Journal of Engineering Research 11 (2023) 100061

Contents lists available at ScienceDirect

Journal of Engineering Research

journal homepage: www.journals.elsevier.com/journal-of-engineering-research

An improved approach to Arabic news classification based on hyperparameter tuning of machine learning algorithms

Imad Jamaleddyn[a,*], Rachid El ayachi[a], Mohamed Biniz[b]

[a] Laboratory TIAD, Department of Computer Science, Moulay Slimane University, Faculty of Sciences and Technics, Beni Mellal, Morocco
[b] Laboratory LIMATI, Department of Mathematics and Computer Science, Moulay Slimane University, Polydisciplinary Faculty, Beni Mellal, Morocco

ARTICLE INFO

Keywords:
Text classification
Machine learning
Natural language processing
Hyperparameters tuning

ABSTRACT

This modern generation uses machine learning algorithms to tackle many issues, one of which is text classification. To develop classifiers that can predict the category of texts, and to optimize the output of these classifiers, hyperparameter tuning is required. The automatic optimization of the parameters of a machine learning model is referred to as hyperparameter tuning. Using NLP techniques and machine learning algorithms, we offer many methods in this research to improve the suggested classifier's accuracy values and identify the best hyperparameter. Grid search and random search are two methods for tweaking the hyperparameter. In any case, a comparison with other research works was made in order to assess the effectiveness of the suggested model and compare it against other models. The proposed approach appears to provide a robust solution for accurate Arabic text representation, interpretation, and categorization. It achieves the best performance using the CNN Arabic dataset in terms of overall accuracy, recall, precision, and F1-score by 95.16 %, 94.64 %,94.04 %, and 94.31 %.

Introduction

Techniques to manage this massive amount of data have become necessary in recent years due to the development of technology and the growth in data stored online. The most efficient way to organize the data is through classification. Data is categorized throughout the classification process so that it may later be easily retrieved, sorted, and used. Classification is a data mining approach for organizing unstructured data into classes and groups. It aids in research and planning. Classification involves learning and creating rules and models. Training datasets are large. The second phase, testing datasets and archiving classification model accuracy, follows ([12]. Supervised and unsupervised classifications exist. The best classification requires some steps. NLP techniques preprocess data to make it machine-understandable, then train and test a classifier to categorize texts using machine learning.

NLP studies how computers understand and direct natural language text or speech to do meaningful tasks. NLP professionals study how humans use language to give machine executives with the necessary tools and frameworks to understand and manipulate natural languages to perform desired tasks [16].

Machine learning allows computers to learn the same as people and improve autonomously by using real-world data and observations. Autonomous machine learning algorithms adjust their internal settings based on inputs. These are called "model parameters" [4]. Model parameters describe how input data is turned into the intended output, whereas hyperparameters structure the model. A machine learning model's performance depends on its hyperparameters. Machine learning requires hyperparameter tweaking. The machine learning algorithm must be optimized to work.

Due to the lack of resources in Arabic text classification and the increasing of data in Arabic data on the internet, it's required to build systems o to handle this large among of data, The purpose of this work to test the performance of a proposed approach based on the building of model able to classify the uncategorized Arabic texts with natural language processing NLP techniques and Machine Learning algorithms, and to accurate the efficiency of the system we added a hyperparameter tuning step to choose the best parameter for our approach. In this paper, we concentrated on hyperparameter tuning of machine learning algorithms using two techniques: Grid Search and the Random Search, however before this, we started with a preprocessing step to ensure that our data is clean and well represented in order to start the learning. The rest of the paper is organized as follows: Section 2 describes the state of the art on hyperparameter tuning and classification of Arabic texts. Section 3 presents an overview of hyperparameter tuning, Section 4 shows our architectural model and the functionality of the proposed

* Corresponding author.
E-mail address: i.jamaleddyn@gmail.com (I. Jamaleddyn).

https://doi.org/10.1016/j.jer.2023.100061
Received 20 August 2022; Received in revised form 13 March 2023; Accepted 21 March 2023

system. Section 5 discusses the experimental results. Finally, the last section presents conclusions and future work.

Related work

(Bahassine et al. [2]) used Chi-square feature selection to improve Arabic text categorization (referred to as ImpCHI, hereafter). They extended the current work to evaluate the strategy using an SVM classifier and compare it to three standard features selection metrics, mutual information, information gain, and Chi-square. For 900 characteristics, this model's best f-measures are 90.50 %.

Jamaleddyn and Biniz [11] suggested a classifier that can categorize the Arabic text utilizing a variety of vectorization and classification techniques. "BOW" and "TF-IDF" methods are employed in the steps of vectorization and text representation. Logistic Regression (LR), Support Vector Machine (SVM), and Artificial Neural Networks are the methods used for classification (ANN). The outcomes show that the created methods provided improved classification, with an ANN algorithm score of 94.57 %.

Random search (RS) was used to adjust SVM hyperparameters (Rafael [14]). (SVMs). MTH, RS, and GS were done on 70 VCI data sets, most of which were low-dimensional. The datasets and techniques supported the following claim: RS, a simple hyperparameter fine-tuning strategy, can yield SVM models with predicted accuracy comparable to meta-heuristics. RS also outperformed DF.

Overview of hyperparameter tuning

A crucial stage in the process of using machine learning in practice is optimizing hyperparameters [6]. For a machine learning method, configuring the right hyperparameter configuration needs specialized knowledge, some intuition, and frequently trial and error. The optimization issue that these hyperparameters are typically addressed as captures the predictive capability of the model created by the algorithm. For the optimization of hyperparameters in classification algorithms, various strategies have been put forth. Grid search and random search are two widely used methodologies on which our study is based.

Grid Search. It is a tuning technique that aims to identify the ideal hyperparameter values. It is a comprehensive search conducted on the values of the model's given parameters. This is the conventional approach to hyperparameter optimization, which consists of a thorough search on a given subset of the learning algorithm's hyperparameter space (Fig. 1a).

Random Search. (Fig. 1b) essentially refers to a strategy in which random sets of hyperparameters are employed to get the optimal solution for a model. It is comparable to grid search but has been found to produce comparatively superior results. Random search has the disadvantage of producing a huge variance during computation (Andradóttir [1]). Luck plays a role because the selection of parameters is purely random, and no intelligence is employed to sample these groups. It supplants the exhaustive selection of all possible combinations with their random selection.

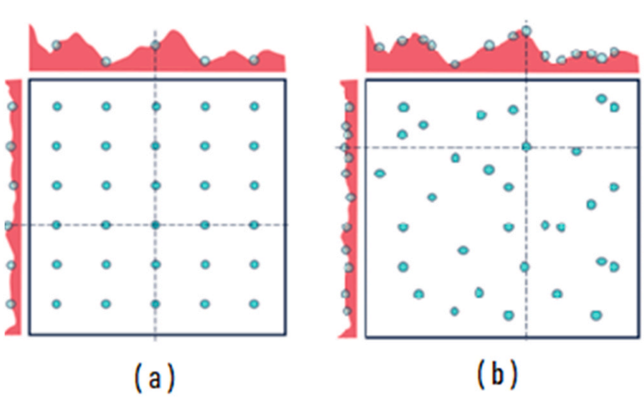

Fig. 1. A representation of a grid search combined with a random search.

Architecture of the proposed model

This section is primarily focused on the description of our Architecturally Proposed System. It discusses the reasons and requirements that led to the development of the classification system, as well as the steps that must be taken in order to construct an Arabic Text classifier based on preprocessing techniques and machine learning algorithms with hyperparameter tuning techniques. The suggested model (Fig. 2) is a classifier that can automate the categorization process and provide accurate predictions about the classes to which Arabic texts belong. It is broken up into two distinct stages, the first of which is the training stage, and the second of which is the prediction stage.

The system can be illustrated as follows (Fig. 2).

As Mentioned in (Fig. 3) before the development of the classifier in the training stage, our data moved through various procedures, the first of which was the preparation phase, which was based on NLP techniques and was intended to clean and represent our data. The data will then be segmented into portions that will be used in the training and testing phases of the process. and lastly, the construction of the model through the use of machine learning algorithms, followed by the search for the model that provides the highest level of classification. We achieved the best possible results with our classifier by tweaking the hyperparameters. In the step of prediction, we choose a new text that has not been categorized. We put it through the preprocessing stages, and then we send it on to the model that we've constructed so that it can make an accurate prediction about the text's classification. In order to understand the architecture of our model, we will be discussing in depth all of the algorithms, approaches, and strategies that were utilized in the process of putting this model into operation.

Pre-processing

Data preprocessing is an essential initial step before applying any ML process because the algorithms learn from data and the effectiveness of learning for problem solving depends greatly on the relevant data required to solve a specific problem, known as features. Machine learning is frequently referred to as feature engineering because these features are crucial for understanding and learning (Holzinger [7]). Our model's preprocessing was completed in two steps: text cleaning and feature extraction.

Data Cleaning. The practice of identifying and deleting unneeded and unnecessary data is known as data cleaning. We applied three methods to clean the data in our model: tokenization, stop word removal, and stemming [3].

Feature Extraction. The process of converting data into a format that a machine can understand is referred to as feature engineering. In this study, we applied the inverse document frequency (TF-IDF) technique and the word frequency. A numerical statistical model called TF-IDF is used to determine how important a word is in a document or group of documents (corpus). A weight is given to this term based on how frequently it appears in the document. The more times a term appears in the document, the higher its TF-IDF value will be [13].

Data splitting

Before developing the model, divide the dataset for training and testing (Fig. 4). This work uses K-fold cross-validation.

Cross-validation is one of the most common data-splitting approach in model selection. It splits data into k segments (called k-folds). The validation set is one-fold. The trained model is applied to the validation set and its predicted performance is recorded. Each part of the validation set is utilized k times (Xu Yan and Goodacre [19]).

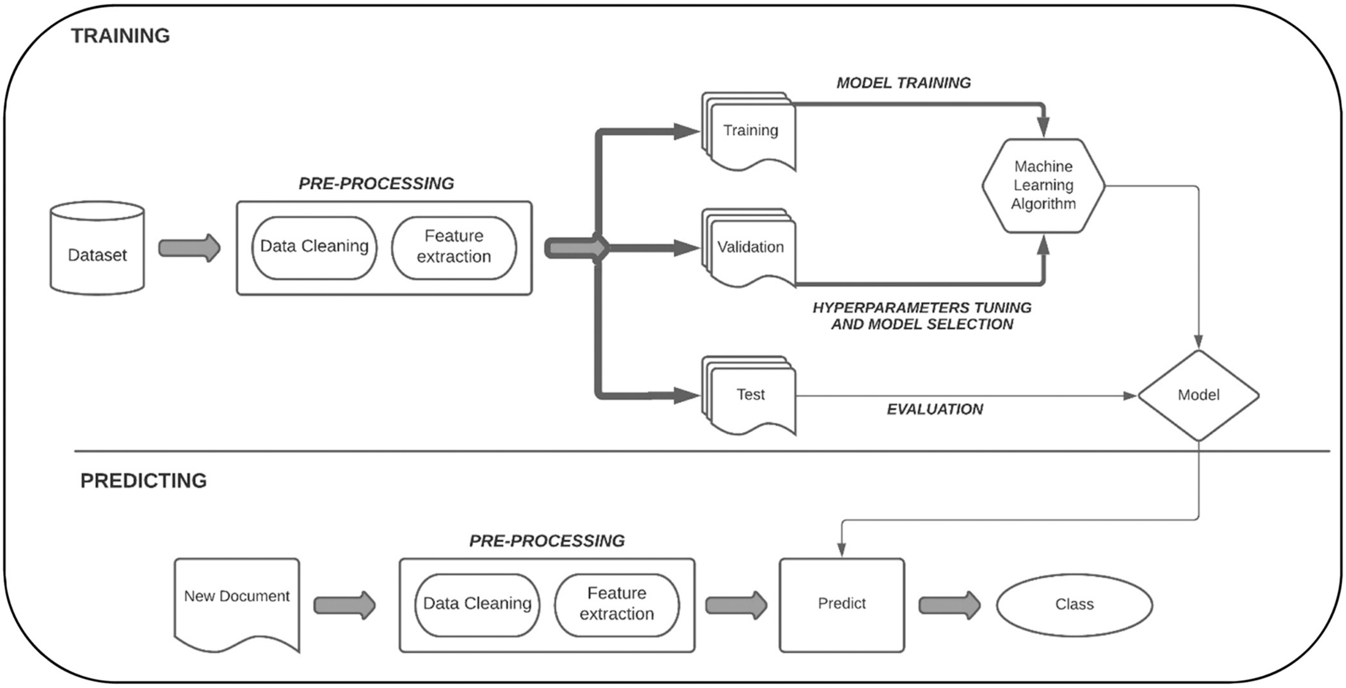

Fig. 2. An illustration of the proposed model architecture for building the Arabic text classification.

Algorithm 1: Pseudo algorithm for the Proposed classifier.

Input : Raw Dataset of Arabic texts **D**, and set of predefined categories **C**.

Output: Classified the set of testing Arabic text D_S.

for *text* $t \in D_S$ **do**
- Remove stop-words.
- Remove all diacritical marking.
- Remove digits and special symbols e.g. !, @, , etc.
- Remove all diacritical marking
- Normalize variants of a letters into a single form.
- Tokenize.
- Features Extraction and selection (TF-IDF).

end for

Split **D** into D_T for training, D_V for validation and D_S for testing.

if *Training phase* **then**
- Train and Construct the model **M** using D_T.
- Hyper-parameter tuning with (Grid Search or random Search) using D_V.

else
- `// for testing phase`
- Apply the Model **M** on D_S.
- Predict the Class $c \in \mathbf{C}$ *for each text in* D_S.

end if

Fig. 3. A Pseudo algorithm describes the steps used for building the code of the proposed model.

Machine learning and hyperparameters tuning

After pre-processing and splitting steps, the data must be trained with machine learning algorithms to build the model. This section gives a brief overview of the machine learning algorithm used and the hyperparameters tuned.

Multinomial logistic regression

MLR extends logistic regression to multi-class problems (Shreyash et al. [18]). Tuned hyperparameters, C, are the inverse of regularization strength. Regularization increases with lower values. Multi-Class: Multinomial is used since the problem has numerous classes. Solver

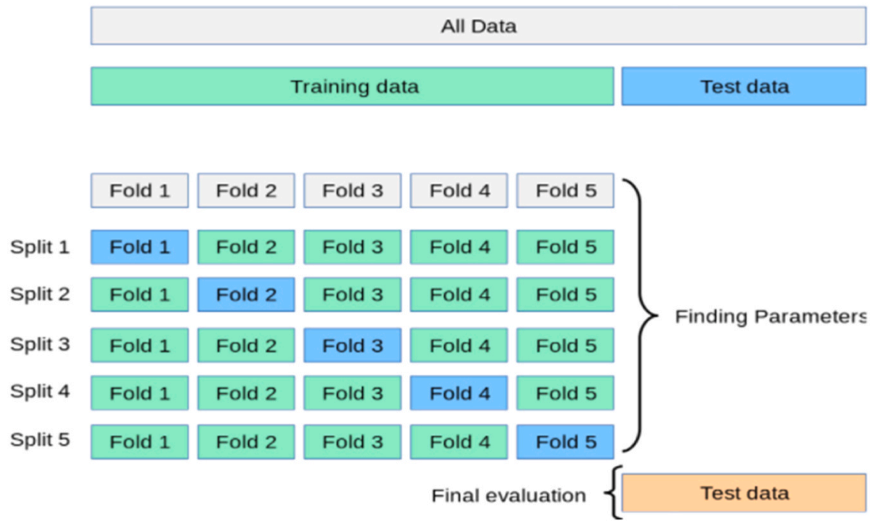

Fig. 4. An illustration of the K-fold Cross Validation technique.

Table 1
Hyperparameter used in MLR.

Parameter	Values
C	0.001, 0.01, 0.1, 1, 10, 100, 1000
Penalty	l1, l2, elasticnet, none
Multi_class	Auto, ovr, multinomail
Solver	newton-cg, lbfgs, sag, saga

Table 2
Hyperparameter used in SVM.

Parameter	Values
Kernel	linear, poly, rbf, sigmoid
C	1, 2, 3, 300, 500
Gamma	1, 0.1, 0.01, 0.001, 0.0001

Table 3
Hyperparameter used in ANN.

Parameter	Values
Epochs	10, 30, 50
BatchSize	10, 20, 40
Activation	relu, softmax, tanh, sigmoid
Optimizer	SGD, Adadelta, RMSprop, Adagrad, Adam

algorithm. Newton-cg, sag, saga, and lbfgs handle multinomial loss only for multi-class classification. Penalization norm. Newton-cg, sag, and lbfgs support l2 penalty.

Support vector machine

SVMs are used for classification in supervised machine learning. Classification is done by finding a separation hyperplane that maximizes the distance between the classes' nearest points. The margin center is the optimum separation hyperplane, while the margin points are support vectors. optimum separation [5,15]. Hyperparameters tuned below C: Penalty error word. Kernel type: Kernel coefficient: Gamma.

Artificial neural network

An artificial neural network (ANN) is a network of artificial neurons that stores and shares experience [9,10]. The input layer processes the dataset's features, and the hidden layer(s) disseminate the data's learned models to improve them [11]. Tuned hyperparameters: Epochs: The number of times each training example went through the network during training. Each gradient descent (GD) update trains BatchSize instances. Defines how input connections and bias term trigger a neuron or group of neurons ("spike") (s). Optimizers: Algorithms that change neural network parameters to reduce losses.

Results and discussion

This section shows the results obtained by our models built by machine learning algorithms and after optimizing them by hyperparameter tuning techniques. we also present a discussion based on the comparison between the proposed models and other models existing in the literature. The training and testing of our models were performed on the following dataset.

Table 4
The Optimal hyperparameters obtained by the Grid Search.

MLR		SVM		ANN	
Parameter	*Value*	*Parameter*	*Value*	*Parameter*	*Value*
C	100	Kernel	500	Epochs	40
Penalty	L2	C	0.01	BatchSize	10
Multi_class	Ovr	Gamma	Rbf	Activation	softmax
Solver	Newton-cg			Optimizer	adam

Table 5
The Optimal hyperparameters obtained by the Random Search.

MLR		SVM		ANN	
Parameter	*Value*	*Parameter*	*Value*	*Parameter*	*Value*
C	1000	Kernel	300	Epochs	10
Penalty	None	C	0.1	BatchSize	40
Multi_class	Auto	Gamma	Rbf	Activation	Sigmoid
Solver	Saga			Optimizer	Adagrad

Table 6
The Results obtained with Grid Search.

Accuracy (%)			Precision (%)			Recall (%)			F1-Score (%)		
MLR	*SVM*	*ANN*	*MLR*	*SVM*	*ANN*	*MLR*	*SVM*	*ANN*	*MLR*	*SVM*	*ANN*
93.92	92.03	***94.97***	93.58	91.22	***94.23***	92.92	91.41	***94.00***	93.21	91.30	***94.11***

Table 7
The Results obtained with Random Search.

Accuracy (%)			Precision (%)			Recall (%)			F1Score (%)		
MLR	*SVM*	*ANN*	*MLR*	*SVM*	*ANN*	*MLR*	*SVM*	*ANN*	*MLR*	*SVM*	*ANN*
94.57	94.28	***95.16***	94.18	94.34	***94.64***	93.43	93.69	***94.04***	93.77	93.99	***94.31***

Table 8
The execution time of random search and grid search.

Grid Search (h)			Radom Search (h)		
MLR	*SVM*	*ANN*	*MLR*	*SVM*	*ANN*
12,91	9,68	**17.31**	2,36	1,68	***18.75***

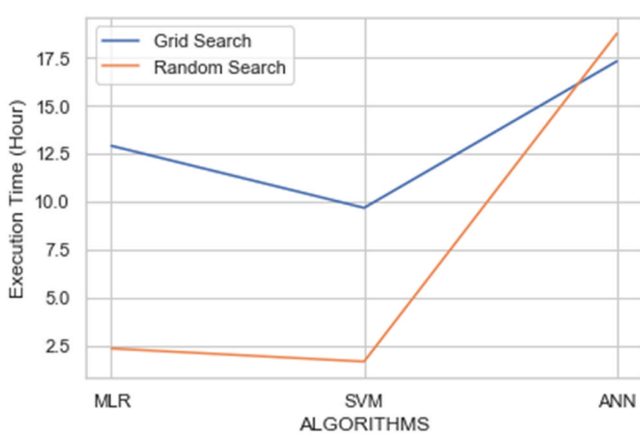

Fig. 5. Comparison of the Execution time between the used algorithms.

Dataset

CNN Arabic corpus: obtained from cnnarabic.com, comprises 5070 text documents. Each text file falls into one of six categories (Business 836, Entertainments 474, Middle East News 1462, Science & Technology 526, Sports 762, World News 1010). After removing stop words, the corpus comprises 2,241,348 (2.2 M) words and 144,460 district keywords [17].

Evaluation metrics

To evaluate the model's performance, it is required to apply evaluation metrics to estimate the model's capabilities and to determine the optimal model based on these metrics. We utilized Accuracy, Precision, Recall, and F1-score in this paper. In general, the accuracy metric measures the proportion of accurate predictions relative to the total number of instances anticipated. Precision measures the proportion of accurately predicted positive models relative to the total number of positive models predicted. Recall is used to measure the proportion of correctly categorized positive models. F1score measures the harmonic mean of recall and precision levels [8].

Results and analysis

This paper presents numerous experiments based on NLP preparation techniques, such as data cleansing and feature extraction. In addition, MLR, SVM, and ANN are the machine learning methods utilized in this work. And to optimize the parameters used by these algorithms, we employed random search and grid search as tuning approaches for hyperparameters. Each algorithm's hyperparameters are listed in Tables 1–3.

Experiment 1: random search vs grid search
For tuning the proposed hyperparameters in this experiment, we

Table 9
Comparison part between proposed model and other works.

Models	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
(Bahassine et al. [2])	–	90.80	90.50	90.50
Previous Model	94.57	94.24	93.62	93.71
Proposed Model	*95.16*	*94.64*	*94.04*	*94.31*

implemented a tuning on our dataset with the three algorithms (MLR, SVM, and ANN) by dividing our dataset into 10 folds using the cross-validation technique. Grid search (GS) and Random search (RS) were applied as two hyperparameter tuning approaches. The obtained optimal hyperparameters are shown in the following tables:(Tables 4–7).

As shown in the preceding tables, the hyperparameters acquired in experiment 1 are (100, l2, ovr, and newton-cg) for MLR, (500, 0.01, and rbf) for SVM, and (40, 10, softmax, and adam) for ANN, where Grid search technique was utilized.

And after conducting a random search, we discovered. (1000, None, Auto, and Saga) for MLR, (300, 0.1, and Rbf) for SVM, and (1000, None, Auto, and Saga) for MLR (10, 40, Sigmoid and Adagrad).

Experiment 2: Evaluation metrics of Random Search and Grid Search.

After finding the optimum hyperparameters for each algorithm with both approaches (GS and RS), in this experiment 2, we applied these tuned hyperparameters to train and evaluate the model, by calculating the performance metrics for the three algorithms used.

Comparing the two implemented techniques, GS and RS, based on the findings obtained from the evaluation of the models, we find that RS achieved the highest accuracy for the three algorithms, including ANN (95.16 %), MLR (94.57 %), and SVM (94.28 %). The accuracy achieved with GS for ANN is 94.97 %, while MLR and SVM achieve 93.92 % and 92.03 %, respectively. The same observation holds true for the precision, recall, and F1 score values.

On the basis of these results, we may conclude that RS facilitates the development of a stronger classifier compared to the GS technique. We observe that ANN using both techniques has the highest classification accuracy rate. We observe that GS's results are closer to those of RS. And to improve this comparison, we propose a third experiment based on a study of each algorithm's execution time using these two techniques.

Experiment 3: Evaluation metrics of Random Search and Grid Search.

In this experiment we calculated and compared the execution time for the three algorithms with the two techniques used RS and GS. The results are summarized in the table below:

In this experiment, we observe that as mentioned in Table 8 and illustrated in (Fig. 5) the execution time differs significantly for the three algorithms. for GS the best execution time is obtained by SVM with a time of 9.68 h, MLR yields 12.91 h and ANN gives 17.31 h. The SVM gives an execution time of 1.68 h with the RS, MLR provides 2.36 h and ANN yields 18.75 h.

As mentioned in (Fig. 5), the execution time of MLR with RS is much lower than that of GS, we observe the same point for SVM, but for ANN the execution time with RS is about one hour higher than that of GS.

Experiment 4: Comparison of the proposed model with other models

This section compares papers using the same dataset. Bahassine et al. [2] employed Chi-square, Information Gain (IG), ImpCHI, and Mutual Information (Mi) vectorization methods and two algorithms, Decision Tree (DT) and Support Vector Machine (SVM), with different feature sizes. The second paper [11] developed Arabic text classification approaches using feature extraction (bag-of-words BOW and inverse document frequency TF-DIF) and three machine learning algorithms (BOW, TF-DIF, and SVM) with their default parameters (MLR, SVM and ANN). To compare works, we choose their best values. Additionally, to highlight our development in this paper.

We have summarized the results in the following table:

As shown in Table 9, the previous model in [11] gave better results than (Bahassine et al. [2]), this refers to the approaches used on this model such as feature extraction (Bow and TF-IDF) and algorithms (MLR, SVM and ANN), to improve this previous model, hyperparameter tuning techniques are used to increase the efficiency rate of these approaches like (Grid search and Random search). and as we can see in Table 9, the hyperparameter tuning techniques yielded the best results than the compared models.

Conclusion

The tuning of hyperparameters is a technique used to optimize the parameters provided by machine learning algorithms, allowing for the efficient building of models and identification of the correct classification. This study aimed to compare the effectiveness of hyperparameter tuning techniques (random search and grid search) on Arabic news classification. The results showed that random search outperformed grid search in terms of accuracy, precision, recall, and F1-score, as well as providing faster execution times for some of the algorithms. This study contributes to the literature by demonstrating that the effectiveness of random search is not limited to specific languages or datasets but can also be applicable to Arabic news classification. Furthermore, this study highlights the importance of hyperparameter tuning techniques for improving the performance of machine learning algorithms on Arabic text data. Future research could explore other hyperparameter tuning techniques or investigate the effectiveness of different feature extraction methods on Arabic news classification. Overall, this study provides insights for researchers and practitioners working on Arabic text classification tasks.

Funding and acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Andradóttir, Chapter 20 an overview of simulation optimization via random search, Handbooks in Operations Research and Management Science, 2006, pp. 617–631, https://doi.org/10.1016/s0927-0507(06)13020-0
[2] S. Bahassine, A. Madani, M. Al-Sarem, M. Kissi, Feature selection using an improved Chi-square for Arabic text classification, J. King Saud. Univ. - Comput. Inf. Sci. (2018), https://doi.org/10.1016/j.jksuci.2018.05.010
[3] Nikita Desai, D.J. Landani, Stopword Identification and Removal Techniques on TC and IR applications: A Survey, 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS), Coimbatore, India 2020 (2020) 466–472, https://doi.org/10.1109/ICACCS48705.2020.9074166
[4] E. Elgeldawi, A. Sayed, A.R. Galal, A.M. Zaki, Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis, Informatics 8 (4) (2021) 79, https://doi.org/10.3390/informatics8040079
[5] Abdalla El-Habil, An application on multinomial logistic regression model, Pak. J. Stat. Oper. Res. 8 (2) (2012) 271–291, https://doi.org/10.18187/pjsor.v8i2.234
[6] M. Feurer, J.T. Springenberg, F. Hutter, Initializing Bayesian hyperparameter optimization via meta-learning, Proc. AAAI Conf. Artif. Intell. 29 (1) (2015), https://doi.org/10.1609/aaai.v29i1.9354
[7] A. Holzinger, Big data calls for machine learning. Reference module in biomedical, Sciences (2017), https://doi.org/10.1016/b978-0-12-801238-3.10877-3
[8] Mohammad Hossin, M.N. Sulaiman, A review on evaluation metrics for data classification evaluations, Int. J. Data Min. Knowl. Manag. Process 5 (2015) 01–11, https://doi.org/10.5121/ijdkp.2015.5201
[9] I.T. Hovden, Optimizing Artificial Neural Network Hyperparameters and Architecture, (2019).
[10] N.Q. Hung, M.S. Babel, S. Weesakul, N.K. Tripathi, An artificial neural network model for rainfall forecasting in Bangkok, Thailand, Hydrol. Earth Syst. Sci. 13 (8) (2009) 1413–1425, https://doi.org/10.5194/hess-13-1413-2009
[11] I. Jamaleddyn, M. Biniz, Contribution to arabic text classification using machine learning techniques, Int. Conf. Bus. Intell. (2021) 18–32, https://doi.org/10.1007/978-3-030-76508-8_2
[12] P. Koturwar, S. Girase, D. Mukhopadhyay, A survey of classification techniques in the area of big data, ArXiv, abs/1503 (2015) 07477.
[13] Kumar, Sandal & Columbus, Christopher & Scholar, Research, Various improved TFIDF schemes for term weighing in text categorization: a survey, Int. J. Eng. Res. 10 (2015) 11905–11910.

[14] Rafael Mantovani, André Rossi, Joaquin Vanschoren, Andre Carvalho, Meta-learning Recommendation of Default Hyper-parameter Values for SVMs in Classifications Tasks, CEUR Workshop Proc. (2015) 1455.
[15] David Meyer, Friedrich Leisch, Kurt Hornik, The support vector machine under test, Neurocomputing 55 (2003) 169–186, https://doi.org/10.1016/S0925-2312(03)00431-4
[16] Ramalingam Ponnusamy, A Systematic Survey of Natural Language Processing (NLP) Approaches in Different Systems, (2018), p. 4.
[17] Motaz Saad, Wesam Ashour, OSAC: Open-Source Arabic Corpora, (2010).
[18] Shreyash S. Nandgaonkar, Ameya Naik, Multi-class text classification using hyperparameter tuned models, international journal of electrical, Electron. Data Commun. (IJEEDC) (2020) 1–5.
[19] Yun Xu, Royston Goodacre, On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning, J. Anal. Test. (2018) 2, https://doi.org/10.1007/s41664-018-0068-2