Journal of Theoretical and Applied Information Technology
31st October 2022. Vol.100. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DETECTION OF FAKE NEWS IN THE SPANISH LANGUAGE USING MACHINE LEARNING TECHNIQUES

SYOMIRA CHAMBI-APAZA[1], ROXANA FLORES-QUISPE[2]
[1,2]School of Computer Science, Universidad Nacional de San Agustín de Arequipa, Arequipa - Perú
E-mail: [1]schambia@unsa.edu.pe, [2]rfloresqu@unsa.edu.pe

ABSTRACT

Nowadays, fake news has become a huge problem that causes damage around the world, especially in the social, political, and economic spheres. Due to the large amount of news generated every day, it is difficult to verify manually all the information to determine if a news item is real or fake. As a result, expert-based manual fact-checking, such as editors and journalists, need new tools that can perform the verification process efficiently. On the other hand, there are many studies focused on the detection of fake news in the English language, however, in the Spanish language, there are only a few researches that address this issue. For that reason, this proposed research explores different machine learning techniques to detect fake news in the Spanish language considering three feature extraction techniques: TF, TF-IDF, and Count Vectorizer; and five machine learning techniques: Logistic Regression, Stochastic Gradient Descent, Gradient Boosting, Random Forest and Support Vector Machine, were investigated and compared between them in order to achieve the classification task. Finally, the experimental results show the best performance with an accuracy rate of 87.18% using Random Forest as a classifier and TF as a feature extractor.

Keywords: *Fake News, Automatic Detection, News Verification, Disinformation, Machine Learning.*

1. INTRODUCTION

It is becoming increasingly difficult to discern what is true from what is false. The use of political, media, and social concepts such as fake news is a global problem that affects the entire population, including the media and the Internet [1]. In fact, fake news is now viewed as one of the greatest threats to democracy, journalism, and freedom of expression [2], and even our economies are not immune to the spread of fake news either, with fake news being connected to stock market fluctuations and large trades [3].

In the political sphere, disinformation has been used mainly in campaigns to discredit political opponents. For example, in the 2016 presidential elections in the United States, an investigation confirmed that during the electoral campaign a total of 115 favorable fake news from Donald Trump were generated, which were shared on Facebook a total of 30 million times, compared to 41 fake news to benefit Hillary Clinton shared 7.6 million times [4].

Likewise, other democratic processes have been clouded by disinformation campaigns, such as the victory of Brexit in the United Kingdom [5], the triumph of "no" in the referendum for peace in Colombia [5], the victory of Bolsonaro in the 2018 elections in Brazil [6], the constitutional reform proposal in Italy [7], among others. Unfortunately, fake news outlets lack the news media's editorial norms and processes for ensuring the accuracy and credibility of information [8].

In this sense, the conventional solution to this issue is to ask professionals such as journalists to check claims against evidence-based on previously spoken or written facts [9]. In addition, currently, there are some expert-based manual fact-checking websites. Despite this, due to the large volume of news that is generated every day, it is difficult to verify all the data to be able to determine whether a news article is real or fake, making these tools inefficient.

On the other hand, there are three stages within a news life cycle: being created, published online, and propagating on social media. To detect fake news at an early stage, i.e., when it is published on a news outlet but not yet spread on social media, one cannot rely on news propagation information as it does not exist. Such early detection is particularly crucial for fake news as if more individuals become exposed to fake news, the more likely they may trust it [2]. However, there is the limitation of only being able to analyze its content, as the characteristics of the news context have not yet been generated.

In addition, there is a great variety of studies focused on the detection of fake news in the English language. However, in the Spanish language, few studies address this same issue [10]. Furthermore, it is necessary to emphasize there is a gap between the number of samples in the English and Spanish datasets, since even some datasets in the English language exceed one million samples, unlike the existing datasets in the Spanish language [10][11].

Based on these limitations, this research can identify news articles in the Spanish language as fake or real using machine learning techniques, where the news content was analyzed using Logistic Regression, Stochastic Gradient Descent, Gradient Boosting, Random Forest, and Support Vector Machine, and then they were compared using TF, TF-IDF, and Count Vectorizer as feature extractors.

The rest of this paper is organized as follows. The review of related works proposed in the literature is presented in Section 2. The general scheme of the proposed method is described in Section 3. The experiments and the results based on a corpus that contains news articles from the real world are shown in Section 4. Finally, the conclusions of this research are mentioned in Section 5.

2. RELATED WORKS

In recent years, the discovery of fake news has received special attention in the literature. Considering that most of the works are oriented to the English language, these have taken the task as a classification problem.

In that way Zhou et al. [3] have proposed an interdisciplinary study that was conducted for the early detection of fake news, this work comprehensively studied and represented news content at four language levels: lexical, syntactic, semantic, and discourse.

Their representation was inspired by well-established theories in social and forensic psychology, taking place within a supervised machine learning framework. Experimental results based on real-world datasets indicate that the performance of the proposed model can reach up to 88%.

Likewise, Gravanis et al. [12] proposed a model for fake news detection using content-based features and machine learning algorithms. To conclude the more accurate model, they evaluated several proposed feature sets for deception detection and also for word insertion. The proposed features combined with the machine learning algorithms obtained an accuracy of up to 95% in all datasets used with AdaBoost to be the first in the rank and the SVM and Bagging algorithms to be the next in the ranking.

Similarly, Ahmed et al. [13] proposed a fake news detection model that uses n-gram analysis techniques and machine learning. They investigated and compared two different feature extraction techniques and six different machine classification techniques. The experimental evaluation produced the best performance using TF-IDF as the feature extraction technique and LSVM as the classifier, with an accuracy of 92%.

Also, some researchers have proposed a new solution for detecting fake news that incorporates sentiment as an important feature to improve accuracy, such as Bhutani et al. [14] who analyze different text preprocessing techniques and selected TF-IDF with cosine similarity as the best approach using accuracy as an evaluation metric.

In addition, Kaur et al. [15] analyzed different machine learning techniques together with three feature extraction techniques (TF-IDF, CV, and HV) based on performance measures. After analyzing the classifiers, an approach was proposed that used the strengths of one model to complement the weakness of another to generate the multilevel voting model, and also integrates various machine learning models based on their false positive rates to retrieve news voting classifier to retrieve better prediction analysis.

On the other hand, some researchers have used a deep learning approach to detecting fake news in the English language, such as Thota et al. [16] who proposed the architecture of a neural network to accurately predict posture based on a given headline and article body. Their model achieved 94.21% accuracy on test data.

Likewise, Liu et al. [17] proposed a model for the early detection of fake news on social media by classifying the routes of news spread. Experimental results on three real-world datasets showed that this model can detect fake news with an accuracy of 85% and 92% on Twitter and Sina Weibo, respectively, within 5 minutes after it starts to spread.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Furthermore, other researchers also considered images to detect fake news such as Yang et al. [18] that proposed a unified TI-CNN model, which can combine text and image information with the corresponding explicit and latent characteristics. The proposed model has great expandability, which can easily absorb other news features. Experimental results showed that TI-CNN can successfully identify fake news with 92.20%.

On the other hand, other researchers have focused on detecting fake news in the Spanish language, such as the work of Posadas-Durán et al. [10] that presented the first corpus of fake news in Spanish and a method for detecting fake news. They trained well-known classification algorithms on lexical features BOW, POS tags, and n-grams. The classification results showed a performance of up to 76.94% using BOW + POS and Random Forest as the classifier.

Similarly, Queiroz et al. [19] evaluated textual characteristics that are not linked to a specific language when describing textual data to detect news. They explored news corpus written in American English, Brazilian Portuguese, and Spanish to study the complexity, stylometric, and psychological text features. They compared four machine learning algorithms to induce the detection pattern and the results showed that the proposed language-independent features are successful with an average detection accuracy of 85.3% using Random Forest.

Additionally, Martinez-Gallego et al. [20] used a deep learning approach to detecting fake news in the Spanish language. The Deep Learning architectures were built on top of different pre-trained Word Embedding representations. They used four datasets, two in English and two in Spanish, and four experimental schemes were tested. According to the results, the best strategy was a combination of a pretrained BETO model and a recurrent neural network based on LSTM layers, yielding an accuracy of up to 80%.

The review of the related works shows that many of these are focused on the detection of fake news for the English language, by varying in accuracy according to the machine learning or deep learning technique and the feature extraction technique used. Considering that the works focused on the detection of fake news in the Spanish language are few, this research proposes to use machine learning based on techniques little discussed in a corpus in Spanish, in order to validate if these techniques can also obtain good performance with this language.

3. PROPOSED METHOD

In figure 1 the graphical representation of the proposed method for the detection of fake news in the Spanish language in five stages is shown. A detailed description of each module is provided below.

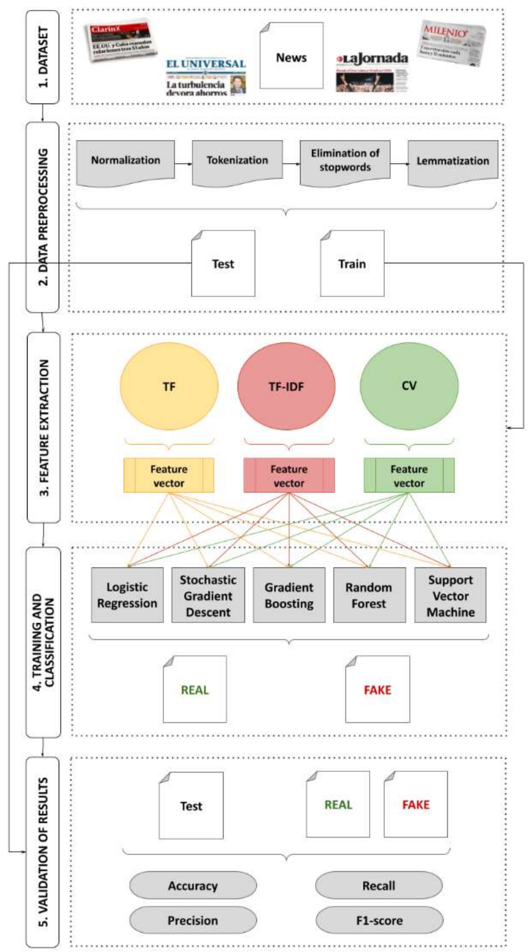

Figure 1. The proposed method for the detection of fake news in the Spanish language.

3.1. News Dataset

The dataset used in this work is The Spanish Fake News Corpus [10] which contains 971 news articles divided into 491 real news and 480 fake news. It covers news from 9 different topics: Science, Sport, Economy, Education, Entertainment, Politics, Health, Security, and Society. In table 1 the distribution of the news by the different categories is shown.

These news articles were collected from January to July of 2018 from several resources on the Web: established newspapers' websites, media companies' websites, special websites dedicated to validating fake news, and websites designated by different journalists as sites that regularly publish fake news; all of them were written in Spanish [10].

This dataset specifically provides the headline, the content, the label, the source, and the topic of each news item, of which only the news content and its label as real or fake were used.

Table 1. Distribution of the news dataset by category.

Category	Real	Fake
Science	46	43
Sport	66	58
Economy	24	19
Education	10	12
Entertainment	70	78
Politics	175	148
Health	23	23
Security	17	25
Society	60	74
Total	**491**	**480**

3.2. Data Preprocessing

Data preprocessing is an essential part of any Natural Language Processing system, since the characters, words, or sentences identified at this stage are the fundamental units passed to all further processing stages [21]. In this work, the training data and the test data were preprocessed to convert them into ready-to-use data, this process was subdivided into four stages: normalization, tokenization, elimination of stop words, and lemmatization.

3.2.1. Normalization

The news content of the chosen dataset has special characters, such as exclamation marks, punctuation marks, quotation marks, commas, among others. In this stage, these characters are removed and the uppercase letters were converted into lowercase since their existence within the data could affect the recognition of the expressions in further stages. Table 2 shows an example.

Table 2. Example of normalization.

Original text	Transformed text
Pagó la deuda de su hermano, Boris Johnson.	pago la deuda de su hermano boris johnson

3.2.2. Tokenization

In this stage, the text flow is divided into words, which will be significant elements called tokens. The generated token list will become input for further processing. Table 3 shows an example.

Table 3. Example of tokenization.

Original text	Transformed text
pago la deuda de su hermano boris johnson	[“pago”, “la”, “deuda”, “de”, “su”, “hermano”, “boris”, “johnson”]

3.2.3. Elimination of stop words

In this stage, stop words (articles, prepositions, pronouns, conjunctions, among others) were removed to discard irrelevant information and it permit to focus only on important information. Table 4 shows an example.

Table 4. Example of elimination of stop words.

Original text	Transformed text
[“pago”, “la”, “deuda”, “de”, “su”, “hermano”, “boris”, “johnson”]	[“pago”, “deuda”, “hermano”, “boris”, “johnson”]

3.2.4. Lemmatization

In this stage, for each inflected word form, its lemma is identified and its inflectional endings are removed, thus returning the words to dictionary form. Table 5 shows an example.

Table 5. Example of lemmatization.

Original text	Transformed text
[“pago”, “deuda”, “hermano”, “boris”, “johnson”]	[“pag”, “deud”, “herman”, “boris”, “johnson”]

3.3. Feature Extraction

Machine learning algorithms operate on a numerical feature space, expecting a numeric vector as input. Therefore, in order to perform machine learning on text, we need feature extraction techniques to transform a text into a feature vector [22]. Three different feature extraction techniques were used in this research: Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), and Count Vectorizer (CV).

3.3.1. Term Frequency

Terms that are frequently mentioned in individual documents appear to be useful as recall-enhancing devices. For this reason, Term Frequency or TF is often used as a term-weighting system, which measures the frequency of occurrence of the terms in a document in the normalized form [23].

3.3.2. Term Frequency-Inverse Document Frequency

TF-IDF is the product of TF and IDF, where IDF or Inverse Document Frequency favors terms concentrated in a few documents of a collection. It varies inversely with the number of documents n to which a term is assigned in a collection of N documents, where a typical IDF factor may be computed as log N/n [23]. In other words, IDF reduces the weight of terms that occur very frequently in the document set and increases the weightof terms that occur rarely [24].

3.3.3. Count Vectorizer

In this method, the documents and queries are all vectorized using a count vectorizer (i.e) the count of each word in the documents (query and statute) [25].

3.4. Training and Classification

After preprocessing the dataset and extracting features that represent the documents involved, the feature vectors are sent to the classification phase in order to start the training stage. In this research, the computational characteristics of each news article were used by five classifiers that are widely accepted and well established, which are: Logistic Regression, Stochastic Gradient Descent, Gradient Boosting, Random Forest, and Support Vector Machine.

3.4.1. Logistic Regression (LR)

Logistic Regression is a type of logarithm linear model that computes the probabilities of different classes through parametric logistic distribution [26]. It can be viewed as arising from a Bernoulli model, where, given a set of predictors x_n, we can determine the probability of a binary outcome y [27], the probability of the model is defined in equation 1.

$$p(y_n = 1|x_n) = \frac{1}{1 + \exp(-w \cdot x_n)} \qquad (1)$$

One way to interpret the logistic regression is to view it as a method to maximize $p(y_n = 1|x_n)$ for each point (x_n, y_n) in the training set [28].

3.4.2. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent is an iterative optimization algorithm that is faster, more reaching, and less prone to reaching bad local minimum than standard gradient descent. In SGD, the weights are updated after the presentation of each example, according to the gradient of the loss function. Each iteration of the SGD algorithm consists in drawing an example z at random and applying the parameter update rule [29], which is shown in equation 2.

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t) \qquad (2)$$

SGD does not require the update direction to be based exactly on the gradient, instead allows the direction to be a random vector and only requires that its expected value at each iteration will equal the gradient direction. More generally, it requires that the expected value of the random vector will be a subgradient of the function at the current vector [30].

3.4.3. Gradient Boosting (GB)

Gradient boosting is a powerful machine learning technique [31] whose main idea is to construct the new base-learners to be maximally correlated with the negative gradient of the loss function, associated with the whole ensemble [32]. Given a training dataset $D = (x_i, y_i)_1^N$ the goal of gradient boosting is to find an approximation $F(x)$ of the function $F^*(x)$, which maps instances x to their output values y, by minimizing the expected value of a given loss function, $L(y, F(x))$ [33]. Gradient Boosting builds, and additive approximation $F^*(x)$ as a weighted sum of functions, such as is shown in equation 3.

$$F_m(x) = F_{m-1}(x) + \rho_m h_m(x) \qquad (3)$$

Where ρ_m is the weight of the m^{th} function, $h_m(x)$ [33].

3.4.4. Random Forest (RF)

Journal of Theoretical and Applied Information Technology
31st October 2022. Vol.100. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [34]. More formally, for a ρ-dimensional random vector X = (X1,...,Xp)T representing the real-valued input or predictor variables and a random variable "Y" representing the real-valued response, we assume an unknown joint distribution PXY (X,Y). The goal is to find a prediction function f (X) for predicting Y [35]. The prediction function is determined by a loss function L(Y, f (X)) to minimize the expected value of the loss, such as is shown in equation 4.

$$E_{XY}(L(Y, F(X))) \tag{4}$$

Where the subscripts denote expectations concerning the distribution of X and Y [35].

3.4.5. Support Vector Machine (SVM)

Support Vector Machines are a set of supervised learning methods [36] which have the aim of determining the location of the decision boundary (hyperplane) that produces the optimal separation of classes [37].

More specifically, from the given training data, SVM splits the data into distinct groups by an optimal separator called hyperplane, the major concern of SVM is to find the hyperplane which separates the points closest to the separator (support vectors) in a data space [37]. An optimal separator can be calculated from equation 5.

$$\underset{\alpha}{argmax} \sum_j \alpha_j - \frac{1}{2}\sum_{j,k} \alpha_j \alpha_k y_j y_k (x_j \cdot x_k) \tag{5}$$

Subject to the constraints $\alpha_j \geq 0$ and $\sum_j \alpha_j y_j = 0$. SVMs create a linear separating hyperplane, but they can embed the data into a higher-dimensional space, using the so-called kernel trick [38].

3.5. Validation of Results

The performance measures for classifiers applied in this research have been evaluated using a confusion matrix defined by the cells that are shown in table 6.

Table 6. Confusion Matrix Representation.

Actual ↓ Predicted →	Fake	Real
Fake	true positives (tp)	false positives (fp)
Real	false negatives (fn)	true negatives (tn)

The conventional performance measures have been evaluated from the above confusion matrix cells. The measures computed were precision, recall, accuracy, and F1-score.

These measures are represented by equations 6, 7, 8, and 9.

$$Precision = \frac{tp}{tp + fp} \tag{6}$$

$$Recall = \frac{tp}{tp + fn} \tag{7}$$

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{8}$$

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

4. EXPERIMENTS AND RESULTS

In the experiments carried out, the dataset mentioned in Section 3.1 was used to evaluate the generality and performance of the proposed models for detecting fake news in the Spanish language. The dataset was randomly divided into training data and test data based on a ratio of 0.8:0.2 taking the value 7 as the random state to minimize overfitting [39].

The experiments were performed using five machine learning techniques, namely, Logistic Regression, Stochastic Gradient Descent, Gradient Boosting, Random Forest, and Support Vector Machine (with the linear kernel), and three feature extraction techniques, namely, Term Frequency, Term Frequency-Inverse Document Frequency, and Count Vectorizer.

Sklearn [40] was used for all experiments, considering the default hyperparameters that it provides for the classification techniques and a maximum frequency of 0.7 for the feature extraction techniques.

After carrying out the corresponding tests, the confusion matrices were assembled in order to obtain the performance of each machine learning technique

Journal of Theoretical and Applied Information Technology
31st October 2022. Vol.100. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in terms of accuracy, precision, recall, and F1- score. Figures 2, 3, and 4 show the confusion matrices which compare the current label with the predicted label.

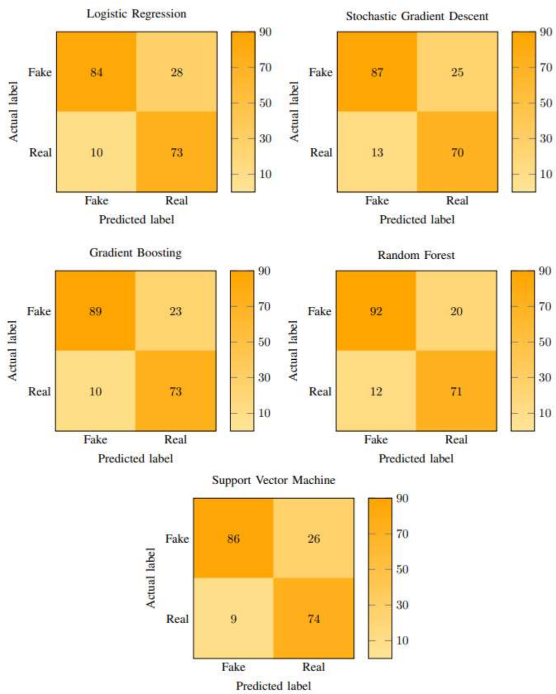

Figure 2. Confusion matrices comparing the actual label of samples with the predicted label when TF was used as a feature extractor.

More specifically, in figure 2 the most remarkable fact is that the model Random Forest model using Term Frequency as the feature extractor managed to correctly classify 92 news articles as fake and only 20 were misclassified as real.

In figure 3, the most remarkable fact is that the model Random Forest using Term Frequency-Inverse Document Frequency as the feature extractor managed to correctly classify 95 news articles as fake and only 17 were misclassified as real.

In contrast, figure 4 shows that using Count Vectorizer as a feature extractor, the Logistic Regression model, and the Support Vector Machine were able to correctly classify 88 news articles as fake and 24 were misclassified as real.

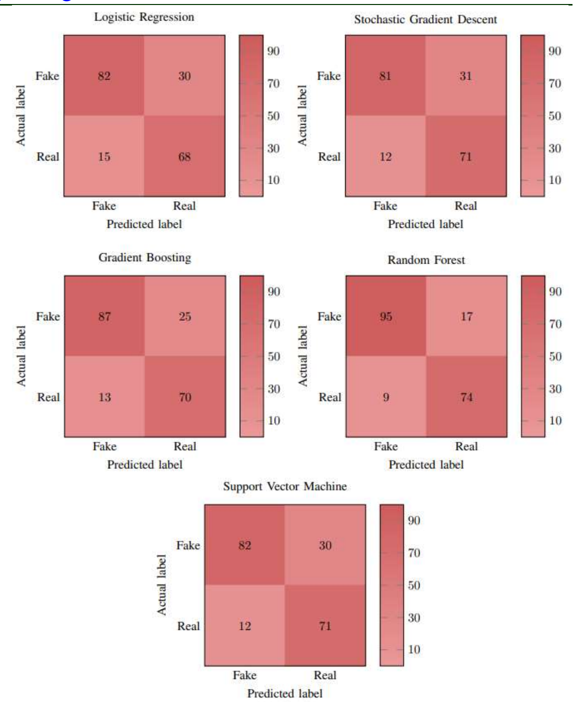

Figure 3. Confusion matrices comparing the actual label of samples with the predicted label when TF-IDF was used as a feature extractor.

From these confusion matrices, the performance measures detailed in Section 3.5 were obtained. The results are presented in figure 5 and they show the performance of the classification models considering the proposed method detailed in Section 3 (using elimination of stop words and lemmatization).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

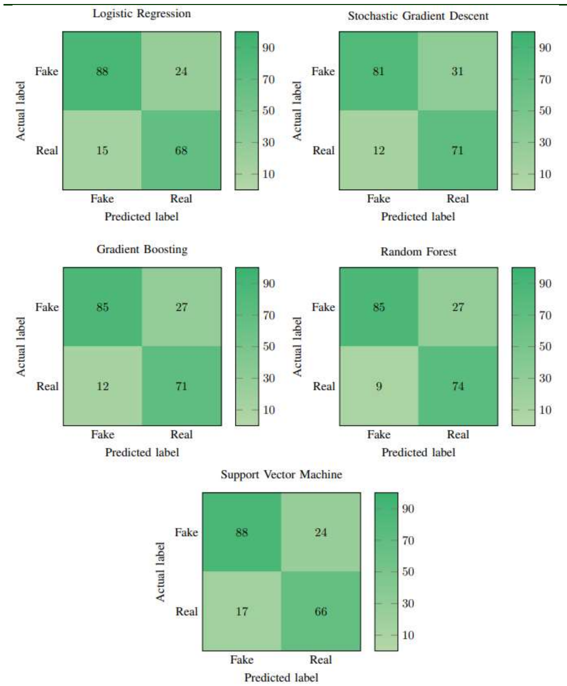

Figure 4. Confusion matrices comparing the actual label of samples with the predicted label when Count Vectorizer was used as a feature extractor.

In general, it can be seen that Random Forest, using TF-IDF as feature extractor, obtains better performance in terms of accuracy (86.67%), precision (91.35%), recall (84.82%), and F1-score (87.96%), which are the highest percentages in the graphs of figure 5.

In that sense, these results can be because when using Random Forest the trees protect each other from their individual errors, since although some trees can be wrong, many other trees will be correct, so as a group the trees can be directed to the correct classification.

In order to improve the results, the application of the lemmatization and elimination of stop words process was alternated for each combination of a machine learning technique and a feature extraction technique. Based on this, table 7 shows the performance of the classification models proposed applied to the Spanish fake news dataset in terms of accuracy, precision, recall, and F1- score.

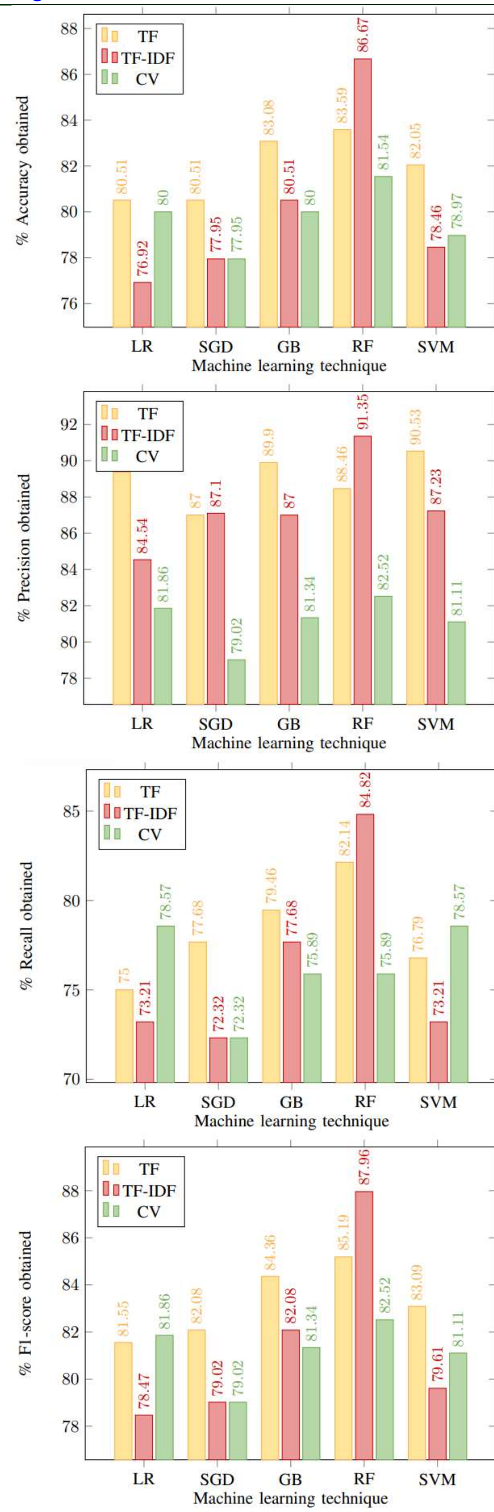

Figure 5. Performance analysis using accuracy, precision, recall, and F1-score metrics.

Journal of Theoretical and Applied Information Technology
31st October 2022. Vol.100. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 7. Comparative Analysis of Accuracy, Precision, Recall, and F1-Score Metrics using Machine Learning Techniques.

Model	Feature extractor	Lemmatization	Elimination of stop words	Precision	Recall	F1-score	Accuracy
LR	TF	NO	YES	0.8977	0.7054	0.7900	0.7846
SGD	TF	NO	YES	0.9011	0.7321	0.8079	0.8000
GB	TF	NO	YES	0.8936	0.7500	0.8155	0.8051
RF	TF	NO	YES	0.9065	0.8661	0.8858	**0.8718**
SVM	TF	NO	YES	0.9121	0.7411	0.8177	**0.8103**
LR	TF	YES	NO	0.8851	0.6875	0.7739	0.7692
SGD	TF	YES	NO	0.9032	0.7500	0.8195	**0.8103**
GB	TF	YES	NO	0.8764	0.6964	0.7761	0.7692
RF	TF	YES	NO	0.9109	0.8214	0.8638	**0.8513**
SVM	TF	YES	NO	0.8776	0.7679	0.8190	0.8051
LR	TF	YES	YES	0.8977	0.7054	0.7900	0.7846
SGD	TF	YES	YES	0.8700	0.7768	0.8208	0.8051
GB	TF	YES	YES	0.8990	0.7946	0.8436	**0.8308**
RF	TF	YES	YES	0.8846	0.8214	0.8519	**0.8359**
SVM	TF	YES	YES	0.9053	0.7679	0.8309	0.8205
LR	TF-IDF	NO	YES	0.8947	0.6071	0.7234	0.7333
SGD	TF-IDF	NO	YES	0.8617	0.7232	0.7864	0.7744
GB	TF-IDF	NO	YES	0.8431	0.7679	0.8037	**0.7846**
RF	TF-IDF	NO	YES	0.8942	0.8304	0.8611	**0.8462**
SVM	TF-IDF	NO	YES	0.8632	0.7321	0.7923	0.7795
LR	TF-IDF	YES	NO	0.9024	0.6607	0.7629	0.7641
SGD	TF-IDF	YES	NO	0.8936	0.7500	0.8155	**0.8051**
GB	TF-IDF	YES	NO	0.9070	0.6964	0.7879	0.7846
RF	TF-IDF	YES	NO	0.9010	0.8125	0.8545	**0.8410**
SVM	TF-IDF	YES	NO	0.8817	0.7321	0.8000	0.7897
LR	TF-IDF	YES	YES	0.8974	0.6250	0.7368	0.7436
SGD	TF-IDF	YES	YES	0.8710	0.7232	0.7902	0.7795
GB	TF-IDF	YES	YES	0.8700	0.7768	0.8208	**0.8051**
RF	TF-IDF	YES	YES	0.9135	0.8482	0.8796	**0.8667**
SVM	TF-IDF	YES	YES	0.8723	0.7321	0.7961	0.7846
LR	CV	NO	YES	0.8571	0.8571	0.8571	**0.8359**
SGD	CV	NO	YES	0.8261	0.6786	0.7451	0.7333
GB	CV	NO	YES	0.8878	0.7768	0.8286	**0.8154**
RF	CV	NO	YES	0.8842	0.7500	0.8116	0.8000
SVM	CV	NO	YES	0.8426	0.8125	0.8273	0.8051
LR	CV	YES	NO	0.8796	0.8482	0.8636	**0.8462**
SGD	CV	YES	NO	0.8625	0.6161	0.7188	0.7231
GB	CV	YES	NO	0.8785	0.8393	0.8584	0.8410
RF	CV	YES	NO	0.8942	0.8304	0.8611	**0.8462**
SVM	CV	YES	NO	0.8491	0.8036	0.8257	0.8051
LR	CV	YES	YES	0.8482	0.8482	0.8482	**0.8256**
SGD	CV	YES	YES	0.8710	0.7232	0.7902	0.7795
GB	CV	YES	YES	0.8763	0.7589	0.8134	0.8000
RF	CV	YES	YES	0.9043	0.7589	0.8252	**0.8154**
SVM	CV	YES	YES	0.8381	0.7857	0.8111	0.7897

In that sense, an improvement in the results was obtained, since the model that showed the best performance in this dataset was Random Forest together with TF using elimination of stop words and no lemmatization, obtaining an accuracy of up to 87.18%. In addition, it can be seen in the same table 7 that while Random Forest together with TF has the best performance, this model also gets a good behavior with TF-IDF and Count Vectorizer, that is, regardless of the extraction technique, Random Forest performs well.

Likewise, another model that performed well was Logistic Regression with Count Vectorizer using lemmatization and without elimination of stop words obtaining an accuracy of up to 84.62%. Also, Logistic Regression and Count Vectorizer perform well regardless of whether using stemming or elimination of stop words. However, unlike Random Forest, this model did not achieve similar performance to the other feature extraction techniques, TF and TF-IDF.

On the other hand, Gradient Boosting obtained an accuracy of up to 84.10% with Count Vectorize as feature extractor using lemmatization and without elimination of stop words. Support Vector Machine with TF and using lemmatization and elimination stop words obtained an accuracy of up to 82.05%. Finally, Stochastic Gradient Descent obtained an accuracy up to 81.03% with TF and using lemmatization and without elimination of stop words.

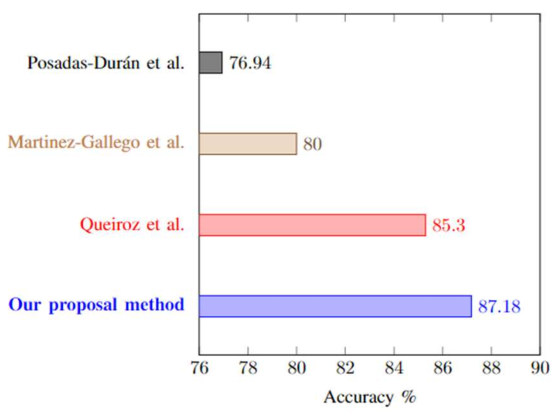

Figure 6. Comparison of the performance between our proposal method and the performance of related works for detecting fake news in the Spanish language in terms of accuracy.

Considering that there are few works focused on the detection of fake news in the Spanish language, this research used machine learning techniques little discussed in a corpus in Spanish to validate if these techniques also obtain good performance with this language. In that sense, to show the effectiveness of our method, we compared the results with related works shown in section 2, it can be seen in figure 6, where our experiments applied to the dataset in Spanish obtained the best results.

5. CONCLUSION

In this research, the objective of developing a method for the detection of false news in the Spanish language using Machine Learning techniques was achieved. To the experiments, only the news content and its label as real or fake were used. Also, the computational features of each news item were used by five widely accepted and well-established classifiers.

Initially, the experimental results based on the proposed method showed that Random Forest, using TF-IDF as a feature extractor, obtains better performance in terms of accuracy (86.67%). Then, to obtain better results, it was experimented with alternating the application of the lemmatization and elimination of stop words for each combination of a machine learning technique and a feature extraction technique.

Based on this, the accuracy of the experimental results reached up to 87.18%, using Random Forest as a classification model and TF as a feature extractor with the elimination of stop words and without lemmatization, being the most optimal combination among those analyzed in this research. The results show the proposed method works well in comparison with related works that also focus on the problem of detecting fake news in the Spanish language.

In addition, it was observed that although Random Forest together with TF has the best performance, this model also obtains great results with TF-IDF and Count Vectorizer, that is, regardless of the extraction technique used by Random Forest, it worked well.

However, the performance achieved in the works focused on the detection of fake news in the English language, shown in section 2, is higher than that obtained in this research, which can be explained by the small size of the dataset used in this work, since the machine learning techniques used depend highly on the amount of experimental data with which they are trained. Therefore, to consistently reveal additional patterns in fake news content compared to real news content, it is suggested to use larger-scale datasets to obtain better results.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] P. P. Valero and L. Oliveira, "Fake news: una revisión sistemática de la literatura" Observatorio (OBS*), vol. 12, no. 5, 2018.

[2] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," ACM Computing Surveys (CSUR), vol. 53, no. 5, pp. 1–40, 2020.

[3] X. Zhou, A. Jain, V. V. Phoha, and R. Zafarani, "Fake news early detection: An interdisciplinary study," arXiv preprint arXiv:1904.11679, 2019.

[4] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," Journal of economic perspectives, vol. 31, no. 2, pp. 211–36, 2017.

[5] C. Pauner Chulvi, "Noticias falsas y libertad de expresión e información. el control de los contenidos informativos en la red," Teoría y realidad constitucional, vol. 41, pp. 297–318, 2018.

[6] C. B. Fernández, "Ciberpolítica 2018: tendencias en Latinoamérica," Nuevas campañas electorales en América Latina, pp. 147–162, 2018.

[7] M. Del Vicario, S. Gaito, W. Quattrociocchi, M. Zignani, and F. Zollo, "News consumption during the italian referendum: A cross-platform analysis on facebook and twitter," in 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 648–657, IEEE, 2017.

[8] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, et al., "The science of fake news," Science, vol. 359, no. 6380, pp. 1094– 1096, 2018.

[9] R. Oshikawa, J. Qian, and W. Y. Wang, "A survey on natural language processing for fake news detection," arXiv preprint arXiv:1811.00770, 2018.

[10] J.-P. Posadas-Durán, H. Gomez-Adorno, G. Sidorov, and J. J. M. Escobar, "Detection of fake news in a new corpus for the spanish language," Journal of Intelligent & Fuzzy Systems, vol. 36, no. 5, pp. 4869–4876, 2019.

[11] K. Nakamura, S. Levy, and W. Y. Wang, "A new multimodal bench- mark dataset for fine-grained fake news detection," arXiv preprint arXiv:1911.03854, 2019.

[12] G. Gravanis, A. Vakali, K. Diamantaras, and P. Karadais, "Behind the cues: A benchmarking study for fake news detection," Expert Systems with Applications, vol. 128, pp. 201–213, 2019.

[13] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in International conference on intelligent, secure, and dependable systems in distributed and cloud environments, pp. 127–138, Springer, 2017.

[14] B. Bhutani, N. Rastogi, P. Sehgal, and A. Purwar, "Fake news detection using sentiment analysis," in 2019 Twelfth International Conference on Contemporary Computing (IC3), pp. 1–5, IEEE, 2019.

[15] S. Kaur, P. Kumar, and P. Kumaraguru, "Automating fake news detection system using multi-level voting model," Soft Computing, vol. 24, no. 12, pp. 9049–9069, 2020.

[16] A. Thota, P. Tilak, S. Ahluwalia, and N. Lohia, "Fake news detection: a deep learning approach," SMU Data Science Review, vol. 1, no. 3, p. 10, 2018.

[17] Y. Liu and Y.-F. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[18] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu, "Ti-cnn: Convolutional neural networks for fake news detection," arXiv preprint arXiv:1806.00749, 2018.

[19] H. Q. Abonizio, J. I. de Morais, G. M. Tavares, and S. Barbon Junior, "Language-independent fake news detection: English, portuguese, and spanish mutual features," Future Internet, vol. 12, no. 5, p. 87, 2020.

[20] K. Martínez-Gallego, A. M. Álvarez-Ortiz, and J. D. Arias-Londono, "Fake news detection in spanish using deep learning techniques," arXiv preprint arXiv:2110.06461, 2021.

[21] S. Kannan, V. Gurusamy, S. Vijayarani, J. Ilamathi, M. Nithya, S. Kannan, and V. Gurusamy, "Preprocessing techniques for text mining," International Journal of Computer Science & Communication Networks, vol. 5, no. 1, pp. 7–16, 2014.

[22] B. Bengfort, R. Bilbro, and T. Ojeda, Applied text analysis with python: Enabling language-aware data products with machine learning." O'Reilly Media, Inc.", 2018.

[23] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," tech. rep., Cornell University, 1987.

[24] S. Vijayarani, M. J. Ilamathi, M. Nithya, et al., "Preprocessing techniques for text mining-an overview," International Journal of Computer

Science & Communication Networks, vol. 5, no. 1, pp. 7–16, 2015.

[25] S. Kayalvizhi, D. Thenmozhi, and C. Aravindan, "Legal assistance using word embeddings.," in FIRE (Working Notes), pp. 36–39, 2019.

[26] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," applied sciences, vol. 9, no. 20, p. 4396, 2019.

[27] A. I. Schein and L. H. Ungar, "Active learning for logistic regression: an evaluation," Machine Learning, vol. 68, no. 3, pp. 235–265, 2007.

[28] A. Smola and S. Vishwanathan, "Introduction to machine learning," Cambridge University, UK, vol. 32, no. 34, p. 2008, 2008.

[29] L. Bottou et al., "Stochastic gradient learning in neural networks," Proceedings of Neuro-Numes, vol. 91, no. 8, p. 12, 1991.

[30] S. Shalev-Shwartz and S. Ben-David, Understanding machine learning: From theory to algorithms. Cambridge university press, 2014.

[31] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," arXiv preprint arXiv:1810.11363, 2018.

[32] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," Frontiers in neurorobotics, vol. 7, p. 21, 2013.

[33] C. Bentejac, A. Cso¨rgo˝, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," Artificial Intelligence Review, vol. 54, no. 3, pp. 1937–1967, 2021.

[34] L. Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5– 32, 2001.

[35] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in Ensemble machine learning, pp. 157–175, Springer, 2012.

[36] I. Muhammad and Z. Yan, "Supervised machine learning approaches: A survey.," ICTACT Journal on Soft Computing, vol. 5, no. 3, 2015.

[37] I. Devi, G. Karpagam, and B. V. Kumar, "A survey of machine learning techniques," International Journal of Computational Systems Engineering, vol. 3, no. 4, pp. 203–212, 2017.

[38] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.

[39] A. Gholamy, V. Kreinovich, and O. Kosheleva, "Why 70/30 or 80/20 relation between training and testing sets: a pedagogical explanation," 2018.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.