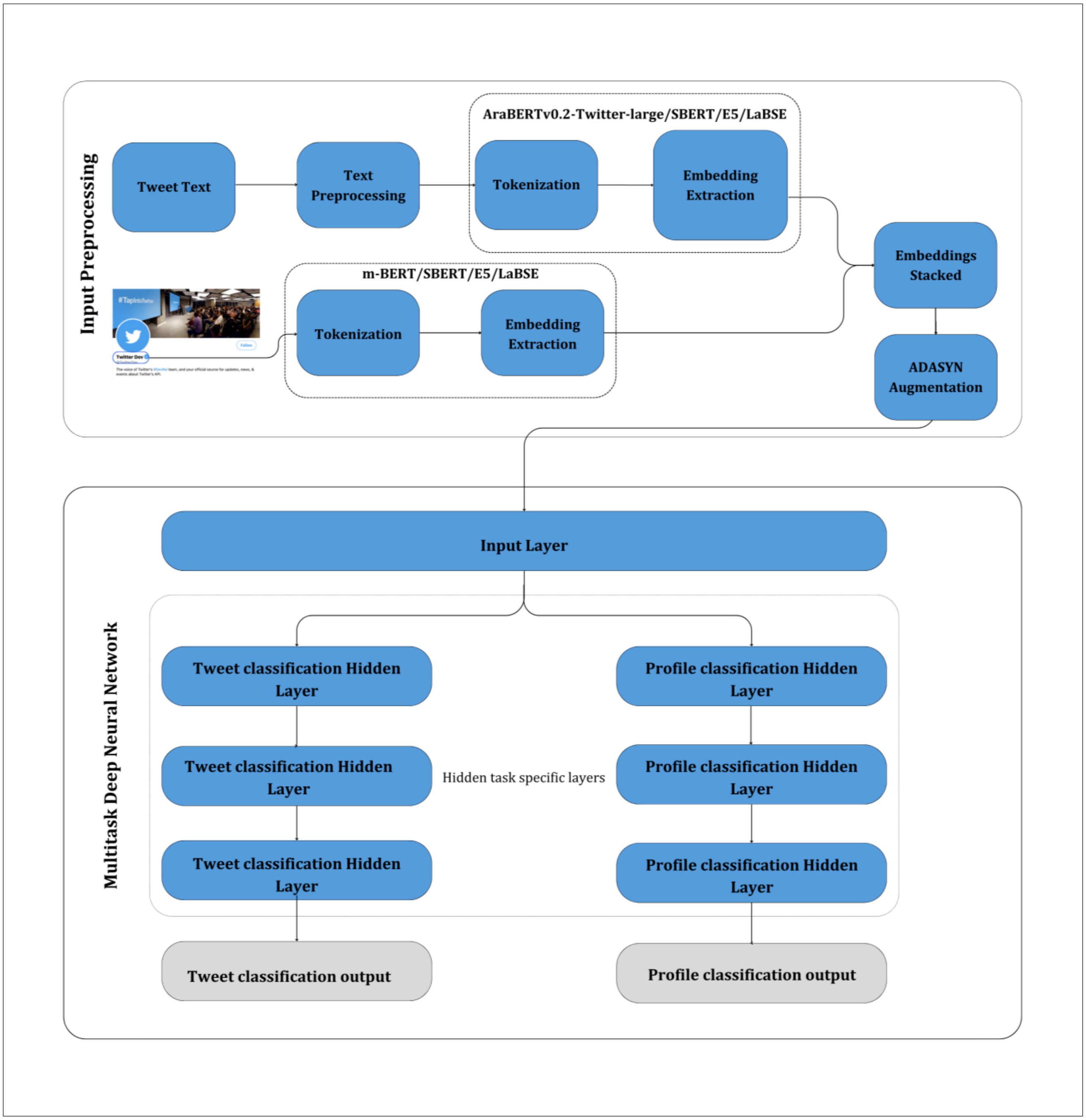
Input Preprocessing
Tweet Text
Text Preprocessing
AraBERTv0.2-Twitter-large/SBERT/E5/LaBSE
Tokenization
Embedding Extraction
m-BERT/SBERT/E5/LaBSE
Tokenization
Embedding Extraction
Embeddings Stacked
ADASYN Augmentation
Multitask Deep Neural Network
Input Layer
Tweet classification Hidden Layer
Tweet classification Hidden Layer
Tweet classification Hidden Layer
Hidden task specific layers
Profile classification Hidden Layer
Profile classification Hidden Layer
Profile classification Hidden Layer
Tweet classification output
Profile classification output

Fig. 5. AraSpam: Arabic spam detection model based on soft parameter sharing.

E. Model Validation

Various prior studies indicated that integrating text features with metadata enhances the accuracy of spam detection models [37][53]. AraSpam aims to achieve superior detection efficiency at both tweet and account levels with minimal prerequisites by eliminating the need to obtain metadata from Twitter. The model was validated by assessing and comparing the performance of three distinct inputs as follows:

- Spam detection metadata.
- Input 1: Tweet text embedding.
- Input 2: Tweet text embedding + screenname embedding.

Input 3: Tweet text embedding + screenname embedding + metadata (profile and content numerical features), as defined in TABLE III.

TABLE III. SPAM DETECTION METADATA

Feature name	Description
word_count	Number of words in the tweet
word_count_pre	Number of words in the preprocessed tweet
word_count_pre/ word_count	Ratio of words in the preprocessed tweet to the original tweet
char_count	Number of characters in the tweet
char_count_pre	Number of characters in the preprocessed tweet
char_count_pre/ char_count	Ratio of characters in the preprocessed tweet to the original tweet
semantic_similarity	Cosine similarity between tweet text and hashtag keywords embeddings
has_media	The presence of an image or video in a tweet
URLs_count	Number of URL(s) in tweet
mention_count	Number of users mentioned in the tweet
tags_count	Number of hashtags in a tweet
emoji_count	Number of emojis in the tweet
rt_count	How many times the tweet was retweeted
source	The software used to publish the tweet
fav_count	How many times the tweet was favorited
age	Number of days from the account creation date to the tweet publishing date
followers	Number of accounts following the user
friends	Number of accounts followed by the user
followers/friends	Ratio of followers to friends
reputation	Ratio of the number of followers to the total number of followers and friends
tweets_count	Number of tweets published by the user
tweets/day	Ratio of the number of tweets to account age
fav_list_count	Number of tweets favorited by the user
fav/day	Ratio of the number of favorites to account age
default_profile	If the user did not change the default profile bio
default_profile_image	If the user did not change the default profile image

IV. RESULTS AND DISCUSSION

This study focuses on building an efficient model to detect both Arabic spam messages and accounts on Twitter. Furthermore, it seeks to achieve high performance with minimal data retrieval requirements. The classification was conducted via text input, thereby avoiding the need to obtain metadata from Twitter. The features were derived from the text input using advanced transformer-based language models, comparing word-level and sentence-level transformers. The assessment was also conducted on two MTL paradigms: hard and soft parameter sharing. The proposed method was evaluated using three dataset versions: DS_Multi_8, DS_Multi_4, and DS_Binary. The performance measures achieved through these experiments are presented in TABLE IV. TABLE V. and TABLE VI. , and visualized in Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10Fig. 11.

Across the three datasets, it is evident that an increase in tweet classes decreases tweet accuracy while enhancing account detection accuracy. In DS_Multi_8, the utilization of AraBERT embeddings showed superior efficacy in spam detection across both paradigms. Sentence transformer embeddings are more effective for account-level detection than for tweet-level detection. Following the reduction in spam classes from 7 to 3, tweet classification improved by approximately 1%, while account classification declined by roughly 2%. Moreover, AraBERT embeddings consistently provide superior performance. In the binary dataset, spam post detection improved, whereas account detection declined, and SBERT and LaBSE surpassed AraBERT at the account level by 1%.

The performance of hard and soft parameter sharing is nearly identical, exhibiting only minor variances. LaBSE and E5 exhibit superior performance using the hard parameter paradigm compared to the soft parameter sharing approach.

The model input was validated by comparing it with three inputs: Input 1, Input 2, and Input 3. The comparison of these inputs is shown in TABLE VII. . In DS_Binary, the experiment demonstrated consistent performance regardless of the input utilized. This conclusion considers Input 1 the optimal selection for this task, as the additional knowledge does not influence model performance. The performance of the multi-spam class datasets varies according to the chosen input. Input 2 achieved superior performance; however, utilizing Twitter text input (Input 1) resulted in a decrease in model performance of approximately 1%. Conversely, the incorporation of metadata (Input 2) has varied between degrading and maintaining the model's accuracy without alteration, determining that the effectual input is Input 2.

For the error analysis, the confusion matrix illustrates the classification performance of the model. Fig. 14 shows that the model achieved strong performance, particularly for sorcery content, pornographic content, money requests, and illegal medicine advertisements. However, a notable confusion arises between the irrelevant content and non-spam classes, with 39 irrelevant tweets mislabeled as non-spam and 21 non-spam tweets misclassified as irrelevant content. This suggests that the model struggles to distinguish ordinary tweets associated with irrelevant hashtags due to the similarity between the two categories. In addition, several commercial advertisement tweets were misclassified as illegal services (8 samples) or irrelevant content (13 samples), indicating that persuasive or promotional language sometimes overlaps with service-oriented or generic content. Such blurring may occur when advertisements avoid explicit commercial keywords. At the account level, Fig. 15 demonstrates excellent overall performance. The few misclassifications are likely attributable to legitimate accounts exhibiting spam-like behavior and, conversely, spammers attempting to mimic legitimate user activity.

Compared to prior research, the work by Mubarak et al. focused on a relatively narrow scope of spam detection, specifically targeting advertisements in [10] and accounts promoting pornographic content in [14]. Additionally, their first study limited the dataset to tweets targeting specific news accounts, thereby reducing generalizability. While Alharthi et al. [49] extended the classification to three spam categories, the model remained dependent on the presence of URLs, which restricted its applicability. Other studies, such as [51] and [52] , focused on spam detection based on specific spam-related keywords, while [53] and [25] primarily targeted spam associated with trending hashtags. In contrast, our proposed model addresses these limitations by generalizing spam classification across diverse categories without dependence on predefined targets or URL-based heuristics. It was trained using a comprehensive dataset, collected by targeting hashtags, user mentions, and spam-related keywords.

Alharthi et al. [49] introduced two separate models—one for spam tweet detection and another for spam account detection—whereas our approach integrates both tasks into a single and unified model. Notably, their account-based model required a minimum of ten tweets per account to achieve reliable classification, while our model demonstrated high accuracy using only a single tweet. Alhassun et al. [53] proposed a framework that relies on both tweet content and extensive metadata, including premium features that require additional effort to extract. In contrast, our findings indicate that incorporating numerical features did not significantly enhance performance, simplifying the model and reducing dependency on complex metadata.

To the best of our knowledge, our model is the first to simultaneously address tweet-level and account-level spam detection in a single framework, highlighting different spamming categories.

TABLE IV. EXPERIMENTAL RESULTS USING DS_MULTI_8

		Hard Parameter Sharing				Soft Parameter Sharing			
task	**Metric**	**AraBERT**	**E5**	**SBERT**	**LaBSE**	**AraBERT**	**E5**	**SBERT**	**LaBSE**
Tweet	**Accuracy**	96	93	94	95	96	91	94	93
	F1-Score	95	93	93	94	96	89	93	93
	Precision	95	94	94	95	96	91	93	93
	Recall	96	92	93	93	95	87	93	92
Account	**Accuracy**	96	95	95	96	96	95	95	96
	F1-Score	96	95	95	96	96	95	95	96
	Precision	96	95	95	96	96	95	95	96
	Recall	96	95	95	96	96	95	95	96

TABLE V. EXPERIMENTAL RESULTS USING DS_MULTI_4

		Hard Parameter Sharing				Soft Parameter Sharing			
task	**Metric**	**AraBERT**	**E5**	**SBERT**	**LaBSE**	**AraBERT**	**E5**	**SBERT**	**LaBSE**
Tweet	**Accuracy**	97	94	95	96	97	92	95	95
	F1-Score	96	94	95	96	96	91	95	95
	Precision	97	94	95	96	97	93	95	95
	Recall	96	94	95	96	96	89	95	95
Account	**Accuracy**	94	93	93	94	94	94	93	94
	F1-Score	94	93	93	94	94	94	93	94
	Precision	94	93	93	94	94	94	93	94
	Recall	94	93	93	94	94	94	93	94

TABLE VI. EXPERIMENTAL RESULTS USING DS_BINARY

		Hard Parameter Sharing				Soft Parameter Sharing			
task	**Metric**	**AraBERT**	**E5**	**SBERT**	**LaBSE**	**AraBERT**	**E5**	**SBERT**	**LaBSE**
Tweet	**Accuracy**	98	96	96	96	98	95	96	96
	F1-Score	98	96	96	96	98	95	96	96
	Precision	98	96	96	96	98	95	96	96
	Recall	98	96	96	96	98	95	96	96
Account	**Accuracy**	91	90	92	92	90	90	91	92
	F1-Score	91	89	92	92	90	90	91	92
	Precision	91	91	92	92	90	90	91	92
	Recall	91	89	92	92	90	90	91	92

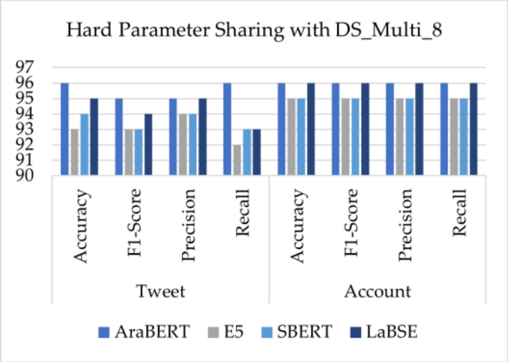

Fig. 6. Performance measures of the hard parameter sharing model with DS_Multi_8.

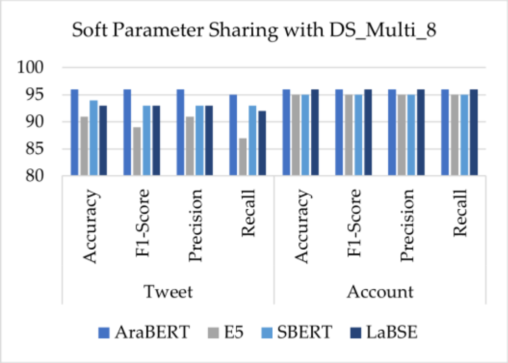

Fig. 7. Performance measures of the soft parameter sharing model with DS_Multi_8.

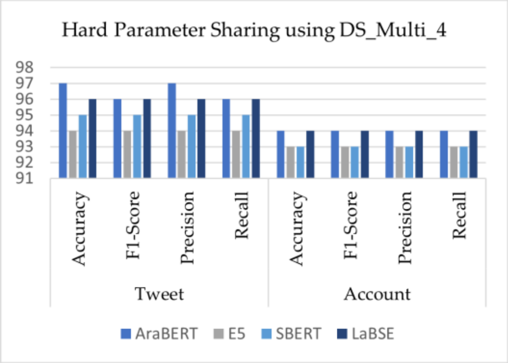

Fig. 8. Performance measures of the hard parameter sharing model with DS_Multi_4.

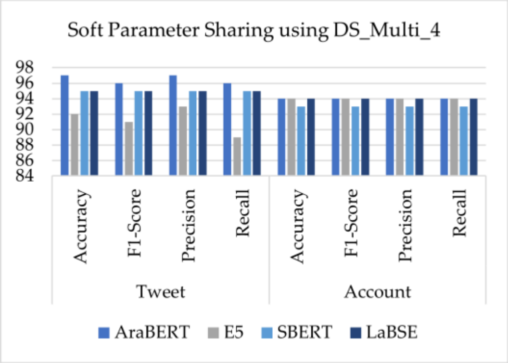

Fig. 9. Performance measures of the soft parameter sharing model with DS_Multi_4.

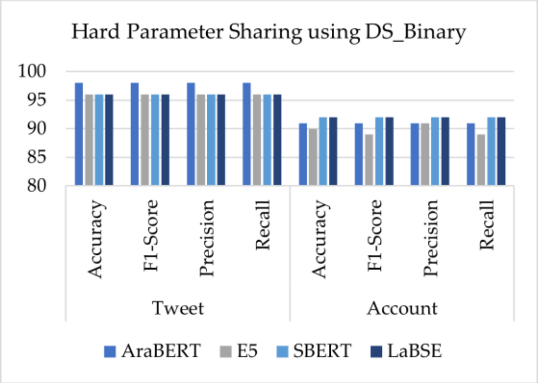

Fig. 10. Performance measures of the hard parameter sharing model with DS_Binary.

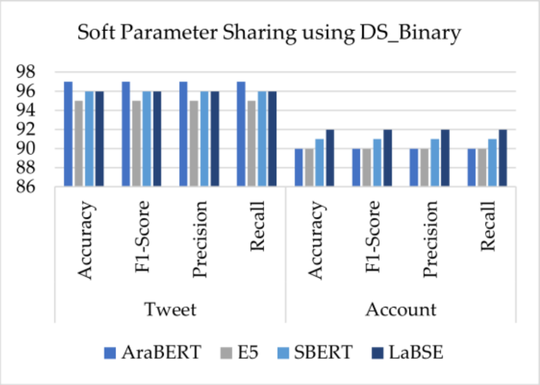

Fig. 11. Performance measures of the soft parameter sharing model with DS_Binary.

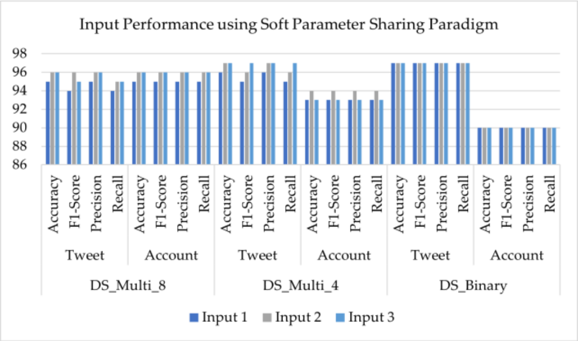

Fig. 12. Input performance using the hard parameter sharing paradigm.

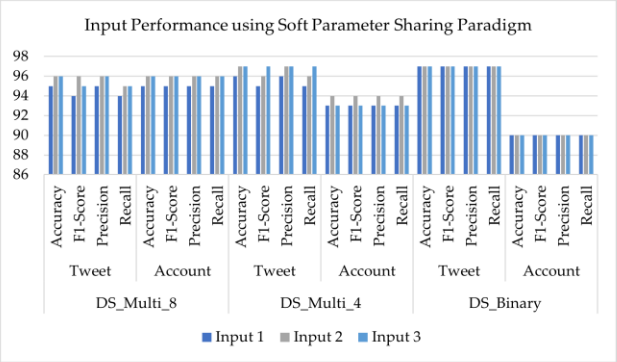

Fig. 13. Input performance using soft parameter sharing paradigm.

TABLE VII. PROPOSED INPUT PERFORMANCE

			Hard Parameter Sharing			**Soft Parameter Sharing**		
DS	**task**	**Metric**	**Input 1**	**Input 2**	**Input 3**	**Input 1**	**Input 2**	**Input 3**
DS_Multi_8	**Tweet**	**Accuracy**	95	96	95	95	96	96
		F1-Score	94	95	95	94	96	95
		Precision	95	95	95	95	96	96
		Recall	93	96	95	94	95	95
	Account	**Accuracy**	95	96	96	95	96	96
		F1-Score	95	96	96	95	96	96
		Precision	95	96	96	95	96	96
		Recall	95	96	96	95	96	96
DS_Multi_4	**Tweet**	**Accuracy**	95	97	96	96	97	97
		F1-Score	95	96	96	95	96	97
		Precision	95	97	96	96	97	97
		Recall	95	96	96	95	96	97
	Account	**Accuracy**	93	94	94	93	94	93
		F1-Score	93	94	94	93	94	93
		Precision	93	94	94	93	94	93
		Recall	93	94	94	93	94	93
DS_Binary	**Tweet**	**Accuracy**	98	98	98	97	98	97
		F1-Score	98	98	98	97	98	97
		Precision	98	98	98	97	98	97
		Recall	98	98	98	97	98	97
	Account	**Accuracy**	91	91	91	90	90	90
		F1-Score	91	91	91	90	90	90
		Precision	91	91	91	90	90	90
		Recall	91	91	91	90	90	90

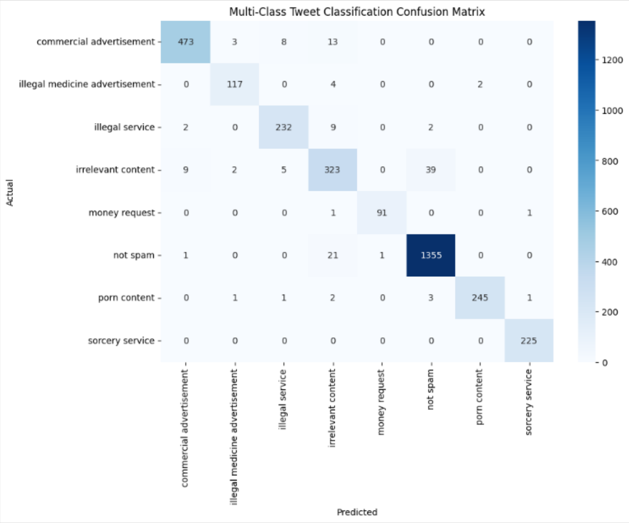

Fig. 14. Confusion matrix of spam tweet classification with DS_Multi_8.

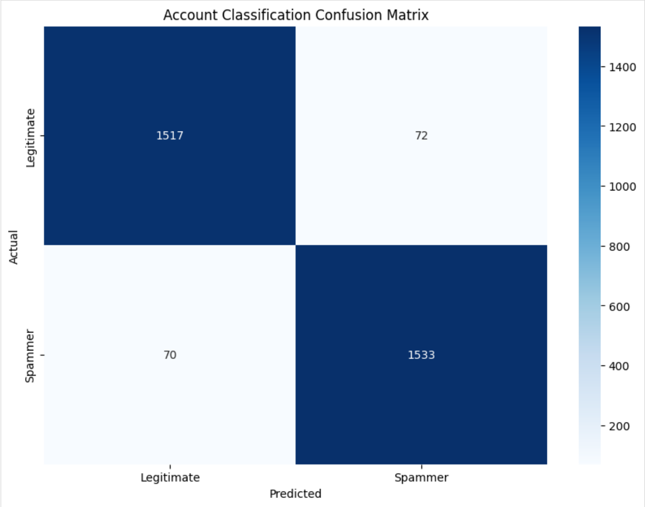

Fig. 15. Confusion matrix of spam account classification with DS_Multi_8.

V. Conclusion

This study introduced AraSpam, a multitask deep learning model designed to detect spam messages and spamming accounts in Arabic Twitter data, addressing limitations in prior models that focused on either content or user-level analysis in isolation. By leveraging transformer-based architectures such as AraBERT and mBERT for robust feature extraction, AraSpam achieved impressive performance, notably attaining a 96% accuracy rate on multi-class spam detection tasks.

The experimental evaluation across diverse datasets (DS_Multi_8, DS_Multi_4, and DS_Binary) highlighted the effectiveness of AraBERT in representing text and capturing semantics from tweet content. Notably, the results demonstrated that reducing the number of spam classes improved tweet-level classification while slightly compromising profile-level detection, reflecting the trade-off between granularity and generalization.

The study also explored different multitask learning paradigms, showing competitive results between hard and soft parameter sharing approaches, with LaBSE and E5 models performing particularly well under hard sharing. Furthermore, input validation experiments confirmed that Input 1, which avoids reliance on Twitter metadata, provides a practical balance between performance and feasibility.

Despite its strong results, AraSpam has room for enhancement. Future work should explore multimodal learning, incorporating image data from tweets to address visual spam content, which remains unprocessed in the current framework. Moreover, the dataset includes tweets with 280 characters maximum length and we are planning to enhance the dataset by including longer tweets.

In summary, AraSpam demonstrates the value of multitask learning and linguistically aware models in tackling the complex challenge of spam detection in Arabic social media.

References

[1] E. Ferrara, "The history of digital spam," Commun. ACM, vol. 62, no. 8, pp. 82–91, 2019, doi: 10.1145/3299768.

[2] L. F. Cranor and B. A. LaMacchia, "Spam!," Commun. ACM, vol. 41, no. 8, pp. 74–83, 1998, doi: 10.1145/280324.280336.

[3] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in Collaboration, electronic messaging, anti-abuse and spam conference (CEAS), 2010, vol. 6, no. 2010, p. 12.

[4] X. H. Center. "Platform manipulation and spam policy." (accessed 14th January, 2025).

[5] A. H. Wang, "Don't follow me: Spam detection in twitter," in 2010 international conference on security and cryptography (SECRYPT), 2010: IEEE, pp. 1-10.

[6] S. Sedhai and A. Sun, "Hspam14: A collection of 14 million tweets for hashtag-oriented spam research," in Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, 2015, pp. 223-232.

[7] D. Alorini and D. B. Rawat, "Automatic spam detection on gulf dialectical Arabic Tweets," in 2019 International Conference on Computing, Networking and Communications (ICNC), 2019: IEEE, pp. 448-452.

[8] H. S. Al-Khalifa, "On the analysis of twitter spam accounts in Saudi Arabia," International Journal of Technology Diffusion (IJTD), vol. 6, no. 1, pp. 46-60, 2015.

[9] N. Jindal and B. Liu, "Opinion spam and analysis," in Proceedings of the 2008 international conference on web search and data mining, 2008, pp. 219-230.

[10] H. Mubarak, A. Abdelali, S. Hassan, and K. Darwish, "Spam detection on arabic twitter," in International Conference on Social Informatics, 2020: Springer, pp. 237-251.

[11] K. S. Adewole, N. B. Anuar, A. Kamsin, K. D. Varathan, and S. A. Razak, "Malicious accounts: dark of the social networks," Journal of Network and Computer Applications, vol. 79, pp. 41-67, 2017.

[12] S. W. Liew, N. F. M. Sani, M. T. Abdullah, R. Yaakob, and M. Y. Sharum, "An effective security alert mechanism for real-time phishing tweet detection on Twitter," Computers & Security, vol. 83, pp. 201-207, 2019.

[13] A. T. Kabakus and R. Kara, "A survey of spam detection methods on twitter," International Journal of Advanced Computer Science and Applications, vol. 8, no. 3, pp. 29-38, 2017.

[14] H. Mubarak, S. Hassan, and A. Abdelali, "Adult content detection on arabic twitter: Analysis and experiments," in Proceedings of the Sixth Arabic Natural Language Processing Workshop, 2021, pp. 136-144.

[15] M. H. Arif, J. Li, M. Iqbal, and K. Liu, "Sentiment analysis and spam detection in short informal text using learning classifier systems," Soft Computing, vol. 22, no. 21, pp. 7281-7291, 2018.

[16] S. B. Abkenar, M. H. Kashani, M. Akbari, and E. Mahdipour, "Twitter spam detection: A systematic review," arXiv preprint arXiv:2011.14754, 2020.

[17] T. Wu, S. Wen, Y. Xiang, and W. Zhou, "Twitter spam detection: Survey of new approaches and comparative study," Computers & Security, vol. 76, pp. 265-284, 2018/07/01/ 2018, doi: https://doi.org/10.1016/j.cose.2017.11.013.

[18] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in Proceedings of the australasian computer science week multiconference, 2017, pp. 1-8.

[19] S. Rao, A. K. Verma, and T. Bhatia, "A Review on Social Spam Detection: Challenges, Open Issues, and Future Directions," Expert Systems with Applications, p. 115742, 2021.

[20] M. Chakraborty, S. Pal, R. Pramanik, and C. R. Chowdary, "Recent developments in social spam detection and combating techniques: A survey," Information Processing & Management, vol. 52, no. 6, pp. 1053-1073, 2016.

[21] F. Persia and D. D'Auria, "A survey of online social networks: challenges and opportunities," in 2017 IEEE International Conference on Information Reuse and Integration (IRI), 2017: IEEE, pp. 614-620.

[22] H. Afzal and K. Mehmood, "Spam filtering of bi-lingual tweets using machine learning," in 2016 18th International Conference on Advanced Communication Technology (ICACT), 2016: IEEE, pp. 710-714.

[23] A. M. Al-Zoubi, J. f. Alqatawna, H. Faris, and M. A. Hassonah, "Spam profiles detection on social networks using computational intelligence methods: The effect of the lingual context," Journal of Information Science, p. 0165551519861599, 2019.

[24] A.-Z. Ala'M, H. Faris, J. f. Alqatawna, and M. A. Hassonah, "Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts," Knowledge-Based Systems, vol. 153, pp. 91-104, 2018.

[25] N. El-Mawass and S. Alaboodi, "Detecting Arabic spammers and content polluters on Twitter," in 2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC), 2016: IEEE, pp. 53-58.

[26] S. Al-Azani and E.-S. M. El-Alfy, "Detection of arabic spam tweets using word embedding and machine learning," in 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2018: IEEE, pp. 1-5.

[27] M. M. Alsulami and A. Y. Al-Aama, "SentiFilter: A Personalized Filtering Model for Arabic Semi-spam Content Based on Sentimental and Behavioral Analysis," Int. J. Adv. Comput. Sci. Appl, vol. 11, 2020.

[28] H. Gupta, M. S. Jamal, S. Madisetty, and M. S. Desarkar, "A framework for real-time spam detection in Twitter," in 2018 10th International Conference on Communication Systems & Networks (COMSNETS), 2018: IEEE, pp. 380-383.

[29] X. Ban, C. Chen, S. Liu, Y. Wang, and J. Zhang, "Deep-learnt features for Twitter spam detection," in 2018 International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec), 2018: IEEE, pp. 208-212.

[30] M. Neha and M. S. Nair, "A novel twitter spam detection technique by integrating inception network with attention based lstm," in 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021: IEEE, pp. 1009-1014.

[31] K. Maithili et al., "An Effective Twitter Spam Detection Model using Multiple Hidden Layers Extreme Learning Machine," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 1s, pp. 01-09, 2024.

[32] A. Ghourabi and M. Alohaly, "Enhancing spam message classification and detection using transformer-based embedding and ensemble learning," Sensors, vol. 23, no. 8, p. 3861, 2023.

[33] M. S. Karakaşlı, M. A. Aydin, S. Yarkan, and A. Boyaci, "Dynamic feature selection for spam detection in Twitter," in International Telecommunications Conference, 2019: Springer, pp. 239-250.

[34] I. Inuwa-Dutse, M. Liptrott, and I. Korkontzelos, "Detection of spam-posting accounts on Twitter," Neurocomputing, vol. 315, pp. 496-511, 2018/11/13/ 2018, doi: https://doi.org/10.1016/j.neucom.2018.07.044.

[35] F. Concone, G. L. Re, M. Morana, and C. Ruocco, "Twitter Spam Account Detection by Effective Labeling," in ITASEC, 2019.

[36] K. S. Adewole, N. B. Anuar, A. Kamsin, and A. K. Sangaiah, "SMSAD: a framework for spam message and spam account detection," Multimedia Tools and Applications, vol. 78, no. 4, pp. 3925-3960, 2019/02/01 2019, doi: 10.1007/s11042-017-5018-x.

[37] Z. Alom, B. Carminati, and E. Ferrari, "A deep learning model for Twitter spam detection," Online Social Networks and Media, vol. 18, p. 100079, 2020.

[38] B. Abu-Salih et al., "An intelligent system for multi-topic social spam detection in microblogging," Journal of Information Science, vol. 50, no. 6, pp. 1471-1498, 2024.

[39] M. Washha, A. Qaroush, and F. Sedes, "Leveraging time for spammers detection on Twitter," presented at the Proceedings of the 8th International Conference on Management of Digital EcoSystems, Biarritz, France, 2016. [Online]. Available: https://doi.org/10.1145/3012071.3012078.

[40] H. Tajalizadeh and R. Boostani, "A novel stream clustering framework for spam detection in Twitter," IEEE Transactions on Computational Social Systems, vol. 6, no. 3, pp. 525-534, 2019.

[41] F. Concone, G. L. Re, M. Morana, and C. Ruocco, "Assisted Labeling for Spam Account Detection on Twitter," in 2019 IEEE International Conference on Smart Computing (SMARTCOMP), 2019: IEEE, pp. 359-366.

[42] K. S. Adewole, T. Han, W. Wu, H. Song, and A. K. Sangaiah, "Twitter spam account detection based on clustering and classification methods," The Journal of Supercomputing, vol. 76, no. 7, pp. 4802-4837, 2020.

[43] R. Alharthi, A. Alhothali, and K. Moria, "Detecting and characterizing arab spammers campaigns in Twitter," Procedia Computer Science, vol. 163, pp. 248-256, 2019.

[44] P.-C. Chen, H.-M. Lee, H.-R. Tyan, J.-S. Wu, and T.-E. Wei, "Detecting spam on Twitter via message-passing based on retweet-relation," in International Conference on Technologies and Applications of Artificial Intelligence, 2014: Springer, pp. 56-65.

[45] S. Gupta, A. Khattar, A. Gogia, P. Kumaraguru, and T. Chakraborty, "Collective classification of spam campaigners on Twitter: A hierarchical meta-path based approach," in Proceedings of the 2018 World Wide Web Conference, 2018, pp. 529-538.

[46] D. Antonakaki, I. Polakis, E. Athanasopoulos, S. Ioannidis, and P. Fragopoulou, "Exploiting abused trending topics to identify spam campaigns in Twitter," Social Network Analysis and Mining, vol. 6, no. 1, p. 48, 2016/07/13 2016, doi: 10.1007/s13278-016-0354-9.

[47] A. Balfagih, V. Keselj, and S. Taylor, "N-gram and word2vec feature engineering approaches for spam recognition on some influential twitter topics in saudi arabia," in Proceedings of the 6th International Conference on Information System and Data Mining, 2022, pp. 101-107.

[48] A. M. Alkadri, A. Elkorany, and C. Ahmed, "Enhancing detection of Arabic social spam using data augmentation and machine learning," Applied Sciences, vol. 12, no. 22, p. 11388, 2022.

[49] R. Alharthi, A. Alhothali, and K. Moria, "A real-time deep-learning approach for filtering Arabic low-quality content and accounts on Twitter," Information Systems, vol. 99, p. 101740, 2021.

[50] N. Imam and V. Vassilakis, "Detecting spam images with embedded arabic text in twitter," in 2019 international conference on document analysis and recognition workshops (ICDARW), 2019, vol. 6: IEEE, pp. 1-6.

[51] A. R. Alharbi and A. Aljaedi, "Predicting rogue content and arabic spammers on twitter," Future Internet, vol. 11, no. 11, p. 229, 2019.

[52] S. Kaddoura, S. A. Alex, M. Itani, S. Henno, A. AlNashash, and D. J. Hemanth, "Arabic spam tweets classification using deep learning," Neural Computing and Applications, vol. 35, no. 23, pp. 17233-17246, 2023.

[53] A. S. Alhassun and M. A. Rassam, "A combined text-based and metadata-based deep-learning framework for the detection of spam accounts on the social media platform twitter," Processes, vol. 10, no. 3, p. 439, 2022.

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

[55] B. Muller, A. Anastasopoulos, B. Sagot, and D. Seddah, "When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models," arXiv preprint arXiv:2010.12858, 2020.

[56] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," arXiv preprint arXiv:2003.00104, 2020.

[57] N. Reimers, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," arXiv preprint arXiv:1908.10084, 2019.

[58] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic BERT sentence embedding," arXiv preprint arXiv:2007.01852, 2020.

[59] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Multilingual e5 text embeddings: A technical report," arXiv preprint arXiv:2402.05672, 2024.

[60] W. Antoun and F. Baly. "AraBERTv0.2-Twitter." (accessed 1st March, 2025).

[61] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), 2008: Ieee, pp. 1322-1328.

[62] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436-444, 2015.

[63] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," Journal of machine learning research, vol. 10, no. 1, 2009.

[64] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295-2329, 2017.

[65] R. Caruana, "Multitask learning," Machine learning, vol. 28, no. 1, pp. 41-75, 1997.

[66] Y. Zhang and Q. Yang, "A survey on multi-task learning," IEEE Transactions on Knowledge and Data Engineering, 2021.

[67] X. Yang, S. Kim, and E. Xing, "Heterogeneous multitask learning with joint sparsity constraints," Advances in neural information processing systems, vol. 22, pp. 2151-2159, 2009.

[68] Y. Zhang and Q. Yang, "An overview of multi-task learning," National Science Review, vol. 5, no. 1, pp. 30-43, 2018.

[69] S. Ruder, "An overview of multi-task learning in deep neural networks," arXiv preprint arXiv:1706.05098, 2017.