applied sciences

MDPI

Article

Arabic News Classification Based on the Country of Origin Using Machine Learning and Deep Learning Techniques

Nuha Zamzami [1,*], Hanen Himdi [1] and Sahar F. Sabbeh [2,3]

[1] Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah 23218, Saudi Arabia; hthimdi@uj.edu.sa
[2] Department of Information Systems and Technology, College of Computer Science and Engineering, University of Jeddah, Jeddah 23218, Saudi Arabia; sfsabbeh@uj.edu.sa
[3] Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt
* Correspondence: nezamzami@uj.edu.sa

Abstract: With the rise of Arabic news articles published daily, people are becoming increasingly concerned about following the news from reliable sources, especially regarding events that impact their country. To assess a news article's significance to the user, it is essential to identify the article's country of origin. This paper proposes several classification models that categorize Arabic news articles based on their country of origin. The models were developed using comprehensive machine learning and deep learning techniques with several feature training methods. The results show the ability of our model to classify news articles based on their country of origin, with close accuracy between machine learning and deep learning techniques of up to 94%.

Keywords: news source; news country of origin; classification; machine learning; deep learning

check for updates

Citation: Zamzami, N.; Himdi, H.; Sabbeh, S.F. Arabic News Classification Based on the Country of Origin Using Machine Learning and Deep Learning Techniques. *Appl. Sci.* **2023**, *13*, 7074. https://doi.org/10.3390/app13127074

Academic Editors: Yujin Lim and Hideyuki Takahashi

Received: 10 May 2023
Revised: 5 June 2023
Accepted: 10 June 2023
Published: 13 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

News has been a fundamental part of human history since its inception. It provides us with a window into the current events of the world and, thus, an understanding of the world around us. Interest in the news is pervasive nowadays, mainly because digitization has allowed individuals to follow news online despite their busy schedules. Due to the abundance of articles released daily, people may concentrate on issues connected to their interests in geopolitics, religion, and work. For news consumers, obtaining reliable information is crucial given that it influences their views and potential reactions regarding events in their communities [1]. For example, Schumaker and Chen [2] researched how financial breaking news affected the price of stock markets, discovering that prices remained just as they were 20 min before the release of news articles and increased after the news release.

Freely available open data, information, and news from diverse sources have been crucial areas associated with the rise of the Internet. Recently, for example, with increasing concerns regarding COVID-19, several health organizations published daily and real-time updates about the global spread of the virus. The news articles were published and circulated globally by various established news agencies and some online news websites. The accuracy of the news published was paramount due to its enormous impact on the public and society. Verification of the news sources became necessary for the readers at that time due to the high level of fake news distributed, which caused panic and overwhelmed governments. Therefore, verifying the source of news articles to determine the credibility of the news can play an essential role in preventing the spread of inaccurate information, which may have severely negative consequences for the public, law enforcement agencies, and other stakeholders in society.

Readers are concerned with verifying the source of news since it can influence their perspective on specific topics. To achieve this, automating the identification of news sources is imperative. Natural language processing, machine learning, and deep learning

approaches can provide effective alternatives to the manual classification of text in specific categories so that identifying news articles' origins can be achieved. Assigning news articles to a specific country can be achieved through the use of classification models. Although there exists much literature on text classification topics, such as fake news classification [1,3], to our knowledge, the verification of the source of news articles in terms of their originating country has received little attention in the research domain. For example, Rao and Sachdev [4] built a classifier that classified English news articles based on their city of origin. The classifier achieved 85% accuracy and was trained on the word parts of speech as textual features. Identifying the source of news in languages for which there are few resources, such as Arabic, is more challenging. In this regard, the main contributions of this work are three-fold:

- Proposal of an automated approach for classifying Arabic news articles based on the country of origin using ML and DL models.
- Application of various ML and DL models combined with different feature extraction methods to comprehensively evaluate classification performance (see Figure 1).
- Building of a large Arabic dataset that includes news articles about Brexit.

The novel contribution of our work is that we rely on the textual content of the articles to identify the articles' country of origin without any associated metadata, such as author name, date, or newspaper name.

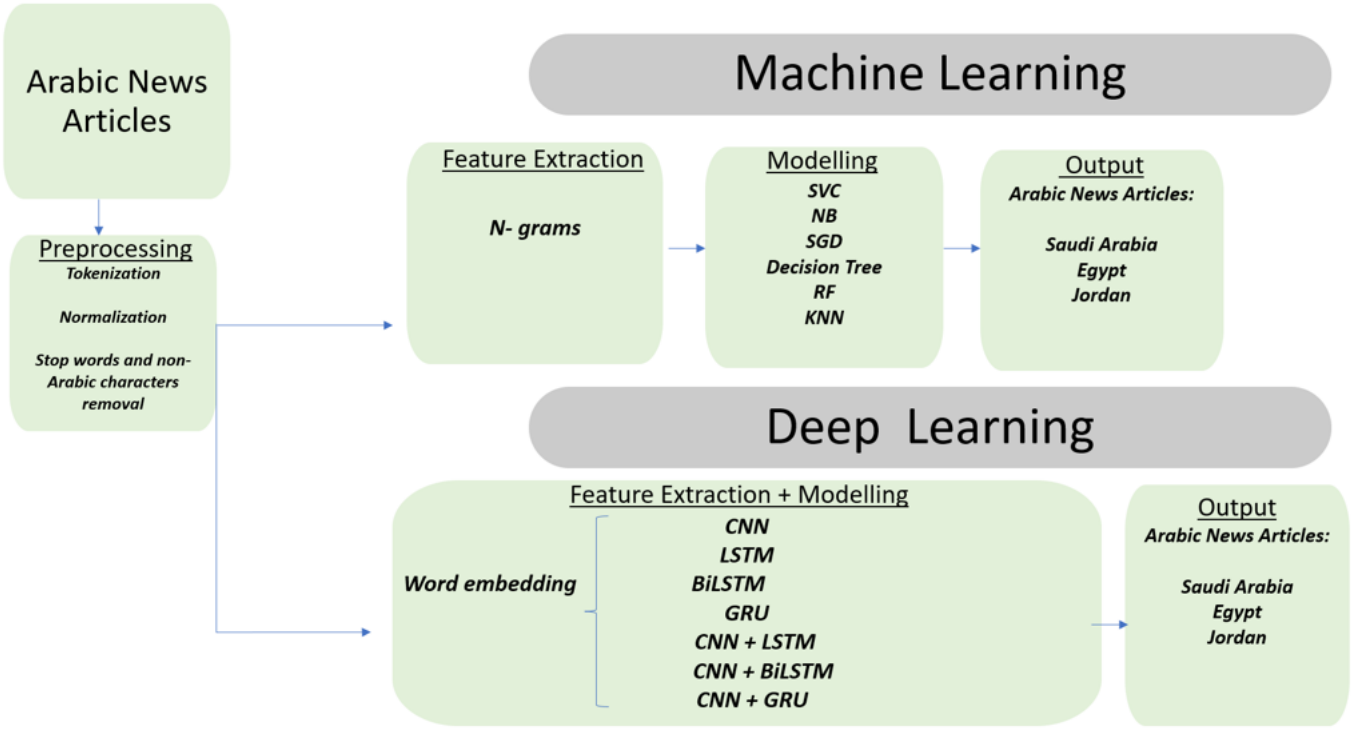


Figure 1. Arabic news classification based on country of origin model framework.

This paper starts with a literature review and then progresses to current research detailing the development of a classification model that classifies Arabic news articles based on their country of origin. The remainder of this work is organized in the following manner: In Section 2, we briefly review related research on news classification. Then we present our dataset and preprocessing technique in Section 3. We then explain the process of training the models with machine learning and deep learning techniques in Sections 4 and 5, respectively. Finally, Section 6 concludes this work with directions for future work.

2. Literature Review

Recently, several researchers have attempted to resolve problems associated with existing news text classification methods, such as their low efficiency and low classification accuracy, using machine learning and deep learning techniques. Here, we briefly review the available body of related work on news classification. In this regard, Abdulla and Awad [5] proposed a new news classification method based on deep learning convolutional neural networks (CNN). The main contribution of this research was the highlighting of how the words in the text were weighted. Instead of using word frequency criteria, Abdulla and Awad created a graph-based technique that described co-occurrent relations between the text's words and preserved structural information of the text to improve the text classification of English news articles. This method was termed the graph-of-words model, in which a graph represents every text in a dataset, and the weighting process follows the graph creation for every text. The TF-IDF criterion (term-frequency/inverse document frequency) forms the basis for weighting every word in a text if the bag-of-words model expresses the text. The datasets R8 and WebKB were used to categorize a collection of texts based on their classes from an existing set of predefined classes. Preprocessing steps were used to recognize the essential words in the dataset, including tokenization, lower-casing, stemming, noise removal, lemmatization, and stop-words removal. The significance of a word in a text was defined utilizing vertex centrality criteria, such as degree centrality and closeness. The results showed that the method's accuracy was superior as it had the highest accuracy for all the WebKB and R8 datasets.

In another study using CNN for news classification, Sreedevi et al. [6] compared a CNN with other machine learning classification techniques, including KNN, Naïve Bayes, and SVM. The authors utilized the widely used AGs news classification dataset and the Twenty Newsgroup dataset, using metrics such as accuracy, training time, and prediction time to compare the classification techniques. The authors found that SVM provided more accuracy with moderate training time and quick predictions, while KNN took less time for training. Moreover, if the number of news articles to be tested is large and less time is available for testing, then when applying CNN or SVM techniques use of bag-of-words for preprocessing should be considered.

Zhu [7] also proposed using a deep learning method, recommending CNN for news text classification to resolve the problem of low classification accuracy and low efficiency related to the existing classification methods used for news texts. The study determined the weight of the news text data by utilizing a viable system model (VSM) vector space model. Zhu utilized a hash algorithm for encoding news text data to evaluate news. Moreover, the basic structure of CNN was analyzed, and the convolutional layer of CNN was utilized to determine the change value of the convolutional kernel; news text data were trained and a news text classifier was built using CNN to complete the news text classification. It was found that using CNN to improve the accuracy and speed of news text classification was feasible as the method was shorter and faster.

Mahajan and Ingle [8] utilized Naïve Bayes for word vectorization in their study of machine learning techniques for news classification. The preprocessing methods used in this study were new tokenization, diacritic removal, stop-word removal, and word stemming, while count vectorization and TF-IDF vectorization were used for feature engineering. The researchers utilized a news classification dataset (.json), precision, recall, the F-score, and support as performance measures to compare the Naïve Bayes classifier with TF-IDF.

The findings of this study suggested that Naive Bayes can successfully categorize news, while TF-IDF was not efficient in terms of the performance measures used in the study.

In another study on news classification using Naïve Bayes, Ahmed and Ahmed [9] implemented a single-class text classifier system for online news posts. The sample dataset consisted of close to 75,000 news articles from seven websites with their tags. The authors used Naïve Bayes and several other classification algorithms, such as KNN (K-nearest neighbors), SVM (support vector machine), and LR (logistic regression) for classification.

They compared diverse classifier outcomes from various news sites on the same dataset. The study's results confirmed that Naïve Bayes showed comparatively better results of 93% accuracy, while KNN displayed the least results with 72% accuracy. The study confirmed that Naïve Bayes performed better when using various news datasets with sufficient accuracy.

Further investigation of other publications of ML classifier research revealed an important work by Saigal et al. [10], who proposed an approach to classifying news articles according to specific categories using support vector machine (SVM)-based classifiers. Specifically, in order to prove the usability and efficacy of the established methods, they tested least square (LS)-SVM, twin (TW)-SVM, and LS-TWSVM for news classification. The authors reported that LS-TWSVM outperformed the other variants of SVM.

Notably, none of the works considered in this literature review classified news articles based on their geographical locations or country of origin. In a recent study, Watanabe [11] introduced Newsmap, a semi-supervised, data-driven approach, to classify news articles based on geographical location. He relied on three classification methods: simple keyword matching, geographical information extraction systems, and a semi-supervised ML classifier that he developed. The first approach, namely, simple keyword matching, was used to extract traits from articles related to locations and people associated with the location, such as USA and Biden. These traits were then run through a geographical information extraction system that included a manually compiled dictionary of many countries with their associated people and traits. The author relied on the semi-supervised approach to remove the burden of manually classifying thousands of news articles to train the model. Using this approach, the classifier could identify places, people, and organizations associated with locations. Moreover, the classifier was able to identify the country associated with the article based on continuous scores attached to words presented in the article with minimal need for human involvement. This research demonstrated experimentally that Newsmap produced promising results of 80% for precision and recall.

Similar work was also carried out by Rao and Sachdev [4], who developed a model that classified articles based on their countries of origin. Their work relied on the part of speech tagging the articles, retaining only noun-singular (NN), noun-plural (NNS), proper-noun-plural (NNPS), and proper-noun-singular (NNP) tags. The task was to assign these tags to their corresponding articles to link them to a specific city. The authors used ML classifiers to train and test their model: Naive Bayes, SVM, and RF. The latter outperformed both previous classifiers, though the other two also achieved promising scores.

Recently, Al-Barhamtoshy et al. [12] developed an Arabic pilgrim dataset that included over 10,000 news articles about the Hajj from three Arabic-speaking countries: Saudi Arabia, Jordan, and Egypt. They also proposed a supervised machine learning model that classified Arabic articles based on their country of origin. Their model extracted four textual feature categories, including polarity, linguistics, emotions, and part of speech. When combining these features with the BoW approach, their model achieved an F-score of 85% using a support vector machine. The ML algorithms used in this study were RF, SVM, Naive Bayes, and LR.

The review, summarized in Table 1, highlighted that most recent publications in this field were concerned with classifying news articles into their domain categories. Even the small amount of research that has been conducted on classifying articles based on countries of origin has used methods of supervised machine learning based on tagging the articles with specific words related to the country, which needs manual effort. To our knowledge, no work has been conducted to test the feasibility of deep learning methods to classify news articles based on the country of origin.

Table 1. Summary of related work.

News Articles Classification Based on Domain Categories	
Study	**Approach and Algorithms**
Abdulla and Awad [5]	Deep learning—CNN
Ahmad and Ahmad [9]	Machine learning—Naive Bayes, KNN, SVM, LR
Sreedevi et al. [6]	Deep learning—CNN Machine learning—KNN,NB,SVM
Zhu [7]	Deep learning—CNN
Mahjan and Ingle [8]	Machine learning—NB
Siagal [10]	Machine learning—SVM
News Articles Classification Based on Geographical Domains	
Study	**Approach and Algorithms**
Rao and Sachdev [4]	Machine learning—SVM, NB, RF
Watanabe [11]	Semi-supervised learning
Al-Barhamtoshy et al. [12]	Machine learning—SVM, NB, RF, LR

3. Datasets and Preprocessing

In this section, we give the necessary details of the dataset used and give samples of the data. We also explain the preprocessing method used for cleaning the data.

3.1. Datasets

The review of related work was crucial in identifying gaps in the literature and how our present research can help fill those gaps. For the first step of our work, a random sample from the following dataset is composed by AlBarhamtoshy et al. [12] study was used. The dataset included 694 articles from Saudi Arabia, 685 articles from Jordan, and 695 articles from Egypt, all focusing on the topic of the Hajj; samples of articles from each country are shown in Figures 2 and 3, respectively. Statistics for the dataset are shown in Table 2.

Table 2. Hajj dataset statistics.

	Saudi	**Egypt**	**Jordan**
No. of articles	694	695	685
Avg. no. of sentences	7	7	7
Avg. no. of characters	2134	1809	1480
Avg. word length (number of characters in word)	6.14	6.02	5.99
Avg. sentence length (number of words)	52	44	35

To further evaluate the effectiveness of our model in analyzing and categorizing news articles on an entirely different topic, and to demonstrate its versatility and potential for application in various fields beyond just the Hajj topic, we assembled a large Arabic dataset regarding the topic of Brexit. The dataset included news articles from various Arabic news agencies from the same Arabic countries the Hajj dataset was collected from, i.e., Saudi Arabia, Egypt, and Jordan, providing a diverse range of articles on the topic. The dataset included 486 Saudi, 481 Egyptian, and 485 Jordanian articles from twelve Arabic news agencies, collected from published articles from January 2019 to January 2020. Table 3 displays the dataset statistics.

Table 3. Brexit dataset statistics.

	Saudi	Egypt	Jordan
No. of articles	486	481	485
Avg. no. of sentences	8	10	8.8
Avg. no. of characters	1163	1995	1404
Avg. word length (number of characters in word)	6.11	5.9	6.03
Avg. sentence length (number of words)	25	30	32

اطلع الأمير خالد الفيصل مستشار خادم الحرمين الشريفين أمير منطقة مكة المكرمة، على خطة قوات أمن المنشآت لموسم حج العام الحالي، التي سينفذها قرابة 9 آلاف ضابط وفرد .استمع الأمير خالد الفيصل لدى استقباله اليوم الإثنين، قائد قوات أمن المنشآت اللواء سعد بن حسن الجباري وعدد من ضباط القوات، إلى شرح عن خطة أمن المنشآت في الحج، التي تتولى إدارة وتنظيم الحشود في قطارات المشاعر إلى جانب مساندة قوات الأمن الأخرى في مهامها لتأمين وحماية ضيوف الرحمن . أوضح قائد قوات أمن المنشآت، أنه جرى عقد العديد من ورش العمل والمحاضرات لتعزيز مستوى الجاهزية لهذه المهمة الدينية العظيمة. في نهاية اللقاء تسلم الأمير خالد الفيصل، تقريرا عن مشاركة قوات أمن المنشآت في مهمة الحج العام الماضي وما تم تحقيقه وتوفيره لتحقيق أعلى درجات الأمن والحماية لمحطات القطار بالمشاعر المقدسة، بهدف تأدية ضيوف الرحمن لمناسكهم بكل راحة واطمئنان، كما تسلم سموه كتيباً إحصائياً عن مهام وأعمال قوات أمن المنشآت بمناطق المملكة المختلفة.

Figure 2. Sample of KSA articles about the Hajj.

قالت اللجنة البرلمانيةلخروج بريطانيا من الاتحاد الاوروبي انها تأمل ان تنشر قبل منتصف شباط القادم "كتابا ابيض" وقالت اللجنة ان مجلس العموم (البرلمان)يجب ان يبلغ بذلك قبل .يتضمن خطتها المفصلة للخروج من الاتحاد الاوروبي وقت كاف من بدء اجراءات الخروج من الاتحاد رسميا. واضافت اللجنة في تقرير مرحلي حول بريكست "نأمل ان نرى وقالت انها تأمل ونظرا للاهمية الكبرى لعملية اطلاق عملية ."هذه الخطة في منتصف شباط 2017 على ابعد حد الانسحاب، ان "تنشر الحكومة هذه الخطط بشكل "كتاب ابيض" يتضمن من بين القضايا موقف الحكومة من البقاء في السوق المشتركة او الانسحاب منها، وهي واحده من المسائل الاساسية في عملية خروج بريطانيا من الاتحاد ودعت اللجنة الحكومة الى التفاوض حول "مرحلة تكيف" بدلا من قطيعة فورية مع الاتحاد الاوروبي، في حال.الاوروبي ورحب ناطق باسم الحكومة بالتقرير مؤكدا "ننوي تقديم ."تغير شروط دخول الشركات البريطانية الى السوق الاوروبية خططنا شرط الا تضر بموقف بريطانيا في المفاوضات، بحلول نهاية آذار"، وهو الموعد الذي حددته رئيسة الحكومة المحافظة تيريزا ماي لتفعيل الماده 50 من معاهده لشبونة

Figure 3. Sample of Jordanian articles about Brexit.

3.2. Data Preprocessing

Data preprocessing involves cleaning the data from unnecessary and unhelpful information to obtain better results. Preprocessing functions aim to minimize information loss while maintaining maximum data dimensionality. When one works with Arabic text, preprocessing is an essential step in data classification due to the unique Arabic morphology, such as diacritics and dots, which are commonly misplaced or unmarked, causing word ambiguity. Normalization, tokenization, and stop-word removal were applied to both datasets to address the previous challenges.

- Tokenization is splitting a sentence into a set of tokens (words), which is important because unstructured text can be converted into independent words that can be easily analyzed. The dataset was tokenized using Keras' utility tokenizer class, automatically removing all punctuation and tokenizing the text.
- Normalization is undertaken to overcome misplaced dots or glitches in a word. This study normalized by replacing letters, as shown in Figure 4.
- Removal of stop-words, non-Arabic characters, and punctuation

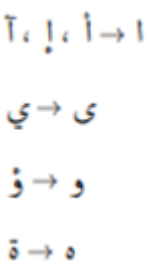


Figure 4. Example of normalization by replacing letters.

4. Model Compilation Using Machine Learning Methods

In this section, we explain the details regarding the training of numerous traditional machine-learning algorithms with n-gram models for classifying news articles based on the country of origin.

4.1. Feature Extraction

Language modeling and natural language processing areas frequently use feature identification and analysis methods. Based on the clean dataset, we applied a widely used frequency distribution approach referred to as n-grams.

N-grams are contiguous sequences of n items from text or speech. They are used in natural language processing for language modeling, text classification, and information retrieval. Bigrams (n-gram = 2) and trigrams (n-gram = 3) are the most commonly studied. n-grams have been found to improve language models and text classification [13].

They help in identifying key phrases and patterns in text corpora. However, their effectiveness depends on the dataset size and complexity, n-gram size choice, and the use of other techniques, such as neural networks and deep learning. Despite some challenges, n-grams remain a powerful tool in natural language processing and have various applications [14].

4.2. Feature Representation

After extracting the n-grams, each article was represented as a feature vector of vocabulary. Term frequency-inverse document frequency (TF-IDF) is used in natural language processing to measure the importance of terms in documents. It calculates the term frequency and inverse document frequency to assign weights to each term, which improves text classification and information retrieval accuracy. TF-IDF measures term importance in a document or corpus using term frequency and inverse document frequency. The resulting scores can be used to represent documents as numerical vectors for text analysis purposes, such as information retrieval, text classification, and sentiment analysis [14]. Hence, TF-IDF is widely used and effective for various natural language processing tasks. Researchers have explored modifications and extensions to improve its effectiveness in specific applications. TF-IDF has been used in combination with other techniques, such as deep learning and word embedding for higher accuracy in text analysis tasks [15]. The formula for TF-IDF is:

$$\text{TF-IDF} = \text{Term Frequency} \times \text{Inverse Document Frequency} \quad (1)$$

The combination of extracting features using n-gram and representing them using TF-IDF is a powerful tool for text classification and can be used to accurately classify texts into predefined categories. The n-grams are used to extract features from the text documents and the TF-IDF is used to weigh the importance of the extracted features. In this regard, we combine these two techniques to allow for effective text classification by taking into account both the occurrence of words and the relative importance of those words.

4.3. Classifiers

Below is a brief description of the machine learning classifiers used:

1. **LINEAR support vector machine (SVC):** Also known as a support vector classifier (SVC), in which each data point is represented as a point in space, and samples from different categories are widely separated. A new sample is first mapped to the space when it arrives. Depending upon which side of the dividing line the new data sample point is on, the new sample is assigned to a group. In the case of binary classification, the classification gap of the SVM can be viewed as a hyperplane. If there are more than two classes, the dividing gap can be seen as a collection of hyperplanes positioned in a high-dimensional region. The optimal hyperplane is chosen to have the maximum possible distance from the nearest sample on each side of the separating hyperplane.
2. **Multinomial Naive Bayes/Bernoulli Naive Bayes:** Naive Bayes is a straightforward learning algorithm that uses the Bayes rule and the basic assumption that, given the class, the attributes are conditionally independent. Even though, in practice, this independence assumption is frequently broken, Naive Bayes frequently yields competitive classification accuracy. Its computational effectiveness and numerous other appealing characteristics contribute to Naive Bayes's widespread use in practice. The multinomial model determines the term frequency or how frequently a term appears in a document. This model's characteristic makes it a viable option for document categorization, given that a phrase could be crucial in determining the sentiment of the content. Additionally, term frequency can be used to determine whether a term will benefit the analysis. In the Bernoulli model, features are independent binary variables that indicate whether a term appears in the document under review. This algorithm, somewhat similar to the multinomial model in the classification process, is also a common method for text classification tasks.
3. **Standard gradient descent (SGD) classifier:** We utilized an assortment of training samples drawn from the probability distribution P (x, y) in supervised learning problems. The association between the input vector x and output label y that we attempted to estimate is represented by the conditional probability P(y | x). A loss function illustrates the discrepancy between the estimated label y and the actual label y. We attempted to quantify the function f that reduces the expected risk. The function f is linearly parameterized by w. The actual gradient of the w vector, commonly approximated as the sum of the gradients induced by each unique training sample, is utilized in standard GD approaches to reduce the empirical risk. SGD "stochastic gradient descent" considers one sample at each iteration and updates the weight vector w iteratively using a time-dependent weighting factor.
4. **Decision trees:** These categorize the objects in the dataset by asking questions about the features connected to the data items. Every internal node points to one child node for each potential response to its question and each question is contained within a node. The questions are then encoded as a tree, creating a hierarchy. In its most basic version, yes-or-no questions are asked, with a "yes" child and a "no" child for each internal node. According to the responses to the item in question, an item is sorted into a class by following the path from the highest node, the root, to a node without children, the leaf. When an object reaches a leaf, it is assigned to the class linked to that leaf. In some instances, every leaf has a probability distribution across the classes to estimate the conditional likelihood that an item will belong to a particular class if it reaches the leaf; however, unbiased probability estimation might be challenging.
5. **Random forest (RF):** This is a common and effective ensemble supervised classification technique. RF has been successfully applied to many machine learning applications, including many in bioinformatics and medical imaging because of its superior accuracy and robustness, and its ability to give insights by ranking its features. The decision trees that make up RF are all generated by the bagging algorithm without pruning, resulting in a "forest" of classifiers that vote for a specific class. A training database with ground-truth class labels is required, along with two parameters: the number of trees in the forest (ntree) and the number of randomly chosen features/variables used to assess each tree node (mtry). The voting threshold or cut-

off (the proportion of trees in the forest required to vote for a particular class), which is used to calculate recall, precision, and the F-score, can also be changed using RF.

6. **K-nearest neighbors (KNN):** This is one of the easiest and most straightforward supervised machine learning techniques. Because the training examples must be kept in memory during run-time, the technique is known as memory-based classification. Without making prior assumptions about the distributions from which the training examples are taken, KNN consistently performs well. It involves a training set of both positive and negative cases. By measuring the distance to the closest training instance, a fresh sample is classified; the classification of the sample is then based on the sign of that point. By selecting the k closest points and designating the majority sign, the KNN classifier expands on this concept. It is customary to use k small and odd numbers to break ties, usually 1, 3, or 5. Larger k values decrease the impact of noisy points in the training dataset, and cross-validation is frequently used to select K.

4.4. Experimental Setup

Both the Hajj and Brexit datasets were divided into 60% training, 20% validation, and 20% testing. The following tables display the results obtained by our tested models using three different values for the n-gram (1, 2, and 3). The eight ML models used the default parameters available in Scikit Learn (https://scikit-learn.org/stable/supervised_learning.html#supervised-learning (accessed on 1 June 2023)).

4.5. Results and Discussion

We tested different widely known supervised machine learning techniques to classify the articles based on the country of origin for both the Hajj and Brexit datasets. The eight models used were linear SVC, SVM, multinomial NB, Bernoulli NB, SGD classifier, decision tree, random forest, and KNN, testing different n-gram values with each. The evaluation of the model's performance is presented in the next Tables 4 and 5 based on the accuracy of each model and other performance metrics, such as precision, recall, and the F-score, where the best results were marked in bold.

Based on the presented results in Table 4, we can see that the linear SVC with n-gram = 1 yielded the best performance in terms of all the considered measures, with an accuracy of up to 93%. This allowed for a robust comparison of our method against the Al-Barhamtoshy et al. [12] model, which reached an 85% F-score when the authors trained their model using textual features (polarity, emotions, linguistics, and part of speech) with the bag-of-words approach using a similar ML algorithm. Perhaps more meaningful is a comparison of the training method features in both models. In their study, the application of the Al-Barhamtoshy et al. [12] model resulted in an F-score of 49% in SVM and 83% in RF when their features were trained using the BoW approach. However, the model achieved the highest F-score of 93% in SVM and a low F-score of 21% in RF when training the model using n-grams, specifically n-gram = 1. This purely reflected the effect the training features had on the model's performance using the same ML algorithms. In fact, Al-Barhamtoshy et al. [12] demonstrated that the textual features achieved an F-score of 69% using RF, and, when combined with BoW, the score jumped to 85%. This observation obviously implies the effect of n-grams, which we clearly see in this study, as they achieved high scores of 93% in linear SVM.

Furthermore, as shown in Table 4 for the Hajj articles dataset, the change in the n-gram values did not have a significant impact on the performance of the tested models, except for Bernoulli Naive Bayes. We can see from the table that the smaller value for n-gram = 1 was better with the Bernoulli NB as the accuracy achieved with unigram was 87%; with larger values, this accuracy decreased to 81.1% and 72.5% when the n-gram values changed to 2 and 3, respectively.

Table 4. The performance of different machine learning classifiers for the Hajj articles dataset.

	N-Gram	Accuracy	Precision	Recall	F1-Score
LinearSVC	1	**93.10%**	**93.10%**	**93.10%**	**93.00%**
SVM	1	91.40%	91.40%	91.40%	91.30%
MultinomialNB	1	89.50%	90.60%	89.50%	89.50%
BernoulliNB	1	87.30%	88.60%	87.30%	87.50%
SGDClassifier	1	92.80%	92.80%	92.80%	92.80%
DecisionTreeClassifier	1	74.40%	78.20%	74.40%	74.50%
RandomForestClassifier	1	35.20%	78.30%	35.20%	21.40%
KNeighborsClassifier KNN	1	85.20%	86.20%	85.20%	85.10%
LinearSVC	2	**91.40%**	**91.50%**	**91.40%**	**91.40%**
SVM	2	91.40%	91.50%	91.40%	91.30%
MultinomialNB	2	89.70%	91.30%	89.70%	89.80%
BernoulliNB	2	81.10%	86.40%	81.10%	81.50%
SGDClassifier	2	91.90%	92.00%	91.90%	91.80%
DecisionTreeClassifier	2	74.90%	79.90%	74.90%	75.00%
RandomForestClassifier	2	32.50%	70.20%	32.50%	17.30%
KNeighborsClassifier KNN	2	84.70%	86.00%	84.70%	84.70%
LinearSVC	3	**91.40%**	**91.60%**	**91.40%**	**91.40%**
SVM	3	**91.10%**	**91.40%**	**91.10%**	**91.10%**
MultinomialNB	3	89.20%	90.90%	89.20%	89.30%
BernoulliNB	3	72.50%	82.50%	72.50%	72.80%
SGDClassifier	3	91.60%	91.90%	91.60%	91.60%
DecisionTreeClassifier	3	74.60%	79.60%	74.60%	74.80%
RandomForestClassifier	3	35.90%	75.80%	35.90%	22.70%
KNeighborsClassifier KNN	3	85.40%	86.60%	85.40%	85.30%

Table 5. The performance of different machine learning classifiers for the Brexit dataset.

	N-Gram	Accuracy	Precision	Recall	F1-Score
LinearSVC	1	**78.60%**	**80.10%**	**78.60%**	**79.00%**
SVC	1	77.20%	80.00%	77.20%	77.90%
MultinomialNB	1	74.80%	77.20%	74.80%	74.20%
BernoulliNB	1	62.90%	71.80%	62.90%	61.80%
SGDClassifier	1	78.20%	79.20%	78.20%	78.60%
DecisionTree	1	73.80%	76.40%	73.80%	74.30%
RandomForest	1	36.10%	41.10%	36.10%	28.20%
KNeighbors	1	65.00%	65.40%	65.00%	64.00%
LinearSVC	2	**82.00%**	**82.60%**	**82.00%**	**82.20%**
SVC	2	77.20%	81.20%	77.20%	77.90%
MultinomialNB	2	74.10%	76.00%	74.10%	73.40%
BernoulliNB	2	53.70%	70.50%	53.70%	51.00%
SGDClassifier	2	**81.00%**	**81.90%**	**81.00%**	**81.30%**
DecisionTree	2	77.20%	78.80%	77.20%	77.50%
RandomForest	2	31.60%	44.00%	31.60%	16.10%
KNeighbors	2	64.60%	65.10%	64.60%	63.60%
LinearSVC	3	**83.00%**	**83.70%**	**83.00%**	**83.20%**
SVC	3	75.20%	80.10%	75.20%	75.90%
MultinomialNB	3	74.10%	75.50%	74.10%	73.60%
BernoulliNB	3	47.60%	68.10%	47.60%	42.00%
SGDClassifier	3	**82.30%**	**82.90%**	**82.30%**	**82.50%**
DecisionTree	3	75.50%	77.70%	75.50%	75.80%
RandomForest	3	31.60%	78.70%	31.60%	16.10%
KNeighbors	3	63.60%	63.90%	63.60%	62.50%

Moreover, the SGD classifier and SVM with any value for n-gram provided comparable high performance according to model accuracy, which was around 92% and 91% for SGD and SVM, respectively. Furthermore, despite a decrease, the results for the Brexit dataset in

Table 5 confirmed the previous results, as both linear SVC and SGD classifiers achieved the best performance across all the experimental models. However, the best accuracy was achieved at n-gram = 3.

On the other hand, the random forest classifier achieved the lowest accuracy for both datasets among all the tested models. Unlike the other models, whose performance decreased for the Brexit dataset, random forest achieved almost the same performance, where its accuracy was between 31 and 36% with the different n-gram values for both datasets.

5. Model Compilation Using Deep Learning Methods

In this section, we describe the deep learning models with different embedding techniques.

5.1. Features Extraction

For the data preparation, all sentences were converted into numerical sequences. The maximum number of words in each sentence and the number of unique tokens in the corpus were calculated. Moreover, sentences were padded to a fixed length, meaning the maximum sentence length. Then, a one-hot-encode was applied to the labels. As for the textual features, the word embedding was used to represent the textual input. Both baseline embedding and pre-trained embedding were used to build an embedding layer as an input layer for the deep neural architectures. In random embedding, the word vectors were learned while training the network, while in pre-trained embedding, the word vectors were generated using Arabic Wordvec, FastText, and mBERT.

5.2. Classifiers

Unlike machine learning models, deep learning networks do not directly perform feature extraction and classification. Without involving a third-party researcher, the deep learning network's hidden layers tacitly carry out all of these tasks. Below is a brief explanation of the deep learning networks used.

1. **Convolutional neural network (CNN):** This is an improvised variant of the multilayer perceptron. A CNN typically consists of an input layer, an output layer, and numerous concealed layers. Convolutional, pooling, and fully connected layers usually comprise a CNN's hidden layers.
2. **Long short-term memory (LSTM):** The units are a unique type of recurrent neural network (RNN) construction block. Long short-term memory refers to the fact that it is capable of holding information for a long period of time. An LSTM network can analyze, categorize, and predict temporal data patterns with time lags of any length, and typically consists of memory, input, output, and ignore gates. The memory in an LSTM network can retain numbers for any duration of time. The three gates are variations of neurons that calculate the activation function of a weighted sum. They are referred to as gates because they regulate the flow of values through LSTM levels. The exploding and vanishing gradient problem is a crucial challenge that LSTM addresses.
3. **Deep bidirectional LSTMs (BiLSTM):** These are an extension of LSTM models in which two LSTMs are applied to the input data. An LSTM is used on the input sequence in the first round (i.e., forward layer). In the next round, the LSTM model is given the input sequence's reverse form (i.e., backward layer). By using the LSTM twice, long-term dependencies become clearer, which enhances the model's accuracy.
4. **Gated recurrent unit (GRU:** This is another type of recurrent neural network that is similar to LSTM but has fewer parameters. GRU has been shown to be effective for sequence classification tasks, especially in cases where long-term dependencies are not as important. In our project, we used GRU to classify the Arabic text data based on the features generated by the different feature extraction techniques.
5. **Hybrid networks (CNN-LSTM)/ (CNN-BiLSTM) / (CNN+GRU):** In hybrid networks, only the convolutional layer is applied as a feature extraction layer with max-pooling to generate output for the next layer (LSTM, BiLSTM, or GRU).

5.3. Experimental Setup

For training the deep learning models, an Adam optimizer with a 0.01 learning rate, weight decay of 0.0005, and 128 batch size was used. A dropout value of 0.5 was used to avoid overfitting and to hasten the learning. The output layer used a softmax activation function. The experiments utilized Python programming Tensorflow and Keras libraries for the machine learning and deep learning models. For hardware, a Windows 10–based machine with a core i7 CPU and 16 GB RAM was used.

Our experiments included two stages. First, we investigated the state-of-the-art deep learning models for Arabic text, namely CNN, LSTM, BiLSTM, and GRU, using different embedding techniques. The second stage included selecting the models that achieved the highest performance in order to build a hybrid model that added a CNN layer for feature extraction and to assess its impact on the models' performance. Below, we briefly explain the process of building each model.

1. **CNN**
   The implemented architecture was, therefore, comprised of the following key pieces (as shown in Figure 5):
   - Word encoding layer: Each sentence consists of a sequence of words $s_i = \{w_{i1}, w_{i2}, \ldots, w_{ik}, \ldots, w_{in}\}$, where w_{ik} denotes the kth word in the ith sentence and n represents the length of the sentence. Each word in the sentence is represented as a vector, known as the word embedding. An embedding matrix is constructed to form the first layer of the model.
   - Convolutional layer: This layer learns to extract meaningful features and substructures from the text for the overall prediction task at hand. We implemented a convolutional neural network (CNN) with a linear ('relu') activation function, followed by a pooling layer to reduce the output of the convolutional layer by 20%. Next, the output was flattened to one long 2D vector to represent the features extracted by the CNN. Finally, we added dense layers, including scale and rotating by transforming the vector by multiplying the matrix and vector. A dropout of 0.2 was applied to reduce the model's overfitting. Finally, the output layer used a sigmoid activation function.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 500)]	0
embedding (Embedding)	(None, 500, 300)	15244200
conv1d (Conv1D)	(None, 496, 128)	192128
max_pooling1d (MaxPooling1D)	(None, 99, 128)	0
flatten (Flatten)	(None, 12672)	0
dropout (Dropout)	(None, 12672)	0
dense (Dense)	(None, 180)	2281140
dropout_1 (Dropout)	(None, 180)	0
dense_1 (Dense)	(None, 3)	543

Total params: 17,718,011
Trainable params: 17,718,011
Non-trainable params: 0

Figure 5. The CNN network architecture.

2. **LSTM**
   The architecture is shown in Figure 6; the first layer was the word encoding layer. The next layer was the LSTM layer, with 128 memory units. The activation function was softmax, and categorical cross-entropy was used as the loss function.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 500, 300)          15244200

 lstm (LSTM)                 (None, 300)               721200

 dense_2 (Dense)             (None, 300)               90300

 dense_3 (Dense)             (None, 300)               90300

 dense_4 (Dense)             (None, 3)                 903

=================================================================
Total params: 16,146,903
Trainable params: 16,146,903
Non-trainable params: 0
_________________________________________________________________
```

Figure 6. The LSTM network architecture.

3. **BiLSTM**

   The architecture is shown in Figure 7. The word encoding layer was the first layer of the model, then a bidirectional LSTM layer was added. Three dense layers were then added to scale, rotate and transform the vectors and the softmax activation function was used to generate the final classification.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 500, 300)          15244200

 bidirectional (Bidirectiona  (None, 600)              1442400
 l)

 dense_6 (Dense)             (None, 300)               180300

 dense_7 (Dense)             (None, 3)                 903

=================================================================
Total params: 16,867,803
Trainable params: 16,867,803
Non-trainable params: 0
_________________________________________________________________
```

Figure 7. The BiLSTM network architecture.

4. **GRU**

   In GRU, the embedding layer was the first layer in the network fed into a GRU layer with 128 units, which had a dropout rate of 0.2 to prevent overfitting. A dense layer with a softmax activation function was added to output the classification probabilities for the three classes. The model was compiled with a categorical cross-entropy loss function Adam optimizer, as shown in Figure 8.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_10 (Embedding)    (None, 500, 300)          15000000

 gru_7 (GRU)                 (None, 128)               165120

 dense_16 (Dense)            (None, 3)                 387

=================================================================
Total params: 15,165,507
Trainable params: 15,165,507
Non-trainable params: 0
_________________________________________________________________
```

Figure 8. The GRU network architecture.

5. **CNN+LSTM**
   CNNs were used as feature extractors for the LSTMs on input data, as shown in Figure 9. The first layer was the word encoding layer, then a conventional layer was added, followed by a max-pooling layer and a drop-out layer to help prevent the network from overfitting. An LSTM layer with a hidden size of 128 was added and the softmax activation function was used to generate the final classification, as shown in Figure 9.

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 500)]	0
embedding (Embedding)	(None, 500, 300)	15244200
conv1d_1 (Conv1D)	(None, 496, 64)	96064
max_pooling1d_1 (MaxPooling 1D)	(None, 99, 64)	0
dropout_2 (Dropout)	(None, 99, 64)	0
conv1d_2 (Conv1D)	(None, 95, 64)	20544
max_pooling1d_2 (MaxPooling 1D)	(None, 19, 64)	0
dropout_3 (Dropout)	(None, 19, 64)	0
lstm_1 (LSTM)	(None, 32)	12416
dense_5 (Dense)	(None, 3)	99

Total params: 15,373,323
Trainable params: 15,373,323
Non-trainable params: 0

Figure 9. The CNN+LSTM network architecture.

6. **CNN+BiLSTM**
   In this architecture, the word encoding layer was the first layer. Conventional layers were added followed by a max-pooling and a dropout of 0.2 to help prevent overfitting. Afterward, a bidirectional LSTM layer was added. Finally, a dense layer with a softmax activation function was used to generate the final classification, as shown in Figure 10.

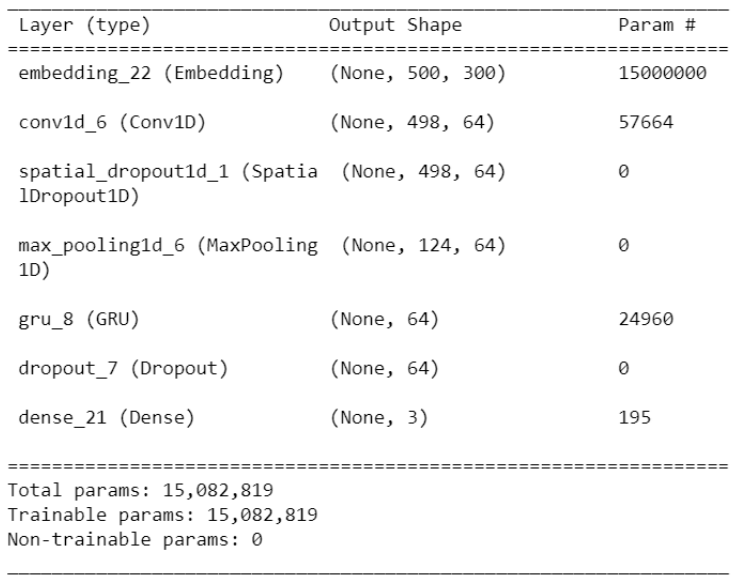
Layer (type)	Output Shape	Param #
input_6 (InputLayer)	[(None, 500)]	0
embedding (Embedding)	(None, 500, 300)	15244200
conv1d_3 (Conv1D)	(None, 496, 64)	96064
max_pooling1d_3 (MaxPooling 1D)	(None, 99, 64)	0
dropout_4 (Dropout)	(None, 99, 64)	0
conv1d_4 (Conv1D)	(None, 95, 64)	20544
max_pooling1d_4 (MaxPooling 1D)	(None, 19, 64)	0
dropout_5 (Dropout)	(None, 19, 64)	0
bidirectional_1 (Bidirectio nal)	(None, 64)	24832
dense_8 (Dense)	(None, 3)	195

Total params: 15,385,835
Trainable params: 15,385,835
Non-trainable params: 0

Figure 10. The CNN+BiLSTM network architecture.

7. **CNN+GRU:**
   In this architecture, a fully connected conventional layer was added after the word encoding layer followed by a max-pooling and a dropout of 0.2 to help prevent overfitting. The output of the conventional layer was then fed to the GRU layer with

128 units, which had a dropout rate of 0.2 to prevent overfitting. A dense layer with a softmax activation function was used to generate the final classification, as shown in Figure 11.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_22 (Embedding)     (None, 500, 300)          15000000

conv1d_6 (Conv1D)            (None, 498, 64)           57664

spatial_dropout1d_1 (Spatia  (None, 498, 64)           0
lDropout1D)

max_pooling1d_6 (MaxPooling  (None, 124, 64)           0
1D)

gru_8 (GRU)                  (None, 64)                24960

dropout_7 (Dropout)          (None, 64)                0

dense_21 (Dense)             (None, 3)                 195

=================================================================
Total params: 15,082,819
Trainable params: 15,082,819
Non-trainable params: 0
_________________________________________________________________
```

Figure 11. The CNN+GRU network architecture.

5.4. Results and Discussion

5.4.1. Impact of Different Models

As explained above, we tested four different deep learning models, namely CNN, LSTM, BiLSTM, and GRU across two datasets. Our experiments used some of the most prevalent word-embedding techniques (i.e., random embedding, pre-trained Word2vec, and FastText). Tables 6 and 7 show the performance of the tested models according to precision, recall, and the F-score as performance metrics for both Hajj and Brexit datasets; while the accuracy of the tested models is presented in Tables 8 and 9. In all tables, we marked the best results in bold.

For the Hajj dataset, the results showed that, in general, CNN performed the best among the tested models when using random embedding or FastText, with an F-score of 0.91 and 0.92, respectively. However, using the pre-trained Word2vec, we noticed that BiLSTM yielded the best results, with an accuracy of up to 86.6%. Moreover, it was observed that CNN and GRU achieved their lowest precision, recall, and F1 using pre-trained Word2vec with approximately 11% and 9% lower accuracy, respectively. This may have been because, among the 50,814 words in the trained corpus, 18,107 words in the data were not found in the Word2vec vocabulary.

Additionally, BiLSTM achieved the same performance for both Word2vec and FastText, which was 2% lower than the random embeddings. Finally, LSTM achieved its highest accuracy using Word2vec.

Table 6. The performance of the basic tested deep learning models using different embedding techniques for the Hajj articles dataset.

Embedding	**CNN**			**LSTM**			**BiLSTM**			**GRU**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	F1	**P**	**R**	**F1**
Random	0.92	0.92	**0.91**	0.79	0.82	0.80	0.89	0.89	0.89	0.84	0.83	0.83
Word2vec	0.81	0.81	0.81	0.81	0.81	0.81	0.87	0.87	**0.87**	0.74	0.73	0.74
FastText	0.92	0.92	**0.92**	0.80	0.76	0.76	0.87	0.87	0.87	0.84	0.83	0.83

The performance scores achieved by the models for the Brexit dataset presented in Table 7 confirmed the previous results. CNN outperformed the other models using random embedding and FastText with F-scores of 0.75 and 0.77 compared to 0.58 using the pre-trained Word2vec embedding. whereas, BiLSTM had the best scores using Word2vec. Additionally, both CNN and GRU maintained their worst scores using pre-trained Word2vec, with approximately 9% and 18% lower accuracy, respectively.

Table 7. The performance of the basic tested deep learning models using different embedding techniques for the Brexit dataset.

Embedding	CNN			LSTM			BiLSTM			GRU		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Random	0.75	0.75	**0.75**	0.62	0.62	0.62	0.60	0.60	0.60	0.65	0.65	0.65
Word2vec	0.58	0.58	0.58	0.55	0.55	0.55	0.64	0.64	**0.64**	0.48	0.48	0.48
FastText	0.77	0.77	**0.77**	0.65	0.65	0.65	0.64	0.64	0.64	0.64	0.64	0.64

Table 8. Accuracy of the basic tested deep learning models for the Hajj articles dataset.

Embedding Model	Accuracy			
	CNN	LSTM	BiLSTM	GRU
Random Embedding	**91.60%**	79.00%	88.80%	82.78%
Word2vec	80.86%	80.62%	**86.60%**	73.68%
FastText	**92.10%**	76.08%	86.84%	83.49%

Table 9. Accuracy of the basic tested deep learning models for the Brexit dataset.

Embedding Model	Accuracy			
	CNN	LSTM	BiLSTM	GRU
Random Embedding	**74.82%**	61.56%	59.86%	64.63%
Word2vec	57.82%	55.10%	**64.28%**	47.96%
FastText	**76.87%**	64.62%	63.94%	63.61%

5.4.2. Impact of Hybrid Models

We extended our experiment to build hybrid network structures, including CNN with LSTM, CNN with BiLSTM, and CNN with GRU. CNN was chosen due to its superiority over the other models. Moreover, hybrid models have widely used a feature extraction layer. The performance of the hybrid models is shown in the tables below for the two tested datasets, relating to the application of performance metrics for each model that utilized different embedding techniques where we marked the best results in bold.

The results of the hybrid models for the Hajj dataset are presented in Table 10. As we can see that CNN+LSTM and CNN+BiLSTM performed similarly with respect to precision, recall, and the F-measure, with a slight difference in the performance of the two structures in terms of accuracy. Referring to Table 11, the accuracy of CNN+BiLSTM for the Hajj dataset was higher than for all the other models using random embedding and Word2vec, whereas CNN+LSTM performed better with FastText.

We observed that the performance of the three basic models was enhanced when combined with CNN. For instance, the accuracy of the three models improved from 79% to 94.4% for LSTM, from 88.8% to 93.54% for BiLSTM, and from 82.78% to 88.04% for GRU using the random embedding technique. Although the GRU performance improved when combined with CNN, all the metrics showed that the hybrid CNN+GRU yielded the lowest performance among the three hybrid models. It is worth mentioning that the performance of CNN+GRU improved significantly when using FastText embedding compared to the other embedding techniques. Furthermore, Word2vec achieved the lowest performance

for all the experimental models with approximately 8–11% difference between random embedding and FastText.

Table 10. The performance of the hybrid deep learning models using different embedding techniques for the Hajj articles dataset.

Embedding Technique	CNN+LSTM			CNN+BiLSTM			CNN+GRU		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Random Embedding	0.94	0.94	**0.94**	0.94	0.94	**0.94**	0.88	0.88	0.88
Word2vec	0.84	0.84	0.84	0.86	0.86	**0.86**	0.79	0.79	0.79
FastText	0.93	0.93	**0.93**	0.92	0.92	0.92	0.91	0.91	0.91

Table 11. Accuracy of the hybrid deep learning models for the Hajj articles dataset.

Embedding Model	Accuracy		
	CNN+LSTM	**CNN+BiLSTM**	**CNN+GRU**
Random Embedding	**94.4%**	93.54%	88.04%
Word2vec	84.21%	**86.36%**	77.75%
FastText	**93.06%**	92.34%	91.15%

For the Brexit dataset, as shown in Table 12, the results confirmed that adding a CNN layer enhanced the performance scores achieved by the deep learning models. The accuracy score of LSTM+CNN increased by approximately 3%, 12%, and 11% using random embedding, Word2vec, and FastText, respectively. Likewise, the accuracy of GRU+CNN increased by approximately 5%, 7%, and 3% for the three embedding techniques. On the other hand, BiLSTM+CNN achieved higher accuracy using both random embedding and FastText, but its accuracy decreased by 2% using Word2vec embedding. Furthermore, as shown in Table 13, the accuracy of CNN+LSTM for the Brexit dataset was marginally better than CNN+BiLSTM using random embedding and FastText, while CNN+BiLSTM performed better with the pre-trained Word2vec with an accuracy of 86.36% compared to 84.21% for CNN+LSTM.

Table 12. The performance of the hybrid deep learning models using different embedding techniques for the Brexit dataset.

Embedding Technique	CNN+LSTM			CNN+BiLSTM			CNN+GRU		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Random Embedding	0.73	0.73	0.73	0.75	0.75	**0.75**	0.70	0.70	0.70
Word2vec	0.59	0.59	0.59	0.63	0.63	**0.63**	0.54	0.54	0.54
FastText	0.76	0.76	**0.76**	0.71	0.71	0.71	0.66	0.66	0.66

Table 13. Accuracy of the hybrid deep learning models for the Brexit dataset.

Embedding Model	Accuracy		
	CNN+LSTM	**CNN+BiLSTM**	**CNN+GRU**
Random Embedding	73.46%	**74.83%**	70.07%
Word2vec	58.84%	**62.59%**	53.74%
FastText	**75.51%**	70.75%	66.33%

5.4.3. Error Analysis

Here, we analyze the misclassified articles in the country-of-origin classification model for each approach used: ML or DL. We provide examples of the misclassified articles

(with their English translation) and propose reasons for the errors. For the ML model, we found that the linear SVC with n-gram = 1 in the Hajj dataset and n-gram = 3 with the same ML algorithm in the Brexit dataset resulted in a lower error rate. According to the experiments' confusion matrix pertaining to the Hajj dataset, we found that 138 out of 152 of the Egyptian articles in the testing set were correctly classified by the classifiers, while 10 were classified as Saudi and four as Jordanian. Moreover, 126 out of the 128 Saudi articles were classified correctly, with one misclassified as Jordanian and another as Egyptian. Likewise, 126 Jordanian articles were correctly classified, with six misclassified as Egyptian and five as Saudi (see Figure 12). On the other hand, for the Brexit dataset, more misclassified articles under the same classifier were noted. Specifically, 11 and 19 Egyptian articles were mistaken as being of Saudi and Jordanian origin, respectively, Figure 13.

Similarly, random embedding with the hybrid model of CNN+LSTM classified 138 Egyptian articles correctly and misclassified two articles. There was a noticeable difference between the results for misclassified Saudi and Jordanian articles, with 13 and 12 incorrect articles for both KSA and Jordan, respectively, as seen in Figure 14. These insights might indicate that training the model on smaller n-grams involved taking the text one unit at a time, compared to higher n-grams, which take more than one unit at a time. This finding was also found in studies for text classification [16]. However, these insights may not be generalizable, as we found, in contrast, that n-gram = 3 was more influential for the Brexit dataset, compared to the unigram (n-gram = 1) for the Hajj dataset. This was not the case in the Brexit dataset; we found that the number of misclassified Jordanian articles as Saudi articles were relatively high—34 out of 91 (see Figure 15).

A possible explanation is that most of the articles on the topic of Hajj were distinctive between all three Arabic countries, Saudi Arabia, Egypt, and Jordan because the topic had a direct impact on them. This impact created a link between unique words that formulated the news articles for each country. For example, most Saudi articles presented senior political figures with their full titles, such as "His Royal Highness Prince Khaled AlFaisal"; however, some Jordanian articles would indicate him as "Prince Khaled AlFaisal". So, words such as "highness" would be linked mostly to articles originating from Saudi Arabia. However, since the topic of Brexit did not have the same impact on Arabic articles as the topic of Hajj did, and most of the articles described issues concerning foreign events related to the topic, unique words linking them to the three originating countries were not used. This would explain why the model needed to be trained on higher n-grams than 1—in this case, n-gram = 3.

When further analyzing the misclassified articles, we observed that the most common feature of the false negative articles was the standard creation of journalistic terminologies written in modern standard Arabic (MSA). Because all news articles are written in a formal journalistic format, there was a need to analyze the text on a word level, assigning specific words to specific nations. In support of this idea, samples of the misclassified articles showed that most discussed topics related to other nations, including some of the misclassified Egyptian articles presenting a topic that happened in Saudi Arabia. Figure 16 shows examples of this.

5.5. Limitations

The country-based classification of news articles has limitations that must be considered. First, it is difficult to accurately classify news articles based on their country of origin when the articles' topic has no direct impact on the country. This is relevant as Arabic news articles are written in modern Arabic and use specific journalistic registers that result in nuanced differences between articles originating from one country or another. With the prevalence of international news outlets and the ever-growing interconnections of the world, news articles can often contain information from multiple countries. Thus, accurately assigning a single country to a given news article can be a difficult task. Second, as there are Arabic news datasets available, they originate from one country. This is important for a multi-class classification task. Thus, there is still a lack of datasets that provide

a variety of news articles from various Arabic countries. Finally, the testing and evaluation of our models were conducted without stemming the articles. We did not perform stemming because we wanted to preserve the authenticity of the articles to improve our comprehension of the model's performance.

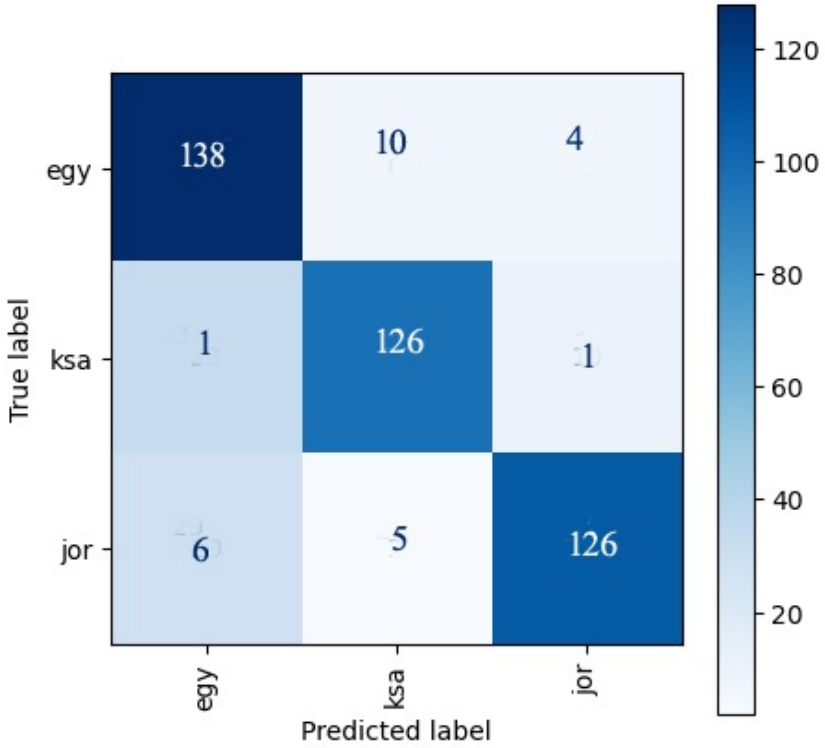


Figure 12. The confusion matrix using SVM classifier with Unigram for the Hajj articles dataset.

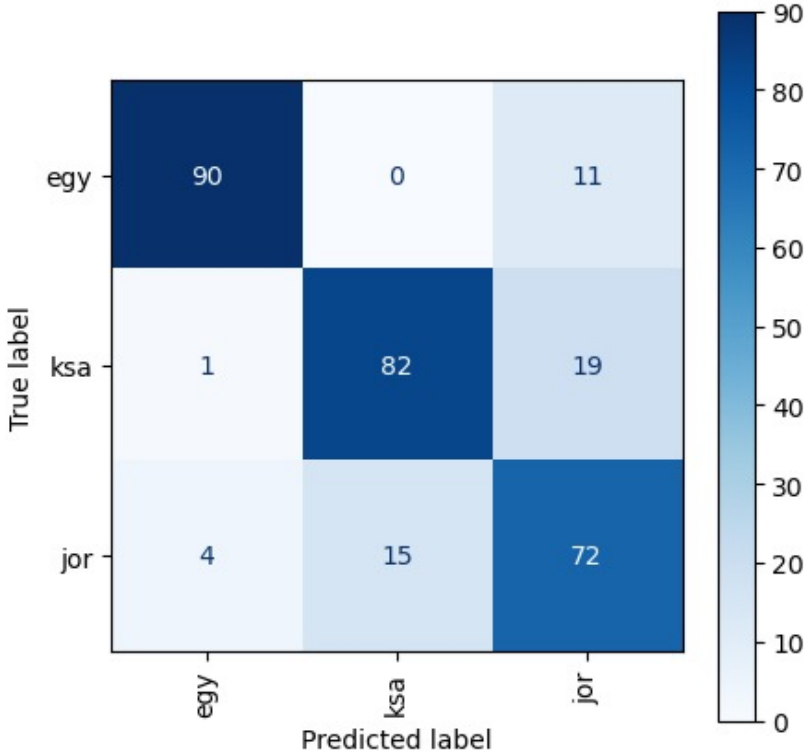


Figure 13. The confusion matrix using SVM classifier with 3-gram for the Brexit articles dataset.

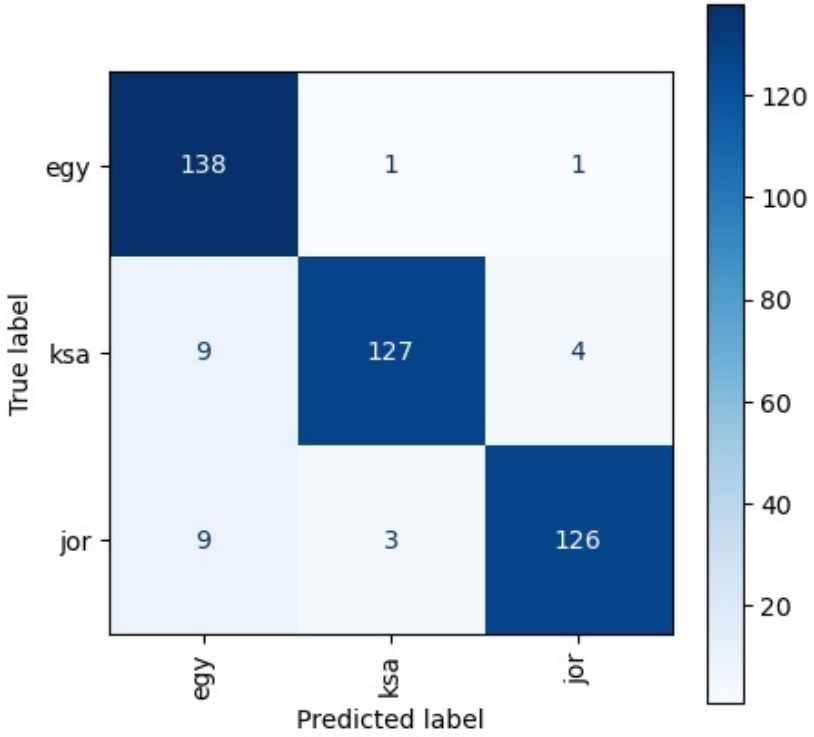


Figure 14. The confusion matrix using CNN+LSTM classifier with random embedding for the Hajj articles dataset.

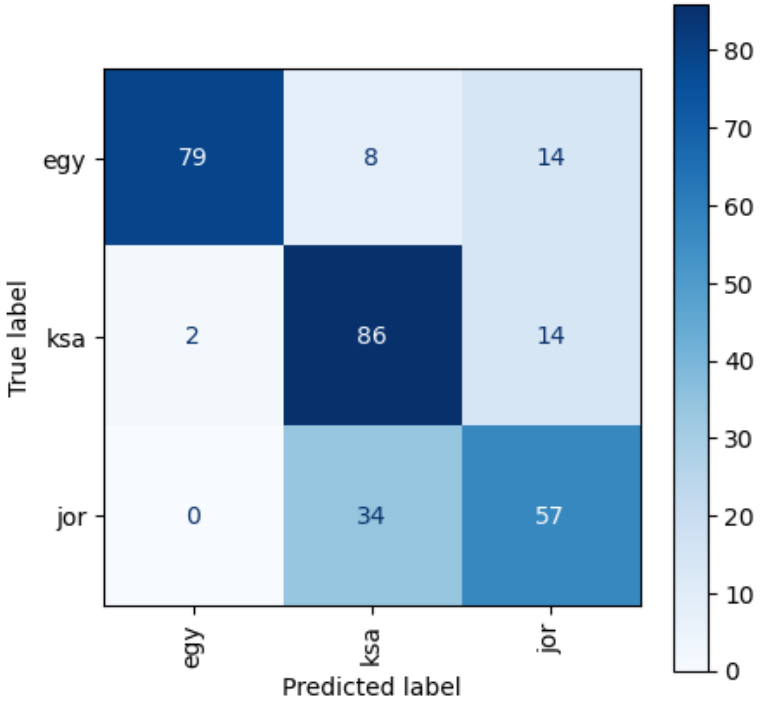


Figure 15. The confusion matrix using CNN+LSTM classifier with FastText for the Brexit articles dataset.

أعلن وزير الحج السعودي الدكتور بندر بن محمد حجار، الجمعة، إنشاء شبكة إلكترونية عالمية تختص بتلقي طلبات تأشيرات العمرة، وإصدارها مرتبطة بوزارتي الداخلية والخارجية والجهات الحكومية السعودية الأخرى .و قال الوزير السعودي في تصريحات إن الشبكة ستطبق الأنظمة التي تخص متابعة الأوضاع النظامية لشركات ومؤسسات العمرة وتحقيقها للاشتراطات والمتطلبات المحددة لتقديم الخدمة بمشاركة شركات ومؤسسات العمرة التي تختص بتقديم الخدمات للمعتمرين. وذكر الوزير أن تطبيق مثل هذه التقنية العالمية من شأنه التسهيل على ضيوف الرحمن لأداء نسكهم مع الحفاظ على عنصر الجودة والرقي في تقديم منظومة الخدمات لهم قبل وصولهم المملكة وأثناء وصولهم حتى مغادرتهم بعد أداء الفريضة .وأوضح أنه يشهد على ذلك ما أعلنته منظمة «شبكة العالم»، المتخصصة في تقييم مراكز المعلومات في العالم من حصول الوزارة من خلال مركز معلومات العمرة على جائزة أفضل مركز معلومات في الشرق الأوسط لاستخدامه أحدث الأساليب التقنية والاتصالات

The Saudi Minister of Hajj, Dr. Bandar bin Muhammad Hajjar, announced on Friday the establishment of a global electronic network specialized in receiving and issuing Umrah visa requests linked to the Ministries of Interior and Foreign Affairs and other Saudi government agencies. And the Saudi minister said in statements that the network will apply the regulations pertaining to following up on regular conditions. For Umrah companies and institutions and their fulfilment of the requirements and specific requirements for providing the service with the participation of Umrah companies and institutions that specialize in providing services to pilgrims. Their arrival until their departure after performing the obligatory prayer. He explained that this is evidenced by what was announced by the Network of the World organization which specializes in evaluating information centers in the world, that the Ministry through the Umrah Information Center won the award for the best information center in the Middle East for its use of the latest technical methods and communications.

Figure 16. A sample of an Egyptian article misclassified as Saudi Arabian from the Hajj articles dataset.

6. Conclusions and Future Work

In this study, we developed a classification model that categorizes Arabic news articles based on their country of origin. We performed a thorough empirical investigation by assessing several feature selection techniques within a classification model using machine learning and deep and hybrid deep learning models.

We trained and tested two datasets of Arabic news articles from two topics, the Hajj and Brexit, from three Arabic countries, Saudi Arabia, Jordan, and Egypt, along with heterogeneous feature spaces. Our work utilized the ML algorithms SVM, Multinomial NB, Bernoulli NB, SGD classifier, decision tree, random forest, and KNN, and demonstrated that classifying articles based on their country of origin was an amenable problem. For the traditional machine learning algorithms, a combination of n-grams with linear SVC was superior to the other ML algorithms for the Hajj dataset, with n-gram = 3 with linear SVC best for the Brexit dataset, achieving accuracies of 93.1% and 83.0%, respectively. On the other hand, for the deep learning and hybrid deep learning models with the Hajj dataset, random embedding with CNN+LSTM reached an accuracy of 94.4%, while FastText embedding with CNN reached the highest accuracy of 76.87% for the Brexit dataset.

We intend to analyze the impact of new features used on the classification results. Additionally, we will explore other data types for classification, such as image and audio augmentation, to improve the model's performance and accuracy. Finally, we intend to carry out an extensive comparison between machine learning algorithms and deep learning models on various datasets containing a substantial number of articles. In conclusion, this research has shown that deep learning is a powerful tool for classifying Arabic news articles by country of origin. An investigation of how to detect and identify the source and alleviate the bias in natural language datasets can be conducted for future research. Additionally, we intend to develop more advanced and hybrid models and train these with more features to achieve improved results.

Author Contributions: Conceptualization, N.Z. and H.H.; methodology, N.Z. and S.F.S.; software, S.F.S. and H.H.; validation, S.F.S. and H.H.; formal analysis, N.Z.; data curation, H.H.; writing—original draft preparation, N.Z. and H.H.; writing—review and editing, H.H.; visualization, N.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fouad, K.M.; Sabbeh, S.F.; Medhat, W. Arabic Fake News Detection Using Deep Learning. *CMC-Comput. Mater. Contin.* **2022**, *71*, 3647–3665. [CrossRef]
2. Schumaker, R.P.; Chen, H. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Trans. Inf. Syst. (TOIS)* **2009**, *27*, 1–19. [CrossRef]
3. Himdi, H.; Weir, G.; Assiri, F.; Al-Barhamtoshy, H. Arabic fake news detection based on textual analysis. *Arab. J. Sci. Eng.* **2022**, *47*, 10453–10469. [CrossRef] [PubMed]
4. Rao, V.; Sachdev, J. A machine learning approach to classify news articles based on location. In Proceedings of the 2017 IEEE International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 7–8 December 2017; pp. 863–867.
5. Abdulla, H.H.H.A.; Awad, W.S. Text Classification of English News Articles using Graph Mining Techniques. In Proceedings of the ICAART (3), Online, 3–5 February 2022; pp. 926–937.
6. Sreedevi, J.; Rama Bai, M.; Reddy, C. Newspaper article classification using machine learning techniques. *Int. J. Innov. Technol. Explor. Eng.* **2020**, *12*, 2278–3075.
7. Zhu, Y. Research on news text classification based on deep learning convolutional neural network. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1–6. [CrossRef]
8. MAHAJAN, S.; Ingle, D. News Classification Using Machine Learning. *Int. J. Recent Innov. Trends Comput. Commun.* **2021**, *9*, 23–27. [CrossRef]
9. Ahmed, J.; Ahmed, M. Online news classification using machine learning techniques. *IIUM Eng. J.* **2021**, *22*, 210–225. [CrossRef]
10. Saigal, P.; Khanna, V. Multi-category news classification using Support Vector Machine based classifiers. *SN Appl. Sci.* **2020**, *2*, 458. [CrossRef]
11. Watanabe, K. Newsmap. *Digit. J.* **2018**, *6*, 294–309. [CrossRef]
12. Al-Barhamtoshy, H.M.; Himdi, H.T.; Alyahya, M. Arabic Pilgrim Services Dataset: Creating and Analysis. In Proceedings of the 2023 IEEE 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), Jeddah, Saudi Arabia, 23–25 January 2023; pp. 1–8.
13. Weikum, G. Foundations of Statistical Natural Language Processing. *SIGMOD Rec.* **2002**, *31*, 37–38. [CrossRef]
14. Aggarwal, C.C.; Zhai, C. A survey of text classification algorithms. In *Mining Text Data*; Springer: New York, NY, USA 2012; pp. 163–222.
15. Guo, B.; Zhang, C.; Liu, J.; Ma, X. Improving text classification with weighted word embeddings via a multi-channel TextCNN model. *Neurocomputing* **2019**, *363*, 366–374. [CrossRef]
16. Braga, I.; Monard, M.; Matsubara, E. Combining unigrams and bigrams in semi-supervised text classification. In Proceedings of the Progress in Artificial Intelligence, 14th Portuguese Conference on Artificial Intelligence (EPIA 2009), Aveiro, Portugal, 12–15 October 2009; pp. 489–500. Available online: https://www.researchgate.net/profile/Maria-Carolina-Monard/publication/228678357_Combining_unigrams_and_bigrams_in_semi-supervised_text_classification/links/544e6ac30cf2bca5ce90b302/Combining-unigrams-and-bigrams-in-semi-supervised-text-classification.pdf (accessed on 1 June 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.