Arabic Sentiment Analysis Using Optuna Hyperparameter Optimization and Metaheuristics Feature Selection to Improve Performance of LightGBM

Mostafa Medhat Nazier[1], Mamdouh M. Gomaa[2], Mohamed M. Abdallah[3], Awny Sayed[4]
Computer Science Department-Faculty of Science, Minia University, Minia 61519, Egypt[1, 2, 3]
Information Technology Department-Faculty of Computing and Information Technology,
King Abdulaziz University, Jeddah 21589, Saudi Arabia[4]

***Abstract*—Sentiment Analysis (SA) effectively examines big data, such as customer reviews, market research, social media posts, online discussions, and customer feedback evaluation. Arabic Language is a complex and rich language. The main reason for the need to enhance Arabic resources is the existence of numerous dialects alongside the standard version (MSA). This study investigates the impact of stemming and lemmatization methods on Arabic sentiment analysis (ASA) using Machine Learning techniques, specifically the LightGBM classifier. It also employs metaheuristic feature selection algorithms like particle swarm optimization, dragonfly optimization, grey wolf optimization, harris hawks optimizer, and a genetic optimization algorithm to identify the most relevant features to improve LightGBM's model performance. It also employs the Optuna hyperparameter optimization framework to determine the optimal set of hyperparameter values to enhance LightGBM model performance. It also underscores the importance of preprocessing strategies in ASA and highlights the effectiveness of metaheuristic approaches and Optuna hyperparameter optimization in improving LightGBM model performance in ASA. It also applies different stemming and lemmatization methods, Metaheuristic Feature Selection algorithms, and the Optuna hyperparameter optimization on eleven datasets with different Arabic dialects. The findings indicate that metaheuristics feature selection with the LightGBM classifier, using suitable stemming and lemmatization or combining them, enhances LightGBM's accuracy by between 0 and 8%. Still, Optuna hyperparameter optimization with the LightGBM classifier, using suitable stemming and lemmatization or combining them, depending on data characteristics, improves LightGBM's accuracy by between 2 and 11%. It achieves superior results than metaheuristics feature selection in more than 90% of cases. This study is of significant importance in the field of ASA, providing valuable insights and directions for future research.**

Keywords*—*Arabic Sentiment Analysis (ASA); big data; Light Gradient Boosting Machine (LightGBM); Optuna hyperparameter optimization; metaheuristics feature selection; machine learning

I. Introduction

Sentiment analysis (SA), also known as opinion mining, is a technique within natural language processing (NLP) that involves several steps: data collection, preprocessing, feature extraction, and sentiment classification. It has recently become popular as an effective tool for examining big data, such as social media posts, customer reviews, market research, online discussions, and social media monitoring [1]. The popularity of SA has grown significantly among marketers and consumers. It enables them to gain insights into products and analyze market behavior. This method, further enhanced by machine learning (ML), data mining (DM), and deep learning (DL) algorithms, objectively assesses whether a text expresses positive or negative emotions or conveys sentiments about a specific issue, instilling confidence in its impartiality [2]. The internet offers valuable insights into Arabic Sentiment Analysis (ASA). However, analyzing Arabic content poses challenges due to the language's complexities, morphological features, inadequate resources, and the absence of suitable corpora [3]. Although plenty of resources exist to understand English social media content, Arabic resources still need improvement. This gap arises primarily from the variety of Arabic dialects in addition to Modern Standard Arabic (MSA). These dialects are linguistically exciting and widely used among Arabic speakers in everyday conversations and on social media. There is an urgent need to develop specialized AI models for language-specific applications in Arabic [4]. The main challenges of SA in Arabic dialects include morphological analysis [5], the scarcity of datasets [4], the complexity of dialects, and dialects scripted in Latin [5].

ASA presents significant challenges due to the linguistic complexities of the Arabic language, including its dialectical variations, morphological richness, and the lack of extensive labeled datasets. Despite the advancement of machine learning techniques, existing sentiment analysis models often struggle with the diversity of Arabic dialects and the need for effective text preprocessing methods. Additionally, optimizing machine learning models for these challenges requires not only efficient algorithms but also hyperparameter optimization and feature selection to enhance performance. Light Gradient Boosting Machine (LightGBM) has been shown to be a powerful model for classification tasks, but its performance in ASA, particularly with large and diverse Arabic datasets, has not been fully explored. This research aims to bridge this gap by enhancing LightGBM's performance using robust preprocessing techniques (ISRI stemmer and Qalsadi lemmatizer) and advanced optimization methods (metaheuristic feature selection and Optuna hyperparameter tuning).

Feature Selection (FS), also called variable subset selection, is a crucial preprocessing step in ML. It helps reduce computational costs and improve classification accuracy [6] [7]. FS achieves this by discarding noisy, redundant, or irrelevant features, focusing instead on a smaller subset that is sufficient to describe the concept of interest. This process improves the predictor's performance, simplifies data processing, and reduces computational demands. Since the 1960s, FS research has emphasized developing efficient methods to handle high-dimensional datasets, which often include irrelevant or obsolete features [8]. Meta-heuristic techniques, particularly swarm-based optimization algorithms like Grey Wolf Optimization (GWO), Dragonfly Optimization (DFO), Particle Swarm Optimization (PSO), Harris Hawks Optimization (HHO), and Genetic Optimization (GO), have emerged as practical solutions for FS. These methods strike a balance between computational efficiency and solution quality, making them suitable for real-world applications where identifying the optimal feature subset is essential for accurate and cost-effective classification.

Hyperparameter optimization and tuning are critical steps in machine learning to enhance model performance by selecting the best combination of hyperparameters, which are parameters not learned from the data but set prior to training [9]. Effective tuning ensures improved model accuracy, stability, and generalization. Methods such as random search, grid search, and advanced algorithms like genetic algorithms, Bayesian optimization, and hyperband are widely used to efficiently explore the hyperparameter space. Optuna framework further streamline this process. Properly tuned hyperparameters can significantly impact both computational efficiency and predictive performance [10].

The motivation behind this study stems from the growing need for effective sentiment analysis tools for Arabic text, particularly as Arabic is a widely spoken language with various dialects. Traditional approaches to ASA often fail to account for the subtleties of the language, including regional variations and complex word forms. Leveraging LightGBM's strengths with tailored preprocessing techniques, such as the ISRI stemmer and Qalsadi lemmatizer, holds the potential to significantly improve the model's accuracy. Furthermore, applying metaheuristic feature selection algorithms and Optuna for hyperparameter optimization offers a promising way to enhance model performance by selecting the most informative features and fine-tuning the model's parameters. The use of large and diverse datasets is crucial to better understanding the impact of these techniques on ASA across different Arabic dialects. The combination of these methodologies could provide a substantial advancement in the field of ASA, making it more adaptable and accurate for real-world applications.

The paper's contributions include using LightGBM classification algorithm in ASA and improving its performance using ISRI stemmer and Qalsadi lemmatizer with metaheuristic feature selection algorithms and Optuna hyperparameter optimization. A key aspect of our research is using large Arabic datasets from different Arabic dialects in ASA. This diversity in the datasets could significantly enhance LightGBM's performance in ASA. The proposed approach involves using ISRI stemmer alone and Qalsadi lemmatizer alone for data preprocessing and combining them, followed by implementing metaheuristic feature selection algorithms and Optuna. We also compare Optuna hyperparameter optimization with metaheuristic feature selection algorithms to see the impact of improving LightGBM's performance in ASA. It also shows their effects on enhancing ASA.

The remainder of the paper is organized as follows: The "Related Work" in Section II examines previous research on ASA, LightGBM, Optuna hyperparameter, and metaheuristics feature selection. The "Proposed Methodology" in Section III provides detailed information about the proposed approach. In the "Results and Analysis" in Section IV, we present and compare the experimental findings with those of other methods. The "Discussion" in Section V interprets the results and contextualizes their significance, bridging the gap between the findings and the broader implications of the study. Finally, the "Conclusion" in Section VI summarizes the main points of the research.

II. Related Work

This section provides an overview of the existing research on ASA, focusing on key areas such as preprocessing techniques, ML and DL algorithms, FS, and hyperparameter optimization. Several studies have contributed to advancing ASA methodologies by exploring diverse approaches for improving sentiment classification accuracy. Research in ASA has addressed challenges related to the linguistic complexity of Arabic, including its morphological richness and dialectical diversity. In particular, studies have investigated various preprocessing techniques like word embedding, stemming, and lemmatization to enhance text representation. Additionally, the integration of advanced classification algorithms such as LightGBM, along with metaheuristic feature selection and hyperparameter optimization methods like Optuna, have emerged as crucial elements in improving the performance of ASA models. This review synthesizes the most prominent contributions in these areas, providing a comprehensive understanding of the current state-of-the-art in ASA and highlighting opportunities for further development. In study [3], this paper compared several ASA models and discussed the DL algorithms employed in ASA within the domain of e-marketing. The paper's contribution includes improving ASA using preprocessing techniques like word embedding. In a study referenced as study [11], the semantic orientation method was devised to determine the overall polarity of Arabic subjective texts. The technique involved using a specialized domain ontology and an established sentiment lexicon. This technique was evaluated using an Arabic dataset from the hospitality industry to construct the domain ontology. In study [12], this paper employs the Levenshtein distance algorithm for data preprocessing and implementing various classification models and introduces a novel method for conducting ASA using the mobile application comments dataset. This study thoroughly [13], reviews textual content analysis in the ASA domain, examining 133 ASA papers published from 2002 to 2020. It explores common themes, methodologies, technologies, and algorithms used in these studies. This paper's key finding indicate various approaches, such as ML, lexicon-based, and hybrid methods, with algorithms like SVM, Naive Bayes (NB), and hybrid methods proving the most effective. The research presented in study [14] introduces an explainable sentiment classification framework tailored for Arabic. A noise layer is incorporated into various DL models, such as BiLSTM and CNN-BiLSTM, to mitigate overfitting. In study [15], the authors introduce LightGBM, a new GBDT algorithm featuring two innovative techniques: Gradient-based one-sided sampling (GOSS) and Exclusive Feature Bundling (EFB). These techniques are designed to handle large data instances and features, offering practical benefits concerning memory efficiency and processing speed. The study in [16] presents an innovative hybrid system for detecting fake news, which integrates a BERT-based model with LightGBM. The performance of this approach is assessed against four classification methods that utilize different word embedding techniques across three actual fake news datasets. In study [17], SA is used to determine sentiments in text. LightGBM is used for efficiency and scalability, but long and short-term memory is preferred to understand the deep context of the text. The LSTM model was trained using the Adam optimizer, and Text Blob was used to train the LightGBM. Short-term memory (92%) scores over a LightGBM (89%) accuracy. The F1 score for Long short-term memory (93%) and LightGBM (92%) is about comparable. In study [18], this paper focuses on calculating emotional scores for product features through comparative sentences and developing a clustering method to analyze the hierarchical relationships among brands. It utilizes an improved computing model that leverages a sentiment dictionary to generate weighted sentiment scores, enhancing the accuracy of an unsupervised algorithm. These scores are then organized into a design structure matrix, facilitating the clustering of brands with similar products. The research presented in study [19] developed four hybrid machine learning techniques for multi-class-based comparative SA using three datasets from diverse domains. The results revealed that the Multilayer Perceptron + Random Forest (MLP + RF) hybrid ML technique, employing a multilayer perceptron as the base estimator, achieved an F1-score of 93.0% and an average accuracy of 93.0%. The research presented in study [20] focused on the Optuna hyperparameter optimization framework in conjunction with the LightGBM algorithm. A 10-fold cross-validated model using Optuna and LightGBM was trained on the FHS dataset. The resulting model achieved an accuracy of 0.930, a sensitivity of 0.897, a specificity of 0.963, an F1 score of 0.929, a precision of 0.963, an area under the receiver operating characteristic curve (AUC-ROC) of 0.978, and a Matthews correlation coefficient (MCC) of 0.861. The study in [21] utilized an ensemble model that combined CNN and LSTM to predict the sentiment polarity of Arabic tweets using the ASTD dataset. The model achieved an F1-score of 64.46% and an accuracy of 65.05% on this dataset. In study [22], researchers explored various DL models, including LSTM and CNN, for ASA. They trained neural language models using two techniques based on word2vec: skip-gram and Continuous Bag of Words (CBOW). The experiments demonstrated that LSTM outperformed CNN in terms of performance. The research presented in study [23] developed a SA model by enhancing the ML approach (using complement Naive Bayes) with features derived from both an Arabic sentiment lexicon and the text itself. In study [24], an attempt was made to address ASA using a DL model. The LABR dataset in this study comprises book reviews.

In conclusion, the literature on ASA highlights the significant progress made in improving sentiment classification through advanced preprocessing techniques, effective feature selection, and optimized machine learning models. LightGBM has been demonstrated as a highly efficient algorithm for ASA, particularly when enhanced by hyperparameter optimization via Optuna and metaheuristic feature selection methods. Studies have shown that preprocessing methods such as word embedding, stemming, and lemmatization can effectively address the challenges posed by the Arabic language's unique structure. Hybrid models and deep learning approaches have demonstrated potential in improving the contextual understanding of sentiment in Arabic texts. However, challenges such as dialectal variation and the limited availability of large, labeled datasets persist. This indicates that future research should focus on addressing these issues while also exploring innovative ways to integrate different algorithms and preprocessing techniques. Ultimately, the continued development and optimization of ASA models will contribute to more accurate and efficient SA tools for Arabic-language applications.

III. Proposed Methodology

This section outlines the proposed methodology for conducting ASA using various preprocessing techniques, stemming, lemmatization, or a combination of them, tokenization, feature extraction, LightGBM as a classifier, metaheuristic feature selection methods, and Optuna hyperparameter optimization strategies to enhance the performance of the LightGBM classifier, as shown in Fig. 1.

A. Preprocessing Data

Preprocessing is a vital phase in ASA, as it prepares the input data for effective SA. This step significantly enhances the quality of SA [25]. In this research, a meticulous data cleansing process was conducted to prepare the datasets. This involved a series of steps, including the removal of non-Arabic words, numbers, symbols, Arabic and English stop words, duplicate characters, Arabic Tashkeel and Tanween, HTML tags, links, and the replacement of characters such as Hamza, Ha, and Ta Marbuta with their simplified equivalents. Tokenization includes breaking down text into smaller units, such as words or sub-words, with high precision. Stemming aims to reduce words to their base forms, known as stems or roots. At the same time, Lemmatization seeks to associate all word variations with their canonical form, called a lemma (the form found in dictionaries) [26]. Stemming and Lemmatization are essential for improving the consistency and accuracy of the SA process. This paper applies ISRI Light stemmer from NLTK to various datasets, successfully eliminating inflections and affixes to expose the base form. It uses a Qalsadi Arabic lemmatizer on data sets [27] [28], and its influence is compared with ISRI stemmers in improving LightGBM performance. It is combined with ISRI stemmer and compared with Qalsadi and ISRI alone. Table I shows comparison between the output of ISRI Stemmer, Qalsadi lemmatizer and ISRI-Qalsadi.

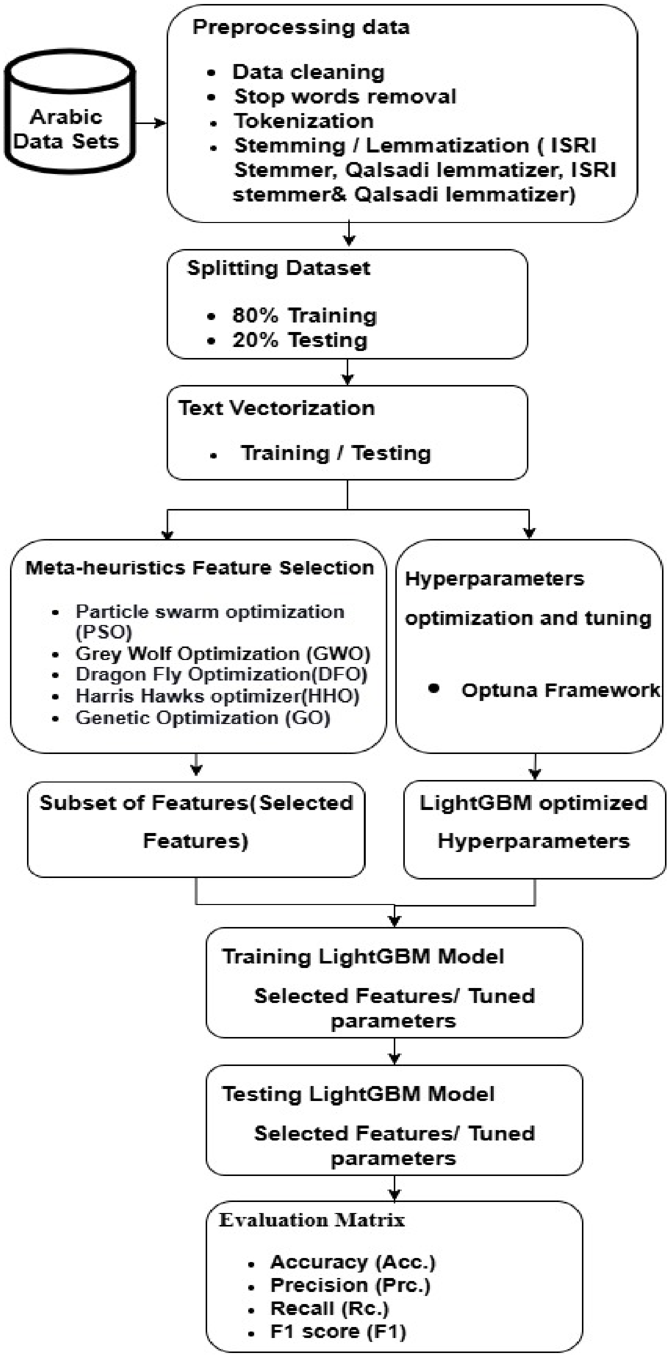

Fig. 1. Proposed methodology.

TABLE I. COMPARISON BETWEEN THE OUTPUT IF ISRI STEMMER, QALSADI LEMMATIZER AND ISRI-QALSADI

Word	ISRI Stemmer	Qalsadi lemmatizer	ISRI-Qalsadi
وبسواعدهما	وبسواعده	سواعد	سواعد
أوصيك	اوص	وصي	اوص
بالدقة	دقة	دقة	دقة
يعجبني	عجب	أعجب	عجب
النرجسيه	رجس	النرجسيه	رجس
الكتاب	كتب	كتاب	تب

B. Spliting Dataset

In this study, the dataset was divided into 80% training and 20% testing sets using the train_test_split method from Scikit-Learn. By using this method, it automatically shuffles the dataset. By shuffling the data, it will be distributed equally in the model, so it will be more accurate for predictions.

C. Text Vectorization

Text vectorization is the process of converting text into numerical representations, enabling their representation in a format suitable for ML models as a feature selection process [29]. In this research, the tokenizer class from Tensor Flow/Keras is used to build the vocabulary based on a list of input texts. It analyzes the texts, extracts unique words, assigns an integer index to each word in the vocabulary and convert it into sequences of indices, assign numerical indices to tokens, and pad sequences (usually zeros) to sequences shorter than the specified length to ensure that all sequences have the same length, which is necessary for processing by ML models [30].

Arabic texts = ["أنا أحب تعلم الآلة", "تحليل النصوص هو جزء من الذكاء الاصطناعي", "نحن نستخدم خوارزميات متعددة لتحليل البيانات"]

The result of text vectorization as below:

Word Index (Vocabulary): {'تحليل': 1, 'نحن': 2, 'نستخدم': 3, 'خوارزميات': 4, 'متعددة': 5, 'لتحليل': 6, 'البيانات': 7, 'أنا': 8, 'أحب': 9, 'تعلم': 10, 'الآلة': 11, 'النصوص': 12, 'هو': 13, 'جزء': 14, 'من': 15, 'الذكاء': 16, 'الاصطناعي': 17}.

Sequences: [[8, 9, 10, 11], [1, 12, 13, 14, 15, 16, 17], [2, 3, 4, 5, 6, 7]].

Padded Sequences: [[8 9 10 11 0 0] [1 12 13 14 15 16 17] [2 3 4 5 6 7 0]].

D. Arabic Text Classification using LightGBM Algorithm

LightGBM, developed by Microsoft, is an advanced gradient-boosting framework known for its faster training times and higher accuracy than other traditional gradient-boosting algorithms [29]. Its efficient memory usage allows it to handle large datasets with minimal resource requirements, resulting in improved performance and cost savings. LightGBM uses histogram-based learning and leaf-wise tree growth to enhance prediction accuracy [30]. It supports distributed GPU learning and parallel training on multi-core CPUs, making it suitable for big data applications. Additionally, its GOSS technique prioritizes critical data points during tree construction, reducing training time and memory usage. LightGBM is applicable for classification, regression, and ranking tasks [30].

The LightGBM algorithm can be represented mathematically as follows: Let X be the training dataset consisting of N examples and M features, and let Y represent the corresponding target values. $f(x_i)$ is defined as a function that maps the input features to the target values. The objective of LightGBM is to minimize the loss function $L(f)$, which measures the difference between the predicted values and the actual target values in relation to the function f as in Eq. (1).

$$L(f) = \sum[y_i - f(x_i)]^2 + \Omega(f) \tag{1}$$

An important regularization term, denoted as Ω(f), enhances the robustness of LightGBM by controlling the complexity of the learned function and preventing overfitting. This term strikes a balance between effectively fitting the training data and generalizing to new data, ensuring a reliable and robust solution. LightGBM addresses this optimization problem by iteratively adding decision trees to the ensemble. At each iteration t, LightGBM constructs a decision tree h_t (x) that minimizes the loss function over a subset of the training examples St:

$$h_t(x) = argmin_h \sum [y_i - f_{t-1}(x_i) - h(x_i)]^2 + \Omega(h) \quad (2)$$

In LightGBM, the ensemble of decision trees from previous iterations, denoted as $f_{t-1}(x_i)$, is essential for the model's performance. Each new tree is trained to address the errors of the prior trees, enhancing predictions iteratively. This ongoing learning process ensures the model adapts and improves over time, reinforcing its reliability. LightGBM employs gradient boosting to optimize the loss function by sequentially adding decision trees. At each iteration (t), it calculates the negative gradient of the loss function based on the predictions from the existing ensemble, as expressed in Eq. (3).

$$g_i = -\partial L(f_{t-1}(x_i)) / \partial f_{t-1}(x_i) \quad (3)$$

LightGBM utilizes the GOSS technique to enhance the training process by selecting a subset of examples. It prioritizes samples with large gradients to ensure their significance while under-sampling those with small gradients to lower computational costs and reduce the risk of overfitting. The algorithm employs a variant of the Gradient-based Decision Tree (GBDT), which constructs decision trees in a leaf-wise manner. In each split, it selects features that maximize loss reduction and prunes the tree based on a minimum gain threshold. This iterative method of adding trees continues until a stopping criterion is met, such as reaching a maximum number of trees or observing minimal improvement in validation error [16]. After training, LightGBM makes predictions by calculating the weighted average of the outputs from the individual trees as expressed in Eq. (4).

$$f(x) = \sum_{t=1}^{T} w_t h_t(x) \quad (4)$$

Where T is the number of trees in the ensemble, $\boldsymbol{w_t}$ is the weight of the t-th tree, and $\boldsymbol{h_t(x)}$ is the prediction of the t-th tree. The LightGBM determines their contribution to reducing the loss function as the weight.

E. *Metaheuristics Feature Selection*

FS is a vital step in ML that helps identify relevant variables related to target outcomes, improving model performance and control. Its key goals include enhancing generalization to reduce overfitting, eliminating redundant features for better inference, and enabling more efficient training with fewer features, shortening training times. Simpler models with fewer features are more easily interpreted [6] [7]. Metaheuristic algorithms, such as PSO, GWO, DFO, HHO, and GO, are practical tools for feature selection due to their reliability and efficiency [31]. However, they may not always guarantee global optimality. Among these, PSO is notable for its simplicity and efficiency in searching for optimal solutions without relying on gradients, making it a straightforward optimization tool with minimal hyperparameters. Inspired by natural behaviors like the collective movement of birds or fish, PSO effectively explores complex solution spaces to find optimal outcomes across various fields [31]. GWO is a recently developed evolutionary algorithm inspired by the social behavior of grey wolves, emphasizing the importance of pack dynamics in achieving reproductive success. In this model, a dominant male and female wolf hold higher ranks and guide the other pack members [32]. DFO is a new swarm intelligence algorithm inspired by the swarming behavior of dragonflies. It mimics five key principles: separation, cohesion, attraction, alignment, and distraction, which help dragonflies avoid collisions, maintain speed, connect with neighbors, seek food, and evade threats. DFO incorporates these behaviors into an optimization technique with two main phases: exploration and exploitation. These phases simulate the social interactions of dragonflies during navigation, food searching, and enemy avoidance in dynamic and static environments [33]. HHO enhances the effectiveness of wrapper-based FS techniques. As a fast and efficient swarm-based optimizer, HHO utilizes straightforward yet powerful exploratory and exploitative mechanisms, including Lévy flight and greedy selection. Additionally, it features a dynamic structure specifically designed for continuous problems. Its efficiency makes HHO a promising tool for a variety of optimization tasks, although it was originally developed for continuous search spaces [34]. GO is a highly effective computational method that is valuable in complex, poorly defined, or high-dimensional search spaces. Its primary goal in feature selection is to reduce the number of features by eliminating redundant and irrelevant ones while maintaining or improving classification accuracy. Various search algorithms have been employed for FS tasks [35].

F. *Hyperparameter Optimization and Tuning*

Hyperparameters play a crucial role in a model's functionality, performance, and structure, making their optimization essential for data scientists [9] [10]. The effectiveness of ML models, such as LightGBM, depends on selecting appropriate hyperparameter values, including learning rate, maximum depth, number of trees, and regularization parameters [36]. A systematic approach to hyperparameter tuning helps balance model complexity and generalization, improving accuracy and training speed. Optuna is noted as an advanced optimization framework that utilizes Bayesian techniques for more effective exploration of parameter spaces, allowing for fewer trials while managing experiments autonomously [10]. This capability enables Optuna to identify optimal hyperparameters that enhance model performance metrics like accuracy, precision, and recall, making hyperparameter optimization vital for maximizing a model's potential and achieving better results [36].

Algorithm 1: A simplified pseudocode for proposed methodology

```
# Step 1: Data Preprocessing
Input: Arabic Dataset
Output: Cleaned and tokenized dataset
BEGIN
    Perform Data Cleaning
        - Remove unnecessary data (e.g., duplicates, special characters)
    Remove Stop Words
    Perform Tokenization
    Apply Stemming/Lemmatization
        - Use ISRI Stemmer
        - Use Qalsadi Lemmatizer
```

```
    - Use ISRI Stemmer & Qalsadi Lemmatizer
END

# Step 2: Split Dataset
Input: Preprocessed dataset
Output: Training set (80%) and Testing set (20%)
BEGIN
  Split dataset into 80% Training and 20% Testing
END

# Step 3: Text Vectorization
Input: Training and Testing sets
Output: Vectorized text data
BEGIN
  Vectorize Text Data for Training and Testing
END

# Step 4: Feature Selection using Meta-heuristics
Input: Vectorized training data
Output: Subset of selected features
BEGIN
  Initialize meta-heuristic optimization algorithms:
    - Particle Swarm Optimization (PSO)
    - Grey Wolf Optimization (GWO)
    - Dragon Fly Optimization (DFO)
    - Harris Hawks Optimization (HHO)
    - Genetic Optimization (GO)
  Perform FS
    - Identify and retain the most relevant features
  Output Selected Features
END

# Step 5: Hyperparameter Optimization and Tuning
Input: Vectorized training data and feature subset
Output: Optimized hyperparameters
BEGIN
  Use Optuna Framework for Hyperparameter Tuning
  Optimize LightGBM's parameters
END

# Step 6: Train LightGBM's Model
Input: Selected features, optimized hyperparameters
Output: Trained LightGBM's model
BEGIN
  Train LightGBM's model using:
    - Selected Features
    - Optimized Hyperparameters
END

# Step 7: Test LightGBM's Model
Input: Testing data, trained model
Output: Evaluation metrics
BEGIN
  Test LightGBM's model
  Evaluate Performance using:
    - Accuracy (Acc.)
    - Precision (Prc.)
    - Recall (Rc.)
    - F1 Score (F1)
END
```

IV. Experiments and Results

This section presents and discusses datasets used in experiments, working environment and experiment setting and classification results and performance evaluation.

A. Datasets Description

The experiments and comparison results use eleven datasets in Table II from GitHub and Kaggle. "qrci " was downloaded from [37]. "Ar_reviews_100k" is a much larger dataset with 100,000 rows and 99999 tweets/reviews. It combines reviews from hotels, books, movies, products, and airlines and was downloaded from [38]. "ARABIC Dataset" was downloaded from [39]. It contains 58751 Arabic tweets. It has three classes (natural, negative, and positive). "ARABIC Dataset_2cat" is "ARABIC Dataset" after removing natural tweets. "mpqa-ar" is an Arabic opinion corpus containing articles from many news sources annotated for opinions downloaded from [37]. LABR is a large ASA dataset. It consists of over 63,000 book reviews [40]. After balancing it, it has 16448 reviews. It was downloaded as a balanced dataset from [37]. "Astd-artwitter" is a combined dataset between ASTD and Artwitter data sets downloaded from [37]. ASTD was downloaded from [37] with 1590 tweets. ASA_SS2030 is a dataset related to social events in the Arabic Saudi Dialect associated with Saudi Arabia's 2030 vision and downloaded from [41]. AJGT introduces an Arabic Jordanian General Tweets (AJGT) Corpus in MSA or Jordanian dialect [42]. "Company Reviews" were collected for SA to produce a score for companies [43]. It has 40K+ reviews in Arabic for SA. It has reviews for Ezz Steel, Talbat, Elsewedy, Hilton, Nestle, Raya, SWVL, Telecom Egypt, TMG, Venus, Domty, and Capiter companies. Table II shows Data Sets Information.

TABLE II. Data Sets Information

Date Set	No Rows	No. Tweets/Reviews	No Categories	No. Positives	No. Negatives	No. Neutral	No. Features
qrci	755	754	2	377	377	0	10
ar_reviews_100k	100000	99999	3	33333	33333	33333	41
ARABIC Dataset_2cat	56498	56497	2	29460	27037	0	10
ARABIC Dataset	58752	58751	3	29460	27037	2254	10
mpqa-ar	9997	9996	2	5399(subjectives)	0	4597	16
LABR-book-reviews	16449	16448	2	8224	8224	0	42
astd-artwitter	3543	3542	2	1771	1771	0	10
ASTD	1590	1591	2	777	812	0	14
ASA_SS2030	4253	4252	2	2436	1816	0	21
AJGT	1801	1800	2	900	900	0	8
Company Reviews	40046	40045	3	23921	14200	1925	8

As shown in Table II, the datasets vary widely in size, ranging from a few thousand to over 100,000 rows. The number of categories also varies, with most datasets having two categories but some having three. The distribution of positive, negative, and neutral examples is only sometimes balanced, especially in larger datasets. The number of features also varies across the datasets, with some having as few as eight features and others having as many as 42.

B. Working Environment and Experimental Setting

The experiments have been done in google Colab using several python libraries such as pandas, numpy, NLTK, Qalsadi, TensorFlow, Scikit-learn, pandas, LightGBM, PSO, GWO, DFO, HHO, GO, zoofs and Optuna framework. Colab is a hosted Jupyter Notebook service (SaaS Service) that offers free access to GPUs and TPUs among other computing resources. It does not require any setup. Colab works particularly effectively with ML, data science, and teaching.

TABLE III. HYPERPARAMETERS SPACE SEARCH CONFIGURATION FOR THE LIGHTGBM MODEL

Model	Hyperparameter settings
LightGBM	search_space ={ ’boosting_type’: Categorical([’gbdt’, ’dart’, ’goss’]),’max_depth’: Integer(1, 750), ’num_leaves’: Integer(2, 400),’learning_rate’: Real(0.01, 1.0, ’log-uniform’), ’subsample’: Real(0.1, 1.0, ’uniform’),’n_estimators’: Integer(50, 1500), ’min_child_samples’: Integer(1, 100),’colsample_bytree’: Real(0.1, 1.0, ’uniform’), ’reg_alpha’: Real(1e-9, 100, ’log-uniform’), ’max_bin’: Integer(100, 700) ’reg_lambda’: Real(1e-9, 100, ’log-uniform’), , ’max_delta_step’: Real(0, 10, ’uniform’) }

Table III presents the hyperparameters search space configuration for the LightGBM model, covering a wide range of values to optimize performance. The search space includes categorical choices for boosting_type (GBDT, DART, GOSS), and numerical ranges for key parameters such as num_leaves (2 to 400), max_depth (1 to 750), and learning_rate (0.01 to 1.0) with a logarithmic uniform distribution to explore both small and large values effectively. Other parameters include n_estimators, min_child_samples, subsample, and colsample_bytree, which control model complexity and generalization. Additionally, regularization parameters such as reg_alpha and reg_lambda are optimized within a logarithmic scale to prevent overfitting. The inclusion of max_bin and max_delta_step further refines the model’s handling of data granularity and convergence stability. This comprehensive hyperparameter tuning strategy aims to enhance LightGBM’s adaptability and accuracy across diverse Arabic sentiment analysis datasets.

C. Classification Results and Performance Evaluation

This section presents and discusses the experiments of the LightGBM classification model, metaheuristic FS algorithms, and Optuna hyperparameter optimization. The experiments in this study are divided into three main dimensions: studying the effect of ISRI stemming and Qalsadi lemmatization methods on LightGBM’ s classification, both individually and in combination with the classification efficiency on different datasets; studying the effects of metaheuristic FS algorithms; and studying the impact of Optuna hyperparameter optimization in the classification task.

Experiment 1: In the first experiment, the ISRI stemming and Qalsadi lemmatization methods and their combination with LightGBM are applied to eleven datasets as shown in table IV.

Table V outlines the hyperparameter settings used for running various metaheuristic algorithms to optimize feature selection in the sentiment analysis task. Each algorithm is configured with a common objective function, log loss, to minimize classification error, and a consistent number of iterations (20) and population size (20) to ensure fair comparisons. Specific parameter settings are applied to individual algorithms to enhance their optimization efficiency. For instance, PSO includes acceleration constants and an inertia weight for balancing exploration and exploitation. GO incorporates selective pressure, elitism, and mutation rate to guide the search process. Meanwhile, GWO, DFO, and HHO follow standard configurations focused on convergence towards optimal feature subsets. These settings ensure a robust evaluation of different optimization techniques in improving the model's performance.

Experiment 2: The second experiment is conducted to study the effects of different metaheuristic feature selection algorithms as shown in Tables VI, VII, VIII, XI, IX, X, XI.

TABLE IV. COMPARISON BETWEEN ISRI-LIGHTGBM, QALSADI-LIGHTGBM, AND ISRI-QALSADI-LIGHTGBM

Datasets	ISRI-LightGBM				Qalsadi-LightGBM				ISRI-Qalsadi-LightGBM			
	Acc.	Prc	Rc	F1	Acc.	Prc	Rc.	F1	Acc.	Prc	Rc	F1
qrci	55.6	56	56	56	56.3	56	56	56	**59**	**59**	**59**	**59**
ar_reviews_100k	62.1	63	62	62	**66.6**	**67**	**67**	**67**	57.3	58	57	57
ARABIC Dataset_2cat	**70.2**	**70**	**70**	**70**	69	69	69	69	69.6	70	70	70
ARABIC Dataset	**70**	**71**	**70**	**70**	69.8	71	70	70	69	70	69	69
mpqa-ar	60.9	61	61	61	**61.2**	**61**	**61**	**61**	59.8	59	60	59
LABR-book-reviews	64.6	65	65	64	65.3	65	65	65	**65.7**	**66**	**66**	**65**
astd-artwitter	67.3	68	67	67	**67.4**	**68**	**67**	**67**	63.1	63	63	63
ASTD	**58.5**	**59**	**58**	**58**	58	58	58	58	57.6	58	58	57
ASA_SS2030	71.1	71	71	71	71.9	72	72	72	**74.4**	**74**	**74**	**74**
AJGT	**73.9**	**74**	**74**	**74**	67.8	68	68	68	63.6	64	64	64
Company Reviews	**76.7**	72	**77**	**74**	75.2	**75**	75	73	76.6	73	77	74

TABLE V. HYPERPARAMETERS SETTINGS FOR RUNNING METAHEURISTIC ALGORITHMS

Algorithm	Hyper parameters
PSO	objective_function= log_loss, population_size=20, n_iteration=20, minimize=True, constant accelerator 1=2, constant accelerator 2=2,weight=0.9
GWO	objective_function= log_loss, population_size=20, n_iteration=20, minimize=True
DFO	objective_function= log_loss, population_size=20, n_iteration=20,method='linear', minimize=True
HHO	objective_function= log_loss, population_size=20,n_iteration=20, minimize=True
GO	objective_function= log_loss, population_size=20, n_iteration=20,selective_pressure=2 ,elitism=2,mutation_rate=0.05,minimize=True

TABLE VI. FEATURE SELECTION (FS) OF EACH ALGORITHM IN EACH DATASET

Data Set	Stemming / Lemmatization	Light GBM No .F.	PSO-Light GBM FS.	GWO –Light GBM FS.	DFO-Light GBM FS.	HHO-Light GBM FS.	GO –Light GBM FS.
qrci	ISRI	10	1	6	1	7	1
	Qalsadi	10	1	7	3	6	2
	ISRI- Qalsadi	10	3	8	5	3	5
ar_reviews_100k	ISRI	41	30	41	32	36	39
	Qalsadi	41	33	41	32	39	33
	ISRI- Qalsadi	41	30	38	31	37	31
ARABIC Dataset_2cat	ISRI	10	10	9	10	10	10
	Qalsadi	10	8	10	9	8	8
	ISRI+ Qalsadi	10	9	8	9	9	8
ARABIC Dataset	ISRI	10	10	10	9	10	8
	Qalsadi	10	10	9	10	9	8
	ISRI- Qalsadi	10	8	8	8	7	9
mpqa-ar	ISRI	16	8	12	12	9	8
	Qalsadi	16	9	15	11	11	12
	ISRI- Qalsadi	16	10	16	9	12	9
LABR-book-reviews	ISRI	42	17	34	20	26	22
	Qalsadi	42	19	39	22	31	16
	ISRI- Qalsadi	42	14	41	21	30	23
astd-artwitter	ISRI	10	7	9	10	10	9
	Qalsadi	10	5	9	5	7	5
	ISRI- Qalsadi	10	3	10	3	7	6
ASTD	ISRI	14	6	10	4	8	6
	Qalsadi	14	6	6	6	6	6
	ISRI- Qalsadi	14	2	11	5	8	6
ASA_SS2030	ISRI	21	14	18	10	13	11
	Qalsadi	21	14	31	15	16	27
	ISRI-Qalsadi	21	13	15	13	12	12
AJGT	ISRI	8	3	7	3	4	6
	Qalsadi	8	3	6	4	5	3
	ISRI- Qalsadi	8	1	5	1	4	3
Company Reviews	ISRI	8	7	8	7	7	5
	Qalsadi	8	6	7	6	6	7
	ISRI- Qalsadi	8	5	7	5	5	5

TABLE VII. COMPARISON BETWEEN ISRI-PSO - LIGHTGBM, QALSADI-PSO - LIGHTGBM, AND ISRI-QALSADI-PSO - LIGHTGBM

Datasets	**ISRI-PSO –LightGBM**				**Qalsadi-PSO- LightGBM**				**ISRI-Qalsadi-PSO - LightGBM**			
	Acc.	**Prc**	**Rc**	**F1**	**Acc.**	**Prc**	**Rc**	**F1**	**Acc.**	**Prc**	**Rc**	**F1**
qrci	58.3	59	58	57	60.9	62	61	59	**64.9**	**65**	**65**	**65**
ar_reviews_100k	61.4	62	61	61	**65.6**	**66**	**66**	**66**	57.3	58	57	57
ARABIC Dataset_2cat	**70.2**	**70**	**70**	**70**	69	69	69	69	69.6	70	70	70
ARABIC Dataset	**70**	**71**	**70**	**70**	69.8	71	70	70	69.5	70	70	69
mpqa-ar	61.4	61	61	61	**61.7**	**61**	**62**	**61**	**61.7**	**61**	**62**	**61**
LABR-book-reviews	65.8	66	66	65	65.3	65	65	65	**67.7**	**68**	**68**	**67**
astd-artwitter	67.6	68	68	68	**69.4**	**70**	**69**	**69**	66.9	67	67	67
ASTD	62	62	62	62	**63.8**	**64**	**64**	**64**	61.6	62	62	62
ASA_SS2030	75	75	75	75	**75.7**	**75**	**77**	**75**	**75.7**	**75**	**76**	**76**
AJGT	**76.1**	**76**	**76**	**76**	71.1	72	71	71	65.8	66	66	66
Company Reviews	**76.7**	**76**	**77**	**74**	75.4	74	75	73	76.6	74	77	74

TABLE VIII. COMPARISON BETWEEN ISRI- GWO - LIGHTGBM, QALSADI- GWO- LIGHTGBM, AND ISRI-QALSADI- GWO - LIGHTGBM

Datasets	**ISRI-GWO - LightGBM**				**Qalsadi-GWO - LightGBM**				**ISRI-Qalsadi-GWO - LightGBM**			
	Acc.	**Prc**	**Rc**	**F1**	**Acc.**	**Prc**	**Rc**	**F1**	**Acc.**	**Prc**	**Rc**	**F1**
qrci	60.9	61	61	61	57	57	57	57	**62.9**	**63**	**63**	**63**
ar_reviews_100k	62.1	63	62	62	**66.6**	**67**	**67**	**67**	57.1	58	57	57
ARABIC Dataset_2cat	**69.8**	**70**	**70**	**70**	69	69	69	69	68.5	69	68	68
ARABIC Dataset	**70**	**71**	**70**	**70**	69.2	70	69	69	69.6	70	70	69
mpqa-ar	60.5	60	60	60	**60.9**	**60**	**61**	**60**	59.8	59	60	59
LABR-book-reviews	64.7	65	65	64	64.3	64	64	64	**65.8**	**66**	**66**	**66**
astd-artwitter	**67.1**	**68**	**67**	**67**	64.7	65	65	65	63.1	63	63	63
ASTD	60.7	61	61	61	**63.8**	**64**	**64**	**64**	55.7	56	56	56
ASA_SS2030	73.9	74	74	74	72.4	72	72	72	**75.1**	**75**	**75**	**75**
AJGT	**75.3**	**75**	**75**	**75**	70.6	71	71	71	64.7	65	65	65
Company Reviews	**76.7**	72	**77**	74	75.4	73	75	73	76.4	**75**	77	**75**

TABLE IX. COMPARISON BETWEEN ISRI- DFO - LIGHTGBM, QALSADI- DFO - LIGHTGBM, AND ISRI-QALSADI- DFO - LIGHTGBM

Datasets	**ISRI- DFO - LightGBM**				**Qalsadi- DFO - LightGBM**				**ISRI-Qalsadi- DFO - LightGBM**			
	Acc.	**Prc**	**Rc**	**F1**	**Acc.**	**Prc**	**Rc**	**F1**	**Acc.**	**Prc**	**Rc**	**F1**
qrci	58.3	59	58	57	**63.6**	**64**	**64**	**64**	61.6	62	62	62
ar_reviews_100k	61.2	62	61	61	**64.8**	**65**	**65**	**65**	57.3	58	57	57
ARABIC Dataset_2cat	**70.2**	**70**	**70**	**70**	69.3	69	69	69	69.6	70	70	70
ARABIC Dataset	**70.4**	**71**	**70**	**70**	69.8	71	70	70	69.5	70	70	69
mpqa-ar	61.9	62	62	62	**62.2**	**62**	**62**	**62**	60.8	60	61	61
LABR-book-reviews	64.8	65	65	65	64.6	65	65	64	**67.5**	**68**	**67**	**67**
astd-artwitter	**67.3**	**68**	**67**	**67**	69.4	70	69	69	66.9	67	67	67
ASTD	**66.7**	**67**	**67**	**67**	63.8	64	64	64	64	64	64	64
ASA_SS2030	**75.1**	**75**	**75**	**75**	73.7	74	74	74	74	74	74	74
AJGT	**76.1**	**76**	**76**	**76**	70.8	71	71	71	65.8	66	66	66
Company Reviews	**76.7**	**76**	**77**	**74**	75.4	74	75	73	**76.7**	**74**	**77**	**75**

TABLE X. COMPARISON BETWEEN ISRI- HHO - LIGHTGBM, QALSADI- HHO - LIGHTGBM, AND ISRI-QALSADI- HHO - LIGHTGBM

Datasets	ISRI- HHO - LightGBM				Qalsadi- HHO - LightGBM				ISRI-Qalsadi- HHO - LightGBM			
	Acc.	Prc	Rc	F1	Acc.	Prc	Rc	F1	Acc.	Prc	Rc	F1
qrci	57	57	57	57	60.3	60	60	60	**62.9**	**63**	**63**	**63**
ar_reviews_100k	62	62	62	62	**66.4**	**67**	**66**	**66**	57.5	58	57	57
ARABIC Dataset_2cat	**70.2**	**70**	**70**	**70**	69	69	69	69	69.6	70	70	70
ARABIC Dataset	**70**	**71**	**70**	**70**	68.8	70	69	69	68.6	69	69	68
mpqa-ar	**60.4**	**60**	**60**	**60**	58.4	58	58	58	59.5	59	59	59
LABR-book-reviews	63	63	63	63	64.2	64	64	64	**65.9**	**66**	**66**	**66**
astd-artwitter	**67.3**	**68**	**67**	**67**	66.3	67	66	66	62.9	63	63	63
ASTD	63.8	64	64	64	63.8	64	64	64	54.1	54	54	54
ASA_SS2030	71.8	72	72	72	74.4	74	74	74	**74.9**	**75**	**75**	**75**
AJGT	58.6	59	59	59	**66.9**	**67**	**67**	**67**	65.3	65	65	65
Company Reviews	**76.7**	**76**	**77**	**74**	71.8	71	72	69	76.6	74	77	74

TABLE XI. COMPARISON BETWEEN ISRI- GO - LIGHTGBM, QALSADI- GO - LIGHTGBM, AND ISRI-QALSADI- GO - LIGHTGBM

Datasets	ISRI- GO - LightGBM				Qalsadi- GO - LightGBM				ISRI-Qalsadi- GO - LightGBM			
	Acc.	Prc	Rc	F1	Acc.	Prc	Rc	F1	Acc.	Prc	Rc	F1
qrci	58.3	59	58	57	60.3	60	60	60	**64.2**	**66**	**64**	**63**
ar_reviews_100k	60.6	61	61	60	**64.9**	**65**	**65**	**65**	57.4	58	57	57
ARABIC Dataset_2cat	**70.2**	**70**	**70**	**70**	69	69	69	69	69.5	70	70	70
ARABIC Dataset	**70**	**71**	**70**	**70**	69.5	70	69	69	69	70	69	69
mpqa-ar	61.7	61	62	62	**62.2**	**62**	**62**	**62**	60.8	60	61	60
LABR-book-reviews	66	66	66	66	65.7	66	66	65	**66.7**	**67**	**67**	**66**
astd-artwitter	66.4	67	66	66	**65.6**	**66**	**66**	**66**	65.4	66	65	65
ASTD	62.6	63	63	63	**63.8**	**64**	**64**	**64**	59.4	60	59	59
ASA_SS2030	73	74	74	74	**75.7**	**75**	**76**	**76**	74.9	75	75	75
AJGT	**74.7**	**75**	**75**	**75**	71.1	72	71	71	66.1	66	66	66
Company Reviews	**76.7**	**73**	**77**	**74**	75.7	74	76	74	76.1	73	76	74

TABLE XII. HYPERPARAMETER SETTINGS FOR RUNNING OPTUNA FRAMEWORK

Method	Hyperparameters
Optuna study	'objective': 'binary', 'metric': 'binary_logloss', 'num_leaves': trial.suggest_int('num_leaves', 2, 256) 'lambda_l1': trial.suggest_loguniform('lambda_l1', 1e-8, 10.0), 'lambda_l2': trial.suggest_loguniform('lambda_l2', 1e-8, 10.0), 'bagging_fraction': trial.suggest_uniform('bagging_fraction', 0.4, 1.0), ,'feature_fraction': trial.suggest_uniform('feature_fraction', 0.4, 1.0), 'bagging_freq': trial.suggest_int('bagging_freq', 1, 7),'min_child_samples': trial.suggest_int('min_child_samples', 5, 100)

Table XII presents the hyperparameter settings used for running the Optuna framework, which is employed to optimize the LightGBM model for Arabic sentiment analysis. The optimization process is guided by the binary classification objective with the evaluation metric set to binary_logloss, ensuring a focus on minimizing classification errors. The search space for key hyperparameters includes lambda_l1 and lambda_l2 for regularization, both explored within a logarithmic range to prevent overfitting. Structural parameters such as num_leaves (ranging from 2 to 256) and min_child_samples (ranging from 5 to 100) are fine-tuned to balance model complexity and generalization. Additionally, feature and data sampling parameters, including feature_fraction and bagging_fraction, are optimized within uniform distributions to enhance model robustness. The bagging_freq parameter, which controls the frequency of bagging operations, is also explored to improve model stability. These hyperparameter settings enable efficient and automated tuning to achieve optimal model performance.

D. Evaluation Matrix

Evaluation metrics are used to assess the effectiveness and performance of statistical or ML models. They help illustrate how well the model's predictions align with the true patterns in the dataset. The key metrics for evaluating ML models include Accuracy (Acc.), Precision (Prc.), Recall (Rc.), and the F1 score (F1). These metrics, calculated using specific equations, are critical in the context of Deep ASA evaluation metrics [16]:

$$\text{Accuracy (Acc.)} = (TP+ TN) / (TP + TN + FP + FN) \quad (1)$$

$$\text{Precision (Prc.)} = TP\ (TP + FP) \quad (2)$$

$$\text{Recall (Rc.)} = TP\ (TP + FN) \quad (3)$$

$$\text{F1 Score(F1.)} = 2 * \text{Prc.} * \text{Rc.}\ (\text{Prc.} + \text{Rc.}) \quad (4)$$

Where TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively.

The primary goal of Experiment 2 is to compare the effectiveness of different metaheuristics FS algorithms. Analyzing values allows us to identify which algorithms and feature selection techniques selected the most relevant features for each dataset that increase ASA accuracy, precision, recall, and F1 score. Comparing the results with stemming, lemmatization or a combination of them can help assess the effect of text preprocessing on feature selection. Comparing the results for all datasets can provide insights into how the algorithms s perform on different data characteristics.

Experiment 3: The second experiment is conducted to study studying the impact of Optuna hyperparameter optimization in ASA as shown in Table XIII.

TABLE XIII. COMPARISON BETWEEN ISRI-OPTUNA - LIGHTGBM, QALSADI-OPTUNA - LIGHTGBM, AND ISRI-QALSADI-OPTUNA - LIGHTGBM

Datasets	ISRI-Optuna - LightGBM				Qalsadi-Optuna - LightGBM				ISRI-Qalsadi-Optuna - LightGBM			
	Acc.	Prc	Rc	F1	Acc.	Prc	Rc	F1	Acc.	Prc	Rc	F1
qrci	64.2	64	64	64	66	65	65	65	**66.9**	**67**	**67**	**67**
ar_reviews_100k	**78**	**78**	**78**	**78**	68.2	68	68	68	59.9	60	60	60
ARABIC Dataset_2cat	**77.2**	**77**	**77**	**77**	76	76	76	76	76.5	77	77	77
ARABIC Dataset	**76.7**	**77**	**76**	**76**	75.2	76	76	76	76.2	77	76	76
mpqa-ar	64	62	62	62	**64.2**	**63**	**63**	**63**	62.5	62	62	62
LABR-book-reviews	67.6	66	66	66	68.1	68	68	68	**70.9**	**71**	**71**	**71**
astd-artwitter	70	68	68	68	**71.1**	**72**	**71**	**71**	70	70	70	70
ASTD	**64.2**	**64**	**64**	**64**	63.5	64	64	63	63	64	63	62
ASA_SS2030	75	75	75	75	76.1	76	76	76	**78.5**	**78**	**78**	**78**
AJGT	**77.8**	**78**	**78**	**78**	74.2	75	74	74	70.5	71	71	71
Company Reviews	78.2	76	78	76	77.3	75	77	75	**78.4**	**76**	**78**	**76**

Table XIV shows the optimal hyperparameter values found by Optuna for the LightGBM algorithm using stemming, lemmatization, or both methods for each dataset. The optimal hyperparameter values and accuracy scores vary between datasets, highlighting the importance of dataset-specific tuning. The choice of stemming or lemmatization can influence the optimal hyperparameters and accuracy. The relative importance of different hyperparameters can vary depending on the dataset and algorithm.

TABLE XIV. OPTUNA-LIGHTGBM HYPERPARAMETERS WITH THE BEST ACCURACY USING STEMMING / LEMMATIZATION OR BOTH METHODS FOR EACH DATASET

Data set	Algorithms	Optuna-LightGBM Hyperparameters		Trial	Acc.
qrci	ISRI-Qalsadi-Optuna - LightGBM	learning_rate	0.07899	118	66.9
		num_leaves	119		
		max_depth	17		
		min_child_samples	76		
		subsample	0.696647		
		colsample_bytree	0.865725		
		n_estimators	693		
ar_reviews_100k	ISRI-Optuna - LightGBM	learning_rate	0.035947	165	77.9
		num_leaves	119		
		max_depth	43		
		min_child_samples	53		
		subsample	0.674727		
		colsample_bytree	0.524262		
		n_estimators	793		
ARABIC Dataset_2cat	ISRI-Optuna - LightGBM	learning_rate	0.072340	501	77.3
		num_leaves	227		
		max_depth	15		
		min_child_samples	26		
		subsample	0.512467		
		colsample_bytree	0.553126		
		n_estimators	860		
ARABIC Dataset	ISRI-Optuna - LightGBM	learning_rate	0.027153	424	76.7
		num_leaves	214		
		max_depth	22		
		min_child_samples	9		
		subsample	0.684298		
		colsample_bytree	0.723168		
		n_estimators	872		
mpqa-ar	Qalsadi-Optuna - LightGBM	learning_rate	0.015316	281	64.2
		num_leaves	24		

		max_depth	33		
		min_child_samples	35		
		subsample	0.766491		
		colsample_bytree	0.760554		
		n_estimators	955		
LABR-book-reviews	ISRI-Qalsadi-Optuna - LightGBM	learning_rate	0.083975	112	70.3
		num_leaves	237		
		max_depth	11		
		min_child_samples	6		
		subsample	0.620707		
		colsample_bytree	0.677474		
		n_estimators	843		
astd-artwitter	Qalsadi-Optuna - LightGBM	learning_rate	0.070423	257	71.1
		num_leaves	94		
		max_depth	6		
		min_child_samples	9		
		subsample	0.743444		
		colsample_bytree	0.774086		
		n_estimators	603		
ASTD	ISRI-Optuna - LightGBM	learning_rate	0.034131	919	64.5
		num_leaves	7		
		max_depth	48		
		min_child_samples	5		
		subsample	0.82711		
		colsample_bytree	0.581206		
		n_estimators	920		
ASA_SS2030	ISRI-Qalsadi-Optuna - LightGBM	learning_rate	0.095296	676	78.5
		num_leaves	29		
		max_depth	39		
		min_child_samples	26		
		subsample	0.946048		
		colsample_bytree	0.804287		
		n_estimators	917		
AJGT	ISRI-Optuna - LightGBM	learning_rate	0.040334	593	77.8
		num_leaves	84		
		max_depth	17		
		min_child_samples	7		
		subsample	0.736873		
		colsample_bytree	0.587376		
		n_estimators	263		
Company Reviews	ISRI-Qalsadi-Optuna - LightGBM	learning_rate	0.0256	212	78.4
		num_leaves	179		
		max_depth	26		
		min_child_samples	72		
		subsample	0.83006		
		colsample_bytree	0.71024		
		n_estimators	673		

The figures below depict bar charts summarizing the classification results and performance evaluation for the best models in the three experiments mentioned above on eleven data sets. It summarizes all classification results and performance evaluation tables. For the "qrci" dataset, the model ISRI-Qalsadi-Optuna– LightGBM achieved an accuracy score of approximately 67%. Optuna with ISRI-Qalsadi increase LightGBM's overall accuracy by 8%, but PSO metaheuristics feature selection, with ISRI stemming, increases LightGBM's overall accuracy by 6% as shown in Fig. 2.

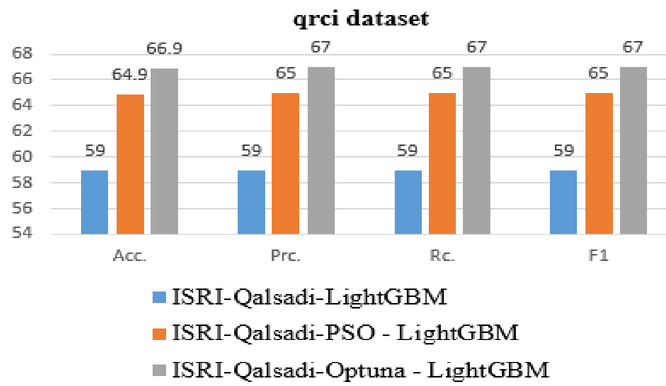

Fig. 2. Comparison between the best in the three experiments for "qrci" dataset.

For the "ar_reviews_100k" dataset, the model ISRI-Optuna - LightGBM achieved an accuracy score of approximately 78%. Optuna with ISRI stemming increase LightGBM's overall accuracy by 11% despite GWO metaheuristics feature selection, with Qalsadi lemmatization having the same value as Qalsadi-LightGBM with an accuracy score of approximately 67% as shown in Fig. 3.

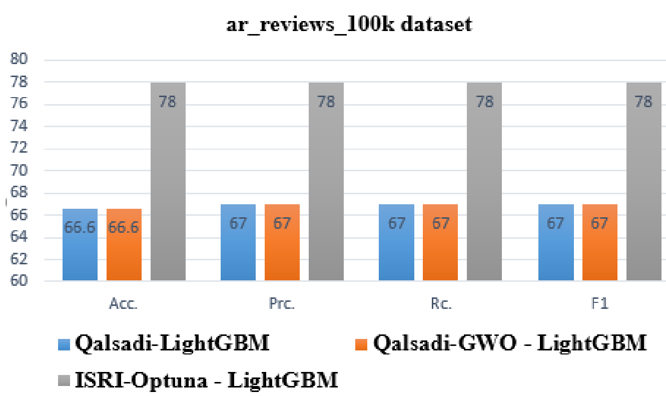

Fig. 3. Comparison between the best in the three experiments for "ar_reviews_100k" dataset.

For the "ARABIC Dataset_2cat" dataset, the model ISRI-Optuna - LightGBM achieved an accuracy score of roughly 77%. Optuna with ISRI stemming increase LightGBM's overall accuracy by 7%. Despite PSO metaheuristics feature selection with ISRI stemming, it has the same value as ISRI-LightGBM with an accuracy score of approximately 70% as shown in Fig. 4.

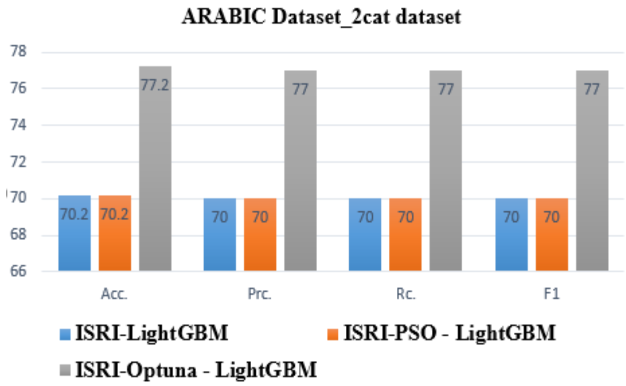

Fig. 4. Comparison between the best in the three experiments for "ARABIC Dataset_2cat" dataset.

For the "ARABIC Dataset" dataset, the model ISRI-Optuna - LightGBM achieved an accuracy score of roughly 77%. Optuna with ISRI stemming increase LightGBM's overall accuracy by 7% despite DFO metaheuristics feature selection, with ISRI stemming increasing LightGBM's overall accuracy by 0.4% as shown in Fig. 5.

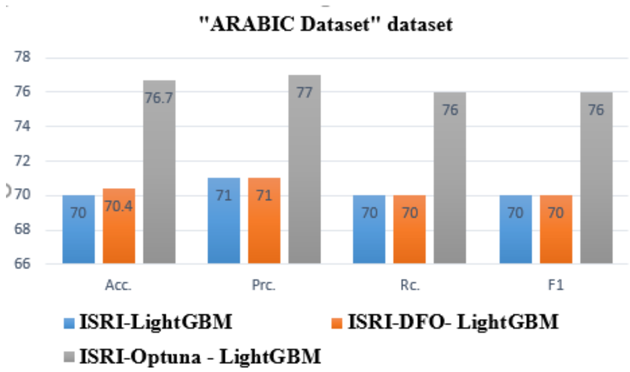

Fig. 5. Comparison between the best in the three experiments for "ARABIC Dataset" dataset.

For the "mpqa-ar" dataset, the model Qalsadi-Optuna - LightGBM achieved an accuracy score of approximately 64%. Optuna with Qalsadi lemmatization increase LightGBM's overall accuracy by 3% despite DFO metaheuristics feature selection, with Qalsadi lemmatization increasing LightGBM's overall accuracy by 1% as shown in Fig. 6.

For the "LABR-book-reviews" dataset, the model ISRI-Qalsadi-Optuna – LightGBM achieved an accuracy score of approximately 71%. Optuna with ISRI-Qalsadi increase LightGBM's overall accuracy by 5% despite PSO metaheuristics feature selection, with ISRI-Qalsadi increasing LightGBM's overall accuracy by 2% as shown in Fig. 7.

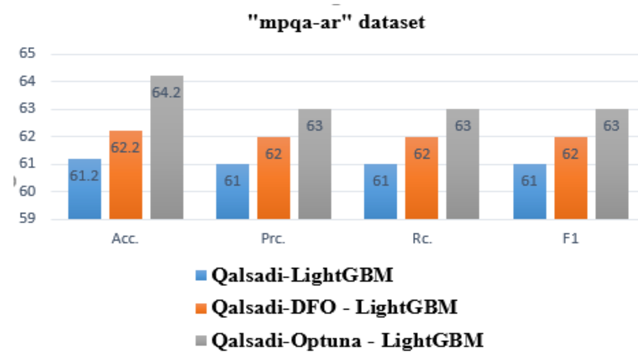

Fig. 6. Comparison between the best in the three experiments for "mpqa-ar" dataset.

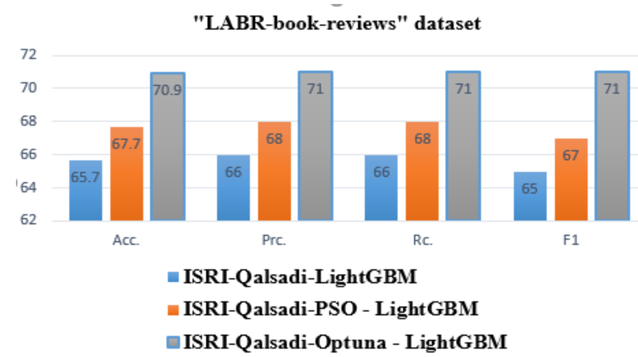

Fig. 7. Comparison between the best in the three experiments for "LABR-book-reviews" dataset.

For the "astd-artwitter" dataset, the model Qalsadi-Optuna - LightGBM achieved an accuracy score of approximately 71%. Optuna with ISRI-Qalsadi increase LightGBM's overall accuracy by 4% despite PSO and DFO metaheuristics feature selection, with Qalsadi increasing LightGBM's overall accuracy by 2% as shown in Fig. 8.

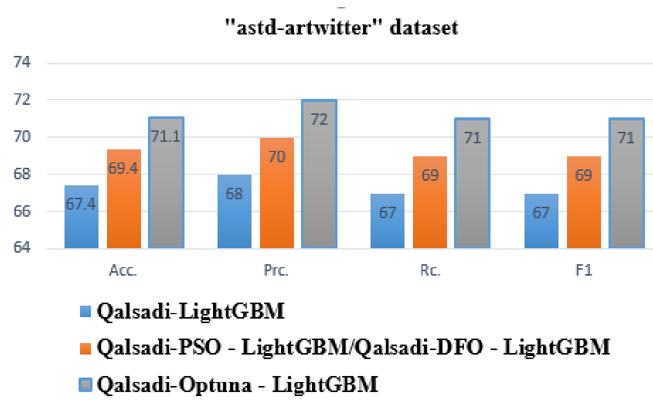

Fig. 8. Comparison between the best in the three experiments for "astd-artwitter" dataset.

For the "ASTD" dataset, the model ISRI-DFO- LightGBM achieved an accuracy score of approximately 64%. DFO metaheuristics feature selection with ISRI increases LightGBM's overall accuracy by 8% despite Optuna with ISRI increasing LightGBM's by 6% as shown in Fig. 9.

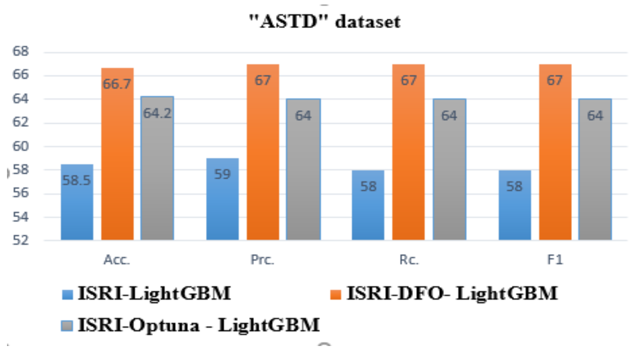

Fig. 9. Comparison between the best in the three experiments for “ASTD” dataset.

For the "ASA_SS2030" dataset, the model ISRI-Qalsadi-Optuna - LightGBM achieved an accuracy score of approximately 78.5%. Optuna with ISRI-Qalsadi increase LightGBM's overall accuracy by 4% despite DFO metaheuristics feature selection with ISRI-Qalsadi and GO metaheuristics feature selection, with Qalsadi increasing LightGBM's overall accuracy by 1.5% as shown in Fig. 10.

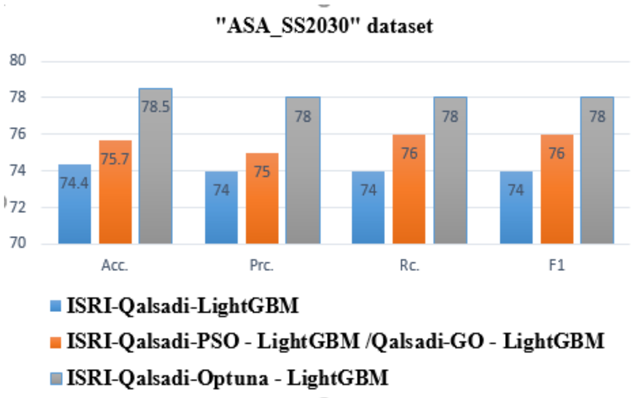

Fig. 10. Comparison between the best in the three experiments for “ASA_SS2030” dataset.

For the "AJGT" dataset, the model ISRI-Optuna - LightGBM achieved an accuracy score of approximately 78%. Optuna with ISRI increase LightGBM's overall accuracy by 4% despite DFO metaheuristics feature selection, with ISRI and PSO metaheuristics feature selection, with ISRI increasing LightGBM's overall accuracy by 2% as shown in Fig. 11.

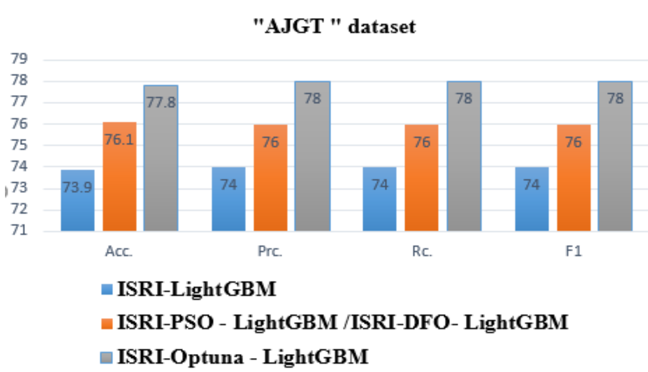

Fig. 11. Comparison between the best in the three experiments for “AJGT” dataset.

For the "Company Reviews" dataset, the model ISRI-Qalsadi-Optuna - LightGBM achieved an accuracy score of approximately 78.4%. Optuna with ISRI increase LightGBM's overall accuracy by 1.7% as shown in Fig. 12. Still, DFO metaheuristics feature selection with ISRI, PSO metaheuristics feature selection with ISRI, and HHO metaheuristics feature selection with ISRI have the same value as ISRI-LightGBM. The results demonstrate the effectiveness of hyperparameter optimization using Optuna-LightGBM in improving LightGBM's performance on ASA. By carefully tuning the hyperparameters, you can significantly improve accuracy and generalization.

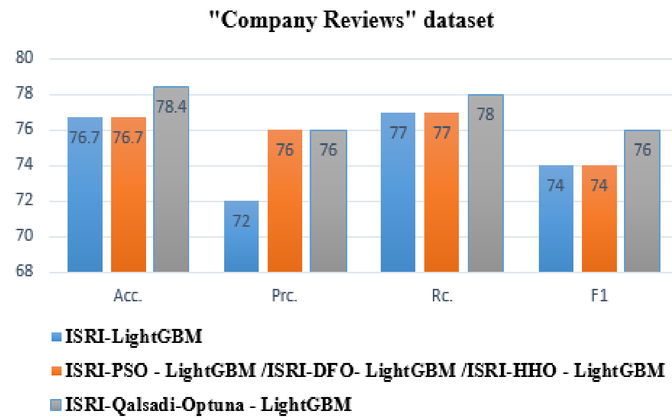

Fig. 12. Comparison between the best in the three experiments for “Company Reviews” dataset.

V. Discussion

This study contributes to the field of ASA by presenting an integrated approach to enhance the accuracy of the LightGBM model through advanced text preprocessing, feature selection, and hyperparameters optimization techniques. The complexity of the Arabic language, with its numerous dialects and MSA, poses a significant challenge for sentiment analysis applications. To address this, the study employs the ISRI stemmer and Qalsadi lemmatizer for effective text preprocessing, alongside metaheuristic FS algorithms such as PSO, GWO, and others to identify the most informative features and reduce noise in the data. Additionally, the study leverages the Optuna framework for hyperparameters tuning, aiming to achieve an optimal balance between computational efficiency and model performance. The findings demonstrate that combining these methodologies can enhance the classification accuracy of LightGBM by up to 11%, highlighting the effectiveness of these strategies in improving ASA.

However, ASA faces numerous challenges that contribute to low accuracy compared to other languages, such as English. These challenges include the morphological complexity of the language, characterized by extensive inflection, derivation, and multiple word forms that increase the difficulty of automated text analysis. Additionally, the scarcity of high-quality labeled datasets covering diverse Arabic dialects limits the model's ability to generalize effectively across various user demographics. The coexistence of MSA and dialectal Arabic, informal writing styles on social media, and the lack of standardized linguistic resources further complicate the analysis. This study provides a valuable contribution by exploring potential solutions to these issues, such as using different stemming and lemmatization methods, optimizing models through feature selection, and fine-tuning hyperparameters to achieve higher accuracy in real-world applications.

VI. CONCLUSION

Analyzing Arabic content poses challenges due to the language's complexities, morphological features, inadequate resources, and the absence of suitable corpora. This study delves into the effectiveness of various preprocessing techniques—ISRI stemming, Qalsadi lemmatization, and their combination—on ASA. The study's primary focus is on the LightGBM classifier, and It systematically compared these methods to find that each preprocessing approach contributes positively to sentiment classification accuracy. Depending on the dataset, using metaheuristic feature selection algorithms significantly enhanced the performance of the LightGBM model by identifying the most relevant features, thus reducing noise and improving LightGBM's classification efficiency between 0 and 8%. PSO metaheuristic feature selection algorithm with suitable stemming between ISRI and Qalsadi or a combination for LightGBM achieves superior results than GWO, DFO, HHO, and GO metaheuristic feature selection in more than 60% of used datasets. DFO metaheuristic feature selection algorithm with suitable stemming between ISRI and Qalsadi or a combination for LightGBM achieves superior results than other metaheuristic feature selection in more than 35% of used datasets. Applying the Optuna hyperparameter optimization framework further demonstrated the potential to refine LightGBM model parameters, effectively resulting in substantial performance gains. Depending on the dataset, Optuna using suitable stemming between ISRI and Qalsadi or a combination improves LightGBM's accuracy by between 2 and 11% and achieves superior results than PSO, GWO, DFO, HHO, and GO metaheuristic feature selection in more than 90% of used datasets. Our findings highlight the critical role that preprocessing and optimization strategies play in ASA. These methodologies improve classification accuracy and highlight the LightGBM model's robustness in this domain. ASA faces challenges such as the morphological complexity of the language, the scarcity of high-quality labeled datasets, and the coexistence of MSA and dialectal Arabic, which hinder its classification accuracy compared to languages like English, but this research explores solutions like advanced stemming, metaheuristics feature selection, and Optuna hyperparameter fine-tuning to improve performance. This research underscores the necessity for continued exploration of advanced techniques in ASA. The potential for future research to explore additional ML, DL models, transformers, and large language models to enhance ASA applications across diverse contexts and rebalance unbalanced used datasets to have higher accuracy is vast and inspiring.

ACKNOWLEDGMENT

This Project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under grand no. (GPIP: 92-611-2024). The authors therefore, acknowledge with thanks, DSR for technical and financial support.

REFERENCES

[1] A. Harjai, A. Charan, and S. Singhal, “Sentiment Analysis of Medications Review Using Deep Learning Algorithms,” in Proceedings of the 5th International Conference on Information Management & Machine Intelligence, November 2023, pp. 1–5.

[2] A. Sayed, M. M. Gomaa, and M. M. Nazier, “Sentiment Analysis on Twitter's Big Data Against the Covid-19 Pandemic Using Machine Learning Algorithms,” Inf. Sci. Lett., vol. 12, no. 8, pp. 2747–2756, 2023.

[3] S. M. Almutairi and F. M. Alotaibi, “A Comparative Analysis of Arabic Sentiment Analysis Models in E-Marketing Using Deep Learning Techniques,” Journal of Engineering and Applied Sciences, vol. 10, no. 1, 2023.

[4] F. Alzamzami and A. E. Saddik, “OSN-MDAD: Machine Translation Dataset for Arabic Multi-Dialectal Conversations on Online Social Media,” arXiv preprint arXiv:2309.12137, 2023.

[5] M. Jbel, I. Hafidi, and A. Metrane, “Sentiment Analysis Dataset in Moroccan Dialect: Bridging the Gap Between Arabic and Latin Scripted Dialect,” arXiv preprint arXiv:2303.15987, 2023.

[6] B. K. Lavine, “Feature Selection: Introduction,” Elsevier EBooks, 2009. [Online]. Available: https://doi.org/10.1016/b978-044452701-1.00028-4.

[7] B. Tran, B. Xue, and M. Zhang, “Overview of Particle Swarm Optimisation for Feature Selection in Classification,” in Simulated Evolution and Learning: 10th International Conference, SEAL 2014, Dunedin, New Zealand, Dec. 15-18, 2014, Proceedings, pp. 605-617. Springer International Publishing.

[8] A.G. Gad, K.M. Sallam, R.K. Chakrabortty, M.J. Ryan, and A.A. Abohany, “An improved binary sparrow search algorithm for feature selection in data classification,” Neural Computing and Applications, vol. 34, no. 18, pp. 15705–15752, 2022.

[9] B. Bischl et al., “Hyperparameter Optimization: Foundations, Algorithms, Best Practices, and Open Challenges,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 13, no. 2, p. e1484, 2023.

[10] T. Akiba et al., “Optuna: A Next-Generation Hyperparameter Optimization Framework,” in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623-2631, Jul. 2019.

[11] S. M. Khabour, Q. A. Al-Radaideh, and D. Mustafa, “A New Ontology-Based Method for Arabic Sentiment Analysis,” Big Data and Cognitive Computing, vol. 6, no. 2, p. 48, 2022.

[12] S. Al-Hagree and G. Al-Gaphari, “Arabic Sentiment Analysis on Mobile Applications Using Levenshtein Distance Algorithm and Naive Bayes,” in 2022 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA), October 2022, pp. 1–6. IEEE.

[13] L. Almurqren, R. Hodgson, and A. Cristea, “Arabic Text Sentiment Analysis: Reinforcing Human-Performed Surveys with Wider Topic Analysis,” arXiv preprint arXiv:2403.01921, 2024.

[14] M. Atabuzzaman, M. Shajalal, M. B. Baby, and A. Boden, “Arabic Sentiment Analysis with Noisy Deep Explainable Model,” in Proceedings of the 2023 7th International Conference on Natural Language Processing and Information Retrieval, December 2023, pp. 185–189.

[15] G. Ke et al., “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” Advances in Neural Information Processing Systems, vol. 30, 2017.

[16] E. Essa, K. Omar, and A. Alqahtani, “Fake News Detection Based on a Hybrid BERT and LightGBM Models,” Complex & Intelligent Systems, vol. 9, no. 6, pp. 6581-6592, 2023.

[17] A. Harjai, A. Charan, and S. Singhal, “Sentiment Analysis of Medications Review Using Deep Learning Algorithms,” in Proceedings of the 5th International Conference on Information Management & Machine Intelligence, pp. 1-5, Nov. 2023.

[18] F. Yang, L. Hu, and Y. Chen, “Product Competitiveness and Sentiment Analysis Based on DSM Clustering of Chinese Online Comparative Comments,” U.P.B. Sci. Bull., Series C, vol. 83, no. 4, 2021.

[19] B. O. Ondara et al., “Hybrid Machine Learning Techniques for Comparative Opinion Mining,” Indonesian Journal of Artificial Intelligence and Data Mining, vol. 6, no. 2, pp. 131-143.

[20] T. O. Omotehinwa, D. O. Oyewola, and E. G. Moung, “Optimizing the Light Gradient-Boosting Machine Algorithm for an Efficient Early Detection of Coronary Heart Disease,” Informatics and Health, vol. 1, no. 2, pp. 70-81, 2024.

[21] M. Heikal, M. Torki, and N. El-Makky, “Sentiment Analysis of Arabic Tweets Using Deep Learning,” Procedia Computer Science, vol. 142, pp. 114-122, 2018.

[22] S. Al-Azani and E. S. M. El-Alfy, “Hybrid Deep Learning for Sentiment Polarity Determination of Arabic Microblogs,” in Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, Nov. 14-18, 2017, Proceedings, Part II, pp. 491-500. Springer International Publishing.

[23] S. R. El-Beltagy et al., “Combining Lexical Features and a Supervised Learning Approach for Arabic Sentiment Analysis,” in CICLing 2016, Konya, Turkey, 2016.

[24] M. Aly and A. Atiya, “Labr: A Large Scale Arabic Book Reviews Dataset,” in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 494-498, Aug. 2013.

[25] M. Abdou, S. AbdelGaber, M. Farhan, and O. Othman, “Predicting Human Behavior Using Arabic Sentiment Analysis on Social Media: Approaches and Challenges,” Informatics Bulletin, Faculty of Computers and Artificial Intelligence, Helwan University, Published Online, 2023.

[26] I. Zeroual and A. Lakhouaja, “Arabic Information Retrieval: Stemming or Lemmatization?” in 2017 Intelligent Systems and Computer Vision (ISCV), pp. 1-6, Apr. 2017.

[27] T. Zerrouki, “Towards an Open Platform for Arabic Language Processing,” 2020.

[28] T. Zerrouki, “Qalsadi: Arabic Morphological Analyzer Library for Python,” 2012. [Online]. Available: https://pypi.python.org/pypi/qalsadi.

[29] “LightGBM: A Comprehensive Guide,” [Online]. Available: https://medium.com/@pelinokutan/lightgbm-a-comprehensive-guide-cb773cfc23b3.

[30] “LightGBM: Light Gradient Boosting Machine,” [Online]. Available: https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/.

[31] N. Van Thieu and S. Mirjalili, “MEALPY: An Open-Source Library for Latest Meta-Heuristic Algorithms in Python,” Journal of Systems Architecture, vol. 139, p. 102871, 2023.

[32] E. Emary, H. M. Zawbaa, and A. E. Hassanien, “Binary Grey Wolf Optimization Approaches for Feature Selection,” Neurocomputing, vol. 172, pp. 371-381, 2016.

[33] A. I. Hammouri et al., “An Improved Dragonfly Algorithm for Feature Selection,” Knowledge-Based Systems, vol. 203, p. 106131, 2020.

[34] T. Thaher et al., “Binary Harris Hawks Optimizer for High-Dimensional, Low Sample Size Feature Selection,” in Evolutionary Machine Learning Techniques: Algorithms and Applications, pp. 251-272, 2020.

[35] F. Hussein, N. Kharma, and R. Ward, “Genetic Algorithms for Feature Selection and Weighting: A Review and Study,” in Proceedings of the Sixth International Conference on Document Analysis and Recognition, pp. 1240-1244, Sep. 2001.

[36] B. Xiang et al., “Based on the WSP-Optuna-LightGBM Model for Wind Power Prediction,” in Journal of Physics: Conference Series, vol. 2835, no. 1, p. 012011, Aug. 2024.

[37] “Arabic Embeddings Datasets,” [Online]. Available: https://github.com/iamaziz/ar-embeddings/tree/master/datasets.

[38] “Arabic 100K Reviews Dataset,” [Online]. Available: https://www.kaggle.com/datasets/abedkhooli/arabic-100k-reviews.

[39] M. Saad, “Arabic Sentiment Analysis Dataset,” GitHub repository, 2019. [Online]. Available: https://github.com/motazsaad/arabic-sentiment-analysis/tree/master/arabic_tweets_tsv/20190413/3labels. [Accessed: Oct. 15, 2023]

[40] “LABR Dataset,” [Online]. Available: https://paperswithcode.com/dataset/labr.

[41] “Arabic Sentiment Analysis Dataset SS2030,” [Online]. Available: https://www.kaggle.com/datasets/snalyami3/arabic-sentiment-analysis-dataset-ss2030-dataset.

[42] “Arabic Twitter Corpus AJGT,” [Online]. Available: https://github.com/komari6/Arabic-twitter-corpus-AJGT.

[43] “Arabic Company Reviews Dataset,” [Online]. Available: https://www.kaggle.com/datasets/fahdseddik/arabic-company-reviews/code.