Bridging the Gap: Enhancing Kurdish News Classification with RFO-CNN Hybrid Model

Soran S. Badawi

Charmo Center for Scientific Research and Consulting – Language and Linguistic Center, Charmo University Chamchamal, Sulaimani, Kurdistan region - F.R. Iraq

***Abstract*—Effective organization and retrieval of news content are heavily reliant on accurate news classification. While the mountainous research has been conducted in resourceful languages like English and Chinese, the researches on under-resourced languages like the Kurdish language are severely lacking. To address this challenge, we introduce a hybrid approach called RFO-CNN in this paper. The proposed method combines an improved version of red fox optimization algorithm (RFO) and convolutional neural network (CNN) for finetuning CNN's parameters. Our model's efficacy was tested on two widely used Kurdish news datasets, KNDH and KDC-4007, both of which contain news articles classified into various categories. We compared the performance of RFO-CNN to other cutting-edge deep learning models such as bidirectional long short-term memory networks and bidirectional encoder representations from transformers (BERT) transformers, as well as classical machine learning approaches such as multinomial naive bayes, support vector machine, and K-nearest neighbors. We trained and tested our datasets using four different scenarios: 60:40, 70:30, 80:20, and 90:10. Our experimental results demonstrate the superiority of the RFO-CNN model across all scenarios, outperforming the benchmark BERT model and other machine learning models in terms of accuracy and F1-score.**

***Index Terms*—News Classification, Kurdish Language, Red fox optimization-Convolutional neural network, Bidirectional long short-term memory, Bidirectional encoder representations from transformers.**

I. Introduction

With the rise of social media, journalism has shifted from traditional print media to digital platforms (Kaliyar, et al., 2021). While this has increased the speed of news dissemination, the editing process is not as strictly controlled as it is with traditional media. This has resulted in an overwhelming amount of news content across different topics. Regrettably, news agencies often neglect to categorize their content, particularly in languages with limited resources. The absence of automated tools for categorizing news in these languages has made it difficult to present viewers and readers with organized content. The categorization of news is a crucial component of natural language processing, as it enables the organization of articles and stories by subject matter. Precise news classification has the potential to enhance news recommendation systems, facilitate more effective article searches and indexing, and offer valuable insights into the latest news trends and public interest in various topics (Jing and Bailong, 2021). According to Jugovac, Jannach, and Karimi (2018), news recommendation systems aim to enhance human and machine relationships by offering users personalized and relevant news articles based on their preferences and attention (Jugovac, Jannach and Karimi, 2018).

Accurately categorizing news text is essential for various industries. First, it can aid in anticipating shifts in market conditions by classifying diverse internet news sources, obtaining economic and current political news, and analyzing such information (Dai and Wang, 2021). Second, it can benefit readers by organizing and analyzing previously read news to suggest additional articles that may pique their interest (Reddy, et al., 2019). Third, categorizing news texts plays a pivotal role in detecting rumors (Verma, et al., 2021), analyzing public sentiment, and filtering spam news (Zhang, Xu and Zhao, 2020). It can help to identify deceptive information and support making informed decisions. Nevertheless, creating news classification systems can be a difficult undertaking, particularly in languages with limited labeled datasets, lexical resources, and pre-trained models.

This article presents an innovative solution called RFO-CNN that employs a hybrid approach to automatically categorize news articles based on their headlines. By combining a convolutional neural network (CNN) with an improved Red Fox Algorithm (RFO), the CNN's parameters are fine-tuned for improved accuracy. This approach is particularly effective in handling the rapidly changing nature of news content and has outperformed existing methods in news classification, including the state-of-the-art BERT, across a range of evaluation metrics. To summarize, our article makes significant contributions to the field of news categorization.

II. Related Works

Over the past few years, an array of studies has been conducted to investigate the classification of news texts.

ARO-The Scientific Journal of Koya University
Vol. XII, No. 1 (2024), Article ID: ARO.11519. 8 pages
DOI: 10.14500/aro.11519
Received: 16 January 2024; Accepted: 26 March 2024
Regular research paper: Published: 03 April 2024
Corresponding author's e-mail: soran.sedeeq@charmouniversity.org

Before delving into an examination of the proposed approach, it would be worthwhile to briefly review the methods that were previously introduced in this context.

Xie, et al. (2019) employed LSTM to extract context and sequence characteristics from Chinese news text for classification purposes (Xie, et al., 2019), while Liu, et al. (2019) presented a hierarchical model that combined LSTM and temporal convolutional networks to extract context and information sequentially (Liu, et al., 2019). Chen, Cong and Lv (2022) introduced a local feature convolution network based on BERT to capture local features and address the characteristics of lengthy Chinese news texts with significant amounts of information (Chen, Cong and Lv, 2022). Furthermore, Zhu (2021) leveraged the VSM (vector space model) to calculate text weight, and information gain, and obtain text features (Zhu, 2021). In another research by Sergio Cleger-Tamayo, a recommender approach has been suggested. Its purpose is to provide users with a ranked list of news articles based on their previous visits, the terms found in articles, and the assigned categories. The authors of the study have proposed a model that identifies semantic relationships in user access to categorize news. The findings of this study suggest that incorporating news classification into the model significantly improves its ability to predict user interest in articles (Cleger-Tamayo, Fernandez-Luna and Huete, 2012). Moreover, an innovative algorithm developed by Bouras et al. has been introduced to provide personalized and succinct articles to users through the PeRSSonal communication channel (Bouras and Tsogkas, 2009). This algorithm was designed to cater to individual preferences and deliver high-quality content that was curated for each user. While the approach is primarily content based, it also employs collaborative filtering features that adapt to the user's evolving profile over time. The authors observed a marked improvement in the system's performance when tested with real users. However, it should be noted that evaluating a summarization system is a challenging and subjective task, as acknowledged by the authors. In a study conducted by Al-Tahrawi, the Alj-News corpus – a collection of 1500 Arabic news documents – was divided into five classes for categorization. Out of these, 240 documents were used for training and 60 for testing. The author utilized the Chi-square method for this task (Al-Tahrawi, 2015). In another study, Garrido, et al. (2011) proposed a system for categorizing media content. This system was specifically designed to accurately classify real newspaper articles that have been manually tagged. By refining the support vector machine and object-based information extraction systems through testing on actual news articles, the authors were able to achieve a high level of accuracy. The authors suggest that incorporating natural language processing and semantic tools can further improve accuracy, especially in situations where the SVM training set requires frequent updates. According to the authors, this approach is user-friendly, simple, and achieves almost 99% accuracy in labeling (Garrido, et al., 2011).

It is worth mentioning that many researchers have used red fox optimization (RFO) for classification purposes. Khorami, Mahdi Babaei, and Azadeh (2021) proposed an innovative approach to COVID-19 virus detection, using a hybrid RFO combined with a CNN. They utilized chest X-ray images and a machine vision-based system to deliver highly precise outcomes. The process involved pre-processing the input X-ray images and isolating the region of interest, followed by extracting a combination of gray-level co-occurrence matrix (GLCM) and discrete wavelet transform (DWT) features from the processed images. The results demonstrated that this method outperformed other methods in the literature, making it a more effective diagnostic tool for COVID-19 virus detection. Mahesh and Hemalatha (2022) proposed a CNN-ARFO approach based on a CNN to assist users in identifying malicious applications. Pugal Priya, Saradadevi Sivarani, and Gnana Saravanan (2021) developed a RFO algorithm that employs deep long short-term memory to classify mild, moderate, and severe stages of NPDR.

In the Kurdish language, there is limited research on news detection. The existing studies focus mainly on detecting fake news. Azad, et al. (2021) sought to identify fake news in the Kurdish language. The researchers created two sets of news, one with 5000 real and 5000 fake news from Facebook pages and Sorani Kurdish websites, and another with 5000 real news and 5000 fake news generated from the real news. The study employed five machine learning classification methods, including Logistic Regression (LR), Naive Bayes, Decision Tree (DT), SVM, and Random Forest (RF), to analyze the news. The SVM classifier achieved the highest accuracy rate of 88.71%, while Random Forest scored 79.08% accuracy for only one set of manipulated text data (Azad, et al., 2021). However, the study did not label the data based on the Kurdistan Journal Syndicate rules and regulations, and only a limited number of features were extracted from the dataset. Salih and Nalbi (2023) classified fake news articles using a framework developed from text data from Kurdish news articles. However, they utilized a dataset that comprises 100,962 news articles from various domains. Of these, 50,751 articles were real, while the other 50,211 were flagged as 1 and 0 for fake news. They analyzed the dataset using three classifiers – RF, SVM, and CNNs – employing machine learning and deep learning techniques. They also utilized various feature extraction techniques, including TF-IDF, count-vector, and word embedding, to obtain different textual features from the articles. Following pre-processing steps, they fed the feature set into the classifiers. Their multi-model approach resulted in a fake news detection system that employs various models to achieve more accurate results. Their findings indicated that the CNN architecture was the most effective in identifying the most erroneous articles, with fewer false negatives a higher accuracy rate of 91%, and a higher f1-score of 95% (Salh and Nabi, 2023).

There is a significant gap in the domain of Kurdish news detection models. To bridge this gap, we have developed the RFO-CNN model. This model incorporates the Modified RFO Algorithm into a CNN architecture to optimize parameters and improve text classification. Our goal is to contribute to the creation of a culturally sensitive and robust solution for news detection in the Kurdish language by addressing the limitations of previous studies. With the RFO-CNN model, we aim to provide a more precise and effective way of determining the authenticity of news articles, thereby filling a crucial gap in the current Kurdish language research landscape.

III. Methodology

The overall RFO-CNN model is depicted in Fig. 1, which consists of several distinct phases. The first phase involves dataset collection. Phase 2 focuses on pre-processing, while Phase 3 concentrates on feature extraction and feature selection. Finally, the fourth phase is responsible for classification.

A. Dataset Collection

The experiment utilized two datasets obtained from GitHub and Mendeley. The first dataset, KDC4007, was chosen for its simplicity and well-documented nature, making it ideal for various text classification studies on Kurdish Sorani news and articles. The corpus consists of 4007 text files categorized into eight distinct categories, namely. Sports, Religion, Arts, Economics, Education, Socials, Styles, and Health, each containing 500 texts (Rashid, Mustafa and Saeed, 2017). Access to the dataset and documents is available at https://github.com/arazom/KDC-4007-Dataset/blob/master/Kurdish-Info.txt.

The second dataset, KNDH, is a massive collection of 50,000 news headlines from popular Kurdish news websites. Each category has an equal number of samples, including social, sport, science, health, and economy (Badawi, et al., 2023). This dataset is freely available to access at https://data.mendeley.com/datasets/kb7vvkg2th/2. These datasets are valuable resources for natural language processing and text classification studies, particularly in the Kurdish Sorani language.

B. Text Preprocessing

After gathering news articles, we begin the process of cleaning the datasets by eliminating irrelevant and noisy information. This involves removing unrelated words such as brackets, semicolons, full stops, and other unnecessary characters. We also remove frequent words that appear in the text, known as stop words (Badawi, 2023b). To do this, we use libraries such as KLPT, NumPy, Pandas, and Scikit-Learn. Moreover, we perform text tokenization on news headlines, which involves dividing text into smaller words or tokens. Each word in both the headlines and content is treated as a string and broken down into smaller tokens. These tokens are then used in text-mining processes. To form a vocabulary of words, the headlines are merged.

We remove diacritics, using language-specific methods, and remove various punctuation marks, special characters, and brackets from words. We also identify and remove stop words, which are connecting or joining words in a text. These frequently occurring words have little impact on the overall meaning of the text. Finally, we employ word stemming to reduce words to their stem or root. It is important to note that the stemmed word may not always be the same as its dictionary root.

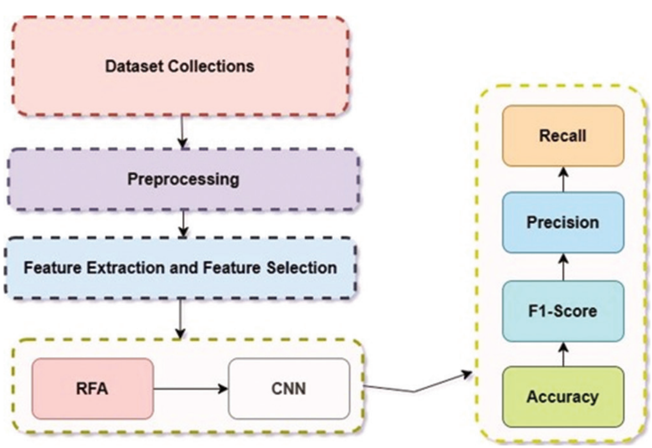

Fig. 1. The diagram of the proposed method.

C. Feature Extraction

One widely used technique in natural language processing for identifying linguistic features is TFIDF, which stands for fequency-inverted document frequency. This method is particularly helpful in classifying news headlines, as it searches for specific tokens within news content. Tokens are assigned weights based on their frequency and the importance of the document. To apply TF-IDF, the Scikit-Learn library is commonly used with document pre-processing pipelines. In this approach, the product of term frequency (TF) is evaluated by measuring the occurrence of each topic within a document and weighing it by its importance value (IDF). Equation (1) provides the formula for calculating the TF, which represents the term frequency of each topic. The IDF, which measures the criticality of the topic, is then used to create a weight matrix for each case in the dataset. Finally, the TF-IDF values can be calculated using the formula provided in equation (2) (Zhang, et al., 2019), which displays the IDF formula:

$$\mathrm{TF}(\text{term, document}) = \frac{\begin{array}{c}\text{No. of times term appears}\\ \text{in a document}\end{array}}{\text{Total words of the document}} \tag{1}$$

$$\mathrm{IDF}(\text{term}) = \frac{\text{Total number of documents}}{\text{Number of documents with term in it}} \tag{2}$$

The next step is to calculate the TF-IDF. TF-IDF is the product of the term frequency and inverse document frequency, which is shown in formula (3):

$$\text{TF}-\text{IDF (term, document)} = \text{TF (term, document)} \times \text{IDF (term)} \tag{3}$$

TF-IDF vectors can be generated using different input tokens, such as characters, words, and n-grams.

IV. Proposed Method

A. Improved RFO

The RFO algorithm is a cutting-edge optimization framework inspired by the foraging behavior of red foxes

(Połap and Wozniak, 2021). Similar to other nature-inspired algorithms, ´ RFO aims to harness the efficiency of biological systems to tackle complex problems. RFO employs a virtual population of “foxes” to represent potential solutions within a defined search space. These foxes emulate the foraging behavior of actual red foxes, skillfully balancing exploration and exploitation to discover optimal solutions. While the RFO Algorithm may offer a promising approach to optimization, traditional implementations may face challenges. Two primary issues are often associated with the conventional RFO.

B. Limited Exploration

In certain scenarios, the algorithm may face challenges in identifying the optimal solution as it only assesses a restricted segment of the search space. Such constraints can hinder the algorithm from discovering a wider range of potential and effective solutions. To address this limitation, we have introduced a novel operation that empowers exploration. This operation incorporates a transformative process that is influenced by a random gamma value, thereby introducing a stochastic element to the population of foxes. This adaptive mechanism enables the algorithm to avoid being trapped in local optima and promotes exploration of diverse regions within the search space as shown in equation (4).

$$X_i = (1 - \lambda) \bullet X_i + \lambda \bullet U(L,R) \tag{4}$$

In this equation, *Xi* represents the vector that signifies the present location of the *ith* fox in the search space. The process of updating this location involves a weighted combination of the current position and a new position. The parameter *(λ)* is a random value that is sampled from a uniform distribution between 0 and 1. This parameter adds an element of chance in the update process. The term *(1- λ). Xi* stands for the influence of the current position, where *(1-λ)* serves as a scaling factor. The term *λ.U(L,R*) signifies the influence of a randomly generated position within the specified range *[L,R]* and of the same dimension as foxes. Ultimately, this equation merges the current position of the *ith* fox with a new position determined by a random factor *(λ).* The incorporation of randomness in this way enriches exploration in the search space, addressing the issue of limited exploration. It empowers the algorithm to effect significant alterations to the fox positions, encouraging diversity and preventing premature convergence to suboptimal solutions.

C. Risk of Premature Convergence

Premature convergence occurs when an algorithm settles on a solution without fully exploring the search space, leading to less-than-optimal results. This is especially problematic for intricate optimization problems with rough terrain, where the algorithm may become stuck in local optima and fail to discover globally optimal solutions. To avoid premature convergence, we can modify the update rule by introducing diversity into the parameter updates. This can be accomplished by introducing an alpha value, which is calculated by determining the Euclidean distance between the current fox and the first fox in the population as shown in the equation. Furthermore, a beta value can be included to introduce randomness into the update process, preventing foxes from converging to similar positions.

$$a = \frac{1}{\sqrt{(X_{i[j]} - X_{0[j]})^2 + (X_{i[j+1]} - X_{0[j]})}} \tag{5}$$

To determine the value of *a*, a random selection is made from a uniform distribution that ranges between 0 and the Euclidean distance between the initial fox in the population and the *ith* fox. This means that a is determined by selecting a random value within the aforementioned range, which is determined by the distance between the position of the *ith* fox and the position of the first fox. The update rule for each dimension *j* of the *ith* fox’s position is given by this equation:

$$X_{i[j]} = X_{i[j]} + a.b.\,(R - L) \tag{6}$$

This equation updates the *jth* dimension of the ith fox’s position by adding a value. The value is calculated based on the product of a, b, and the range ($R - L$). The a and b terms introduce randomness and exploration in the update process. b is a random number uniformly distributed in the range (0,1).

Overall, we developed an enhanced iteration of the RFO algorithm, specifically tailored for fine-tuning the parameters of CNNs. This modified algorithm tackles the challenges of early convergence and insufficient exploration encountered in the original version by incorporating adaptive exploration and diverse update rules. Our results indicate that these modifications effectively improve the optimization process and demonstrate the potential of the Modified RFO algorithm in optimizing CNN parameters. Ultimately, this algorithm presents a promising approach for optimizing problems, particularly in the realm of CNN parameter optimization.

D. Applying RFO to CNN Parameter Optimization

CNN is a powerful deep-learning model that has proven to be remarkably successful in computer vision applications. In recent years, it has also demonstrated impressive results in natural language processing, specifically text classification. The architecture employed in this study encompasses the following parameters as shown in Fig. 2.

- Input dimension: The parameter known as the “input dimension” plays a crucial role in establishing the input space for the CNN model and limiting the number of words that can be included within the input text. Its value is typically derived from the maximum length of sequences or documents in the dataset, making it a key factor to consider.
- Embedding dimension: The modified RFO algorithm offers a solution that yields an optimized embedding dimension. This dimension is subsequently employed to establish the dimensionality of the embedding space. The model must possess the appropriate dimensionality to effectively capture the semantic relationships among words, enabling the model to depict them in a seamless vector space.
- Convolutional layer: This particular layer is classified as a 1D convolutional layer within a neural network. It boasts

http://dx.doi.org/10.14500/aro.11519

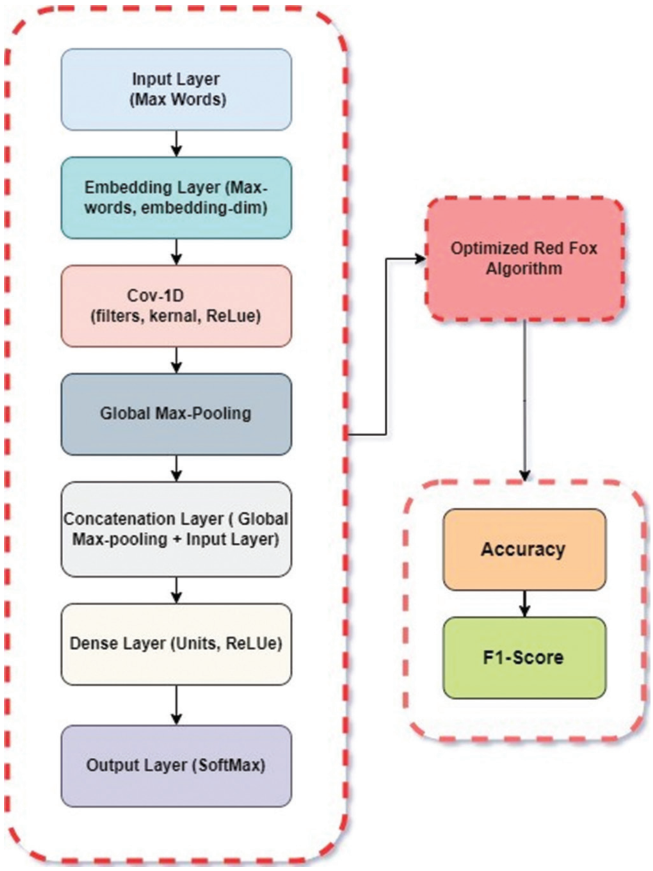

Fig. 2. The architecture of the proposed model.

numerous filters and utilizes a kernel size activated by ReLU. This layer carries out a majority of the network's computation by conducting the convolution operation, resulting in a fresh representation of the input data. During convolution, a kernel is slid over the input spectrum with a specific stride, generating a feature map that captures spatial information. Subsequently, the output feature map passes through an activation function (such as ReLU) to introduce nonlinearity to the network. This allows the network to grasp more intricate patterns found within the text data. The RFO algorithm is utilized to determine the number of filters and filter size present within the convolutional layers (Badawi, 2024).

- Global maxpooling layer: Following the convolutional layer, the architecture incorporates a Global Max Pooling layer. This layer effectively consolidates the most critical features throughout the entire sequence by extracting the maximum value from each feature map. This process condenses the spatial data into a single value for each channel. The *pool_size*, which determines the range over which the maximum value is extracted, is influenced by the RFO. The RFO is a parameter optimization technique that fine-tunes model hyperparameters, including the size of the pooling operation. The *pool_size* is dynamically adjusted based on the characteristics of the dataset to optimize the model's performance.
- Concatenation layer: In the neural network architecture, a concatenation layer follows the global max pooling layer to merge the extracted features from different branches. Specifically, this concatenation layer combines the output from the global max pooling layer with additional engineered features displayed by the input layer. Neural network architectures rely on concatenation as a crucial operation to integrate diverse information sources. By blending the high-level abstract features learned by convolutional layers with additional engineered features, the model gains a more comprehensive understanding of the input data. This enhances its capacity to capture intricate patterns and relationships. The incorporation of concatenation enriches the model's expressive power, allowing it to leverage both learned hierarchical features and domain-specific information, ultimately leading to improved overall performance and adaptability. This architectural element aligns with the holistic optimization approach facilitated by the red fox algorithm, which dynamically influences the neural network structure to achieve optimal outcomes.
- Dense layers: Our model comprises of fully connected layers, known as dense layers, at the fifth and sixth levels. With dense units neurons, these layers establish connections between neurons in the same layer and those in other layers. The activation function of the dense layer is set to Softmax to enable classification, and the model predicts class probabilities as a result. Our model has two dense layers. The first dense layer, optimized with the RFO optimization algorithm, has a variable number of units, C. The second dense layer, designed explicitly for classification, includes 11 units that correspond to the five and eight categories in the KNDH and KDC datasets, respectively.

The input dimension parameter plays a key role in defining the input space for the CNN model, as it determines the maximum number of words in the input text. Typically, this value is determined by the maximum allowable length of sequences or documents in the dataset. In addition, the optimized embedding dimension, which is derived from the RFO, is used to establish the dimensionality of the embedding space. This parameter is critical for the model to effectively capture semantic relationships between words and represent them in a continuous vector space. Finally, the total number of class parameters is used to specify the total number of distinct categories or labels that the CNN model needs to predict.

The initial step involves accessing the first fox in the population and retrieving its first element. To ensure that the embedding dimension remains a whole number. This refined embedding dimension is then utilized to configure the CNN model's architecture The RFO is used to fine-tune essential CNNs parameters. Its objective is to determine the optimal configuration for the CNN model by prioritizing significant factors such as the embedding dimension. The improved RFO returns the parameters of the first fox in the population as the optimized solution. The optimized solution is a vector of specific parameters, represented by the symbol θ. In this case, θ is a one-dimensional vector with the sole element corresponding to the embedding dimension. The embedding dimension determines the number of dimensions in the vector space in which words or tokens from the input text are embedded.

The RFO algorithm is a process that is repeated until a stopping criterion is met. This criterion can be a maximum number of iterations or a minimum fitness value. The final CNN model that is produced has the optimal values for each of its parameters. By using this process, CNN can be effectively tuned and applied for news classification in the Kurdish language. You can find the pseudocode for RFO-CNN in Algorithm.

Algorithm 1 RFO-CNN Algorithm

Begin
Initialize RFO-CNN parameters: population size, iterations, lower and upper bounds.
Generate an initial population of foxes with random values within the specified bounds. Equation (4)
Load the training data, including selected features and labels.
while num-search-iterations < stopping criteria **do**
for e **do**each fox in the population
Calculate a random value alpha based on the distance between the fox and the first fox in the population Equation (5).
Update each dimension of the fox using random value beta.
end for
Introduce a new operation:
for e **do**each fox
Calculate a random value gamma. Equation (6)
Update the fox position using a weighted combination of the current position and a new random position within the bounds.
end for
Evaluate the fitness function for each fox to select the best configuration of parameters.
if the new position > the old position. the old position == the new position.
end while
Extract the optimized parameters from the first fox in the population.
Build CNN model with the extracted parameters with predefined architecture.
Compile the CNN model with the specified loss function, optimizer, and metrics.
Train the CNN model using the selected features and the training dataset.
Evaluate the final model`s performance on the test set.
Return the optimized parameters and the trained CNN model.
end.

V. Experimental Results and Discussions

This section presents an evaluation of RFO-CNN using available datasets collected from online repositories. F-measure and accuracy are used as the metrics for comparison. To assess its performance, RFO-CNN is compared to two baseline deep learning classification models BLSTM (Badawi, 2023a) and BERT (Badawi, 2023b) and three baseline machine learning models (SVM [Saeed, et al., 2023], MNB [Saeed, et al., 2023], and K-nearest neighbors (KNN) [Tan, 2018]) that have been selected from state-of-the-art Kurdish classification. The same setup parameters configurations of the considered baseline models in the original papers are used. The experiments were conducted on Python 3.10 and Google Colab, utilizing Keras, TensorFlow, NumPy, Scikit-learn, Pandas, among other required libraries. Evaluations were carried out on a personal system with an Intel Core-i5 CPU, Windows 11, and 16 Gigabyte RAM. Preprocessing steps were performed using the KLPT Python package as proposed in (Ahmadi, 2020). The input dataset was divided into training and testing datasets and classified into four different scenarios - 90:10%, 80:20%, 70:30%, and 60:40%. To prevent overfitting, we have limited the number of epochs to two. The evaluation metrics were chosen to display the best performance of each method. Each implemented method was run N = 10 times to obtain an average value of each evaluation metric, and 5-fold cross-validation was adopted.

A. Discussion

Tables I and II offer a comprehensive analysis of diverse models utilized in the classification of Kurdish news headlines, tested under varying train-test split scenarios (90:10, 80:20, 70:30, and 60:40). The models evaluated include conventional approaches such as naïve Bayes, SVM, and KNN, as well as advanced deep learning techniques such as BLSTM, BERT, and the suggested RFO-CNN, providing a comprehensive view of their effectiveness.

Regarding KNDH dataset, it is important to mention that the RFO-CNN model consistently demonstrates superior

TABLE I
Performance Metrics for Different Models in Various Scenarios for KNDH Dataset

Scenarios		Performance metrics	
Training and test splitting	Models	Accuracy (%)	F1-score micro
90:10	Naïve Bayes	78.26	78.10
	SVM	74.71	74.69
	KNN	69.59	68.85
	BLSTM	87.50	87.50
	BERT	88.32	88.31
	RFO-CNN	89.25	89.36
80:20	Naïve Bayes	81.75	81.62
	SVM	75.89	75.88
	KNN	76.67	76.15
	BLSTM	87.80	87.85
	BERT	88.56	88.51
	RFO-CNN	89.47	89.49
70:30	Naïve Bayes	83.60	83.48
	SVM	75.19	75.18
	KNN	76.76	76.34
	BLSTM	87.32	87.36
	BERT	87.49	87.45
	RFO-CNN	88.6	88.7
60:40	Naïve Bayes	84.38	84.26
	SVM	75.99	76.02
	KNN	78.78	78.44
	BLSTM	86.27	86.29
	BERT	85.74	85.83
	RFO-CNN	87.4	87.64

SVM: Support vector machine, KNN: K-nearest neighbors, BERT: Bidirectional encoder representations from transformers, RFO-CNN: Red fox optimization-Convolutional neural network

TABLE II
Performance Metrics for Different Models in Various Scenarios for KDC-4007 Dataset

Scenarios		Performance metrics	
Training and test splitting	Models	Accuracy (%)	F1-score micro
90:10	Naïve Bayes	78.25	78.15
	SVM	73.53	73.60
	KNN	59.11	58.88
	BLSTM	81.53	81.57
	BERT	81.52	81.45
	RFO-CNN	91.62	91.54
80:20	Naïve Bayes	83.37	83.47
	SVM	80.56	80.58
	KNN	72.54	72.26
	BLSTM	80.76	81.25
	BERT	80.27	80.00
	RFO-CNN	91.12	91.11
70:30	Naïve Bayes	87.20	87.19
	SVM	82.93	82.90
	KNN	76.24	76.10
	BLSTM	76.40	74.68
	BERT	74.67	75.27
	RFO-CNN	88.40	88.42
60:40	Naïve Bayes	88.52	88.47
	SVM	84.78	84.78
	KNN	79.97	79.98
	BLSTM	71.93	72.33
	BERT	63.54	61.69
	RFO-CNN	84.14	83.19

SVM: Support vector machine, KNN: K-nearest neighbors, BERT: Bidirectional encoder representations from transformers, RFO-CNN: Red fox optimization-Convolutional neural network

performance compared to other models in all situations, showcasing its resilience and adaptability. Of particular significance is its ability to achieve accuracy scores of 89.25% and 89.47% in the 90:10 and 80:20 splits respectively, surpassing even the highly acclaimed BERT model which is recognized as a state-of-the-art in natural language processing. The F1-score micro, a metric that takes into account both precision and recall further emphasizes the exceptional performance of the RFO-CNN. It consistently exceeds 89%, and even outperforms BERT in select scenarios.

While traditional machine learning models such as Naïve Bayes, SVM, and KNN have proven effective in many scenarios, they struggle to handle the complexities of semantic analysis in Kurdish news headlines. On the other hand, the deep learning model BLSTM performs well, but it consistently falls short of the RFO-CNN's impressive results, highlighting the effectiveness of this approach. Even the highly regarded BERT, with its contextualized word embeddings and transformer architecture, is outperformed by the RFO-CNN in multiple scenarios in terms of accuracy and F1-score micro. This underscores the adaptability and optimization capabilities of the RFO-CNN, which benefits from the modified RFO algorithm's dynamic exploration and exploitation. The RFO-CNN has proven to be a successful model by leveraging the modified RFO algorithm and CNN architecture in a highly effective manner. The RFO algorithm has been designed to tackle common optimization challenges, giving it a competitive edge over traditional algorithms. In addition, the CNN's deep learning capabilities enable it to automatically extract hierarchical features from text, which significantly enhances its ability to classify news articles.

Similarly, the performance evaluation of the models trained on the KDC-4007 dataset yields valuable insights into their adaptability across various train-test split scenarios. By analyzing the accuracy and F1-score micro of the models in each scenario, we can identify distinct patterns and assess their performance on handling different data distributions. Our findings indicate that the RFO-CNN model consistently outperforms other models in the 90:10 and 80:20 splits, demonstrating its strength in managing imbalanced datasets. In these scenarios, the RFO-CNN achieves accuracy scores of 91.62% and 90.13%, respectively, showcasing its robustness and superiority. Notably, the F1-score micro for the RFO-CNN surpasses 91% in both cases, indicating its ability to balance precision and recall effectively. Several machine learning models, such as Naïve Bayes, SVM, and KNN, perform well in balanced datasets but struggle when there is a significant class imbalance, such as an 80:20 or 90:10 split. BERT is another popular model that uses contextualized embeddings, but it has lower performance in imbalanced datasets compared to the BLSTM. The RFO-CNN consistently outperforms BERT and shows adaptability and optimization capabilities. Even in datasets with class in different sizes, such as a 70:30 or 60:40 split, the RFO-CNN maintains high accuracy and F1-score. As the dataset distribution becomes more skewed, the RFO-CNN remains effective while other models struggle to maintain accuracy and F1-score. This suggests that the RFO-CNN's use of the modified RFO algorithm, combined with the CNN architecture, contributes to its adaptability and effectiveness in scenarios with varying class sizes. However, it is worth mentioning that Naive Bayes exhibits superior performance to RFO-CNN when the train-test split ratio is 60:0. This is due to the relatively small-size of the dataset containing KDC-4007, which results in fewer class numbers during the training phase. As a result, RFO-CNN is unable to identify the optimal score.

Overall, the RFO-CNN model has proven to be a highly effective solution for classifying Kurdish news headlines. It consistently outperforms both traditional and advanced models, including the state-of-the-art BERT, thanks to the integration of the modified RFO algorithm. This integration has significantly improved the model's accuracy and F1-score micro across various train-test split scenarios, making the RFO-CNN a reliable and robust tool for semantic analysis in natural language processing tasks for low-resourced languages. In particular, the RFO-CNN exhibits exceptional performance in the KDC-4007 dataset, surpassing other models in accuracy and F1-score micro across different train-test split scenarios. Its ability to perform well, especially in imbalanced datasets, makes the RFO-CNN a highly trustworthy and effective model other low-resourced language.

VI. Conclusion

In this study, we presented RFO-CNN, a novel approach for classifying Kurdish news. Our experimental results demonstrated that RFO-CNN outperforms state-of-the-art models, such as BERT, as well as other baseline machine learning methods when experimenting them on two benchmark Kurdish datasets KNDH and KDC007. The red fox algorithm enhanced CNN's parameter configuration, enabling it to detect patterns more effectively in Kurdish news articles. Our findings highlight the importance of utilizing tailored algorithms for specific low-resourced languages. RFO-CNN presents exciting possibilities for optimizing deep learning architectures for underrepresented languages and domains. Future research could explore the interpretability of the RFO-CNN's decision-making processes and identify further linguistic features that could enhance its performance in the context of Kurdish news

References

Ahmadi, S., 2020. KLPT-Kurdish Language Processing Toolkit. *In Proceedings of the Second Workshop for NLP Open Source Software (NLP-OSS)*, pp.72-84.

Al-Tahrawi, M.M., 2015. Arabic text categorization using logistic regression. *International Journal of Intelligent Systems and Applications*, 7(6), pp.71-78.

Azad, R., Mohammed, B., Mahmud, R., Zrar, L., and Sdiqa, S.J., 2021. Fake news detection in low resourced languages "Kurdish language" using machine learning algorithms. *Journal of Computational Science Education*, 12(6), pp.4219-4225.

Badawi, S., 2023. Data augmentation for Sorani Kurdish news headline classification using back-translation and deep learning model. *Kurdistan Journal of Applied Research*, 8(1), pp.27-34.

Badawi, S., 2024. Deep learning-based cyberbullying detection in Kurdish language. *The Computer Journal*, p.bxae024.

Badawi, S., Saeed, A.M., Ahmed, S.A., Abdalla, P.A., and Hassan, D.A., 2023. Kurdish News Dataset Headlines (KNDH) through multiclass classification. *Data in Brief*, 48, p.109120.

Badawi, S.S., 2023. Using multilingual bidirectional encoder representations from transformers on medical corpus for Kurdish text classification. *ARO-The Scientific Journal of Koya University*, 11(1), pp.10-15.

Bouras, C., and Tsogkas, V., 2009. Personalization Mechanism for Delivering News Articles on the User's Desktop. In: *2009 Fourth International Conference on Internet and Web Applications and Services*, pp.157-162.

Chen, X., Cong, P., and Lv, S., 2022. A Long-text classification method of Chinese news based on BERT and CNN. *IEEE Access*, 10, pp.34046-34057.

Cleger-Tamayo, S., Fernandez-Luna, J.M., and Huete, J.F., 2012. Top-N news recommendations in digital newspapers. *Knowledge-Based Systems*, 27, pp.180-189.

Dai, Y., and Wang, T., 2021. Prediction of customer engagement behaviour response to marketing posts based on machine learning. *Connection Science*, 33(4), pp.891-910.

Garrido, A.L., Gomez, O., Ilarri, S., and Mena, E., 2011. NASS: News Annotation Semantic System. IN: *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pp.904-905.

Jing, W., and Bailong, Y., 2021. News Text Classification and Recommendation Technology Basedon Wide and Deep-Bert Model. In: *2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE)*, pp.209-216.

Jugovac, M., Jannach, D., and Karimi, M., 2018. Streamingrec. In: *Proceedings of the 12th ACM Conference on Recommender Systems*, pp.269-273.

Kaliyar, R.K., Goswami, A., and Narang, P., 2021. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia Tools and Applications*, 80(8), pp.11765-11788.

Khorami, E., Mahdi Babaei, F., and Azadeh, A., 2021. Optimal diagnosis of COVID-19 based on convolutional neural network and red fox optimization algorithm. *Computational Intelligence and Neuroscience*, 2021, p.4454507.

Liu, J., Xia, C., Yan, H., Xie, Z., and Sun, J., 2019. Hierarchical Comprehensive Context Modeling for Chinese Text Classification. *IEEE Access*, 7, pp.154546-154559.

Mahesh, P.C.S., and Hemalatha, S., 2022. An efficient android malware detection using adaptive red fox optimization based CNN. *Wireless Personal Communications*, 126(1), pp.679-700.

Połap, D., and Wozniak, M., 2021. Red fox optimization algorithm. *Expert Systems with Applications*, 166, p.114107.

Pugal Priya, R., Saradadevi Sivarani, T., and Gnana Saravanan, A., 2022. Deep long and short term memory based Red Fox optimization algorithm for diabetic retinopathy detection and classification. *International Journal for Numerical Methods in Biomedical Engineering*, 38(3), p.e3560.

Rashid, T.A., Mustafa, A.M., and Saeed, A.M., 2017. Automatic Kurdish Text Classification Using KDC 4007 Dataset. In: *International Conference on Emerging Intelligent Data and Web Technologies*.

Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., and Venkatesh, B., 2019. *Content-Based Movie Recommendation System Using Genre Correlation*. Springer, Singapore, pp.391-397.

Saeed, A.M., Badawi, S., Ahmed, S.A., and Hassan, D.A., 2023. Comparison of feature selection methods in Kurdish text classification. *Iran Journal of Computer Science*, 7, pp.55-64.

Salh, D.A., and Nabi, R.M., 2023. Kurdish fake news detection based on machine learning approaches. *Passer Journal of Basic and Applied Sciences*, 5(2), pp.262-271.

Tan, Y., 2018. An Improved KNN Text Classification Algorithm Based on K-Medoids and Rough Set. In: *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, pp.109-113.

Verma, P.K., Agrawal, P., Amorim, I., and Prodan, R., 2021. WELFake: Word embedding over linguistic features for fake news detection. *IEEE Transactions on Computational Social Systems*,8(4), pp.881-893.

Xie, J., Chen, B., Gu, X., Liang, F., and Xu, X., 2019. Self-attention-based BiLSTM model for short text fine-grained sentiment classification. *IEEE Access*, 7, pp.180558-180570.

Zhang, C., Gupta, A., Kauten, C., Deokar, A.V., and Qin, X.J., 2019. Detecting fake news for reducing misinformation risks using analytics approaches. *European Journal of Operational Research*, 279(3), pp.1036-1052.

Zhang, Y., Xu, B., and Zhao, T., 2020. Convolutional multi-head self-attention on memory for aspect sentiment classification. *IEEE/CAA Journal of Automatica Sinica*, 7(4), pp.1038-1044.

Zhu, Y., 2021. Research on news text classification based on deep learning convolutional neural network. *Wireless Communications and Mobile Computing*, 2021, p.1508150.