https://doi.org/10.14483/23448393.21846

Research paper

# Community-Based Early Warning System Model for Stream Overflow in Barranquilla

## Modelo de sistema de alerta temprana para desbordamiento de arroyos en barranquilla basado en la comunidad

**Iván Andrés Felipe Serna-Galeano**[1], **Ernesto Gómez-Vargas**[1], **and Julián Rolando Camargo-López**[1]

[1]Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

Correspondence: jcamargo@udistrital.edu.co

Received**:** February 14th, 2024
Modified: March 17th, 2024
Accepted: April 18th, 2024

## Abstract

**Context**: This work aims to design and create a community-based early warning model as an alternative for the mitigation of disasters caused by stream overflow in Barranquilla (Colombia). This model is based on contributions from social networks, which are consulted through their API and filtered according to their location.

**Methods**: With the information collected, cleaning and debugging are performed. Then, through natural language processing techniques, the texts are tokenized and vectorized, aiming to find the vector similarity between the processed texts and thus generating a classification.

**Results**: The texts classified as dealing with stream overflow are processed again to obtain a location or assign a default one, in order to for them to be georeferenced in a map that allows associating the risk zone and visualizing it in a web application to monitor and reduce the potential damage to the population.

**Conclusions**: Three classification algorithms were selected (random forest, extra trees, and k-neighbors) to determine the best classifier. These three algorithms exhibited the best performance and R2 regarding the data processed in the regressions. These algorithms were trained, with the k-neighbor algorithm exhibiting the best performance.

**Keywords**: stream overflow, social network, machine learning, natural language processing

## Resumen

**Contexto:** Este trabajo tiene como objetivo diseñar y crear un modelo de alerta temprana basado en la comunidad como alternativa para la mitigación de desastres causados por el desbordamiento de arroyos en Barranquilla (Colombia). Este modelo se basa en contribuciones de redes sociales, que se consultan a través de su API y se filtran según su ubicación.

**Métodos:** Con la información recogida, se realiza una limpieza y depuración. Luego, mediante técnicas de procesamiento de lenguaje natural, los textos se tokenizan y vectorizan, buscando encontrar la similitud vectorial entre los textos procesados y así generar una clasificación.

**Resultados:** Los textos clasificados como relacionados con el desbordamiento de arroyos se procesan nuevamente para obtener una ubicación o asignar una por defecto, con el fin de georreferenciarlos en un mapa que permita asociar la zona de riesgo y visualizarla en una aplicación web, en aras de monitorear y reducir el daño potencial a la población.

**Conclusiones:** Se seleccionaron tres algoritmos de clasificación (bosque aleatorio, árboles extra y k-vecinos) para determinar el mejor clasificador. Estos tres algoritmos mostraron el mejor rendimiento y R2 con respecto a los datos procesados en las regresiones. Estos algoritmos fueron entrenados, y se encontró que el algoritmo k-vecinos tuvo el mejor rendimiento.

***Palabras clave:*** arroyos, redes sociales, aprendizaje automático, procesamiento de lenguaje natural

https://doi.org/10.14483/23448393.21846

## 1. Introduction

Floods are natural events produced by excess water from rivers in areas that have been invaded under normally dry conditions [1]. The city of Barranquilla (Colombia) exhibits a severe case of overflowing streams, rivers, and creeks that cross or border the city, causing flooding in urban areas, which causes material damage, and even human losses in some cases. This issue is caused by various factors, such as the city's proximity to the tributary of the Magdalena River and the sea [2], as well as its low topographic slope (around 5%) [3]. In addition, social and sanitary problems associated with waste management and poor planning cause the rainwater system to collapse quickly [4]. This makes it essential to deal with emergencies during rainy seasons [2]. One way to mitigate the adverse effects of these climatic events is to design and create an early warning model that allows monitoring and alerting affected communities to reduce the impact of overflows, as detailed in [5]. Currently, there are several warning and tracking methods. For example, in Barranquilla, an early warning system was created which uses sensors and updates information via a web application [4]. This type of system has also been employed to supply the city with solar energy [5]. Moreover, hydrological and hydraulic models have been developed to predict the areas where the flow is high and fast, in order to issue early warnings to the community [6]. On the other hand, Barranquilla has started to monitor atmospheric phenomena, making use of radars that generate alerts before an imminent storm or rain [7].

As for data collected from social networks, in Japan, a system was created which obtains information from Twitter and processes it to generate Tsunami alerts [8]. Likewise, in the departmental risk plan of the department of Atlántico (Colombia), this risk scenario has been identified along with its corresponding background [9].

In light of the above, it is critical to find an additional solution to those proposed so far. A model with input information from social networks will be advantageous and increasingly necessary, as this type of information stands out for being fast and up-to-date. Additionally, it enables active monitoring and the generation of alerts to the community.

The first section of this article presents the proposed model flow, along with the designed system structure, including the interaction between an API, a database, and the instances entrusted with processing the information and predicting stream overflow events. The second section outlines the methodology, which includes the data cleaning process and a previous analysis aimed at determining the models that best fit the behavior of the data. The third section describes the information processing tasks and the training of the selected algorithm. Finally, the results of this research are presented, as well as a discussion and the main conclusions.

## 2. Methodology

### 2.1 Model structure

For the model design and structure, a flow verification was initially established, where the precipitation percentage was validated through a meteorological API. If the required percentage was met, the proposed model began by collecting information from the API of the social network X (formerly known as *Twitter*), as described in Fig. 1. The collected data were stored in a specific database and then processed using the trained algorithm selected for the classification. The database was then updated with each classification result (Fig. 2). In cases where the text contained a location (address), this information was extracted to update the database; otherwise, a default address was assigned. Afterwards, the geocoding process began, yielding coordinates (latitude and longitude) and that were assigned to the data. The next step in the proposed model was to identify whether the recorded events occurred in stream areas. To this effect, it was necessary to make a spatial crossing using a geoprocess, which sought to intercept the location of the events detected as streams *vs.* the polygons where they circulate. A record of streams was established as roads in Barranquilla (Colombia). The result of the geoprocess was the frequency of events in each stream polygon and the location of each river stream event. This information was displayed on a map (*i.e.*, a web map) along with the record of the events found in order to generate alerts to the population in risk areas.

https://doi.org/10.14483/23448393.21846

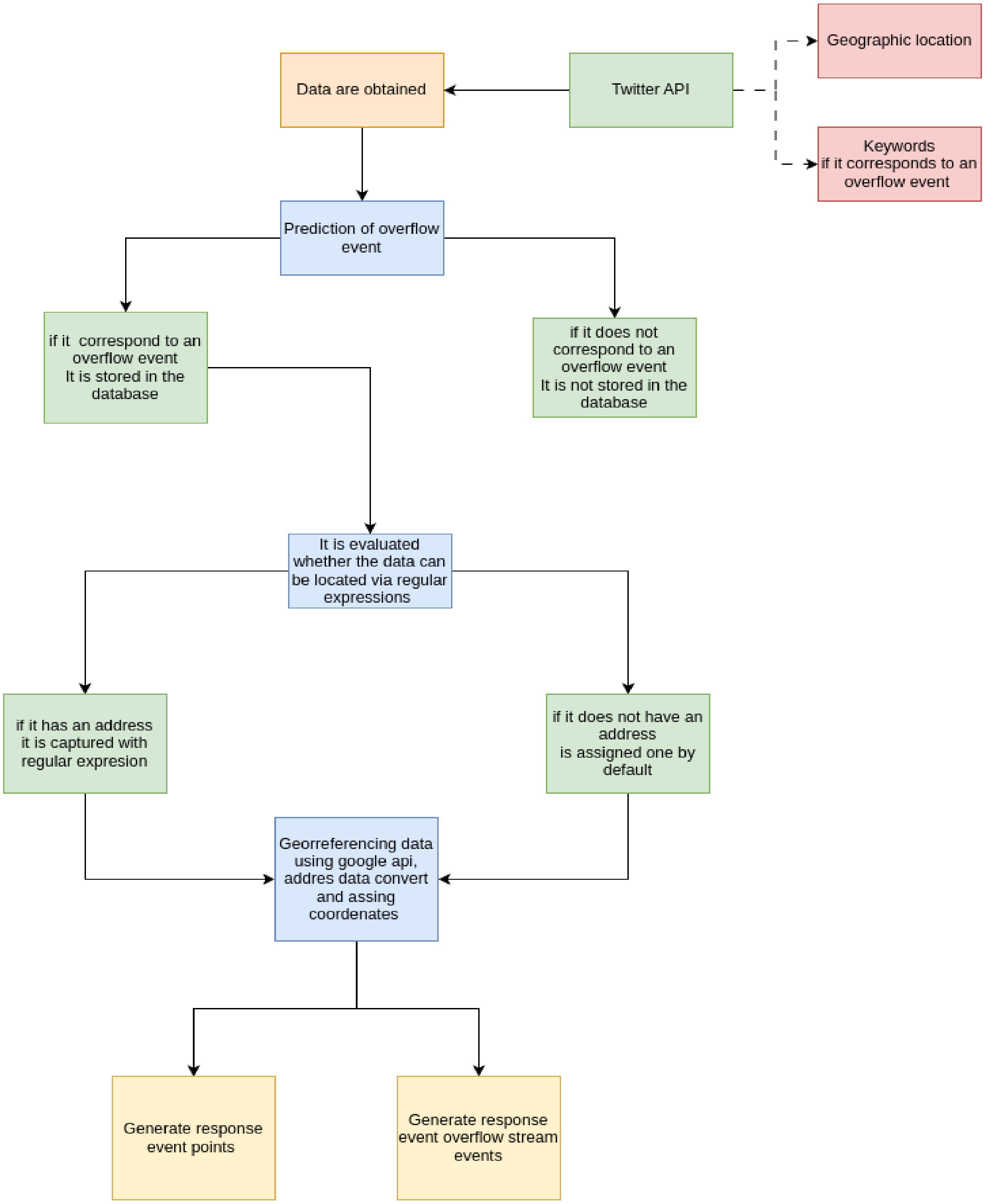
Geographic location
Data are obtained
Twitter API
Keywords
if it corresponds to an overflow event
Prediction of overflow event
if it correspond to an overflow event
It is stored in the database
if it does not correspond to an overflow event
It is not stored in the database
It is evaluated whether the data can be located via regular expressions
if it has an address
it is captured with regular expresion
if it does not have an address
is assigned one by default
Georreferencing data using google api, addres data convert and assing coordenates
Generate response event points
Generate response event overflow stream events

**Figure 1.** Model flow

https://doi.org/10.14483/23448393.21846

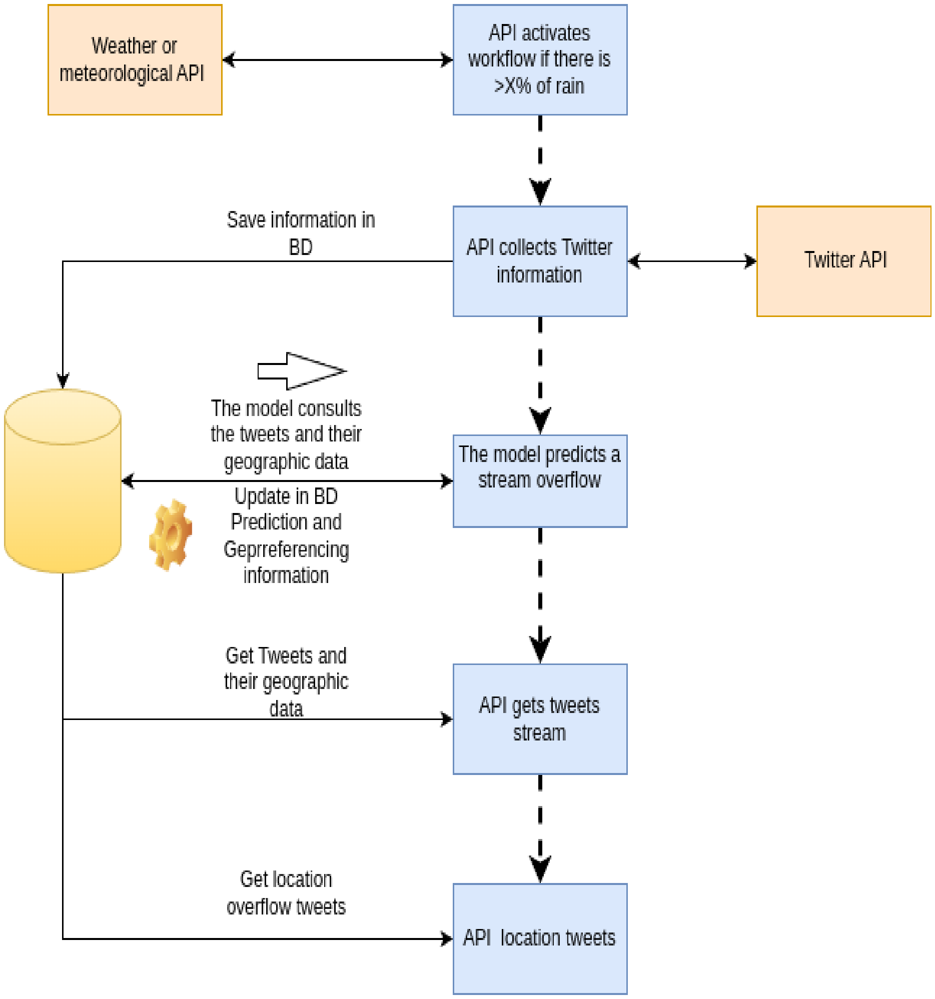
Weather or meteorological API
API activates workflow if there is >X% of rain
Save information in BD
API collects Twitter information
Twitter API
The model consults the tweets and their geographic data
The model predicts a stream overflow
Update in BD Prediction and Geprreferencing information
Get Tweets and their geographic data
API gets tweets stream
Get location overflow tweets
API location tweets

**Figure 2.** Processing flow

## 2.2 Data collection

The information needed to feed the database was obtained by creating a research account in the X social network (formerly known as *Twitter*) [10]. Initially, a search was performed using the X API. The search was filtered by keywords, which were selected based on information from publications such as news and informative posts on the Internet. The words used to perform the search were the following: *arroyo*, *emergencias*, *arroyos Barranquilla*, *arroyos en la calle*, *arroyo en la carrera*, *creciente*, *reporte lluvias*, *calle*, and *carrera*. This was done in a time window from 2006 to December 2022. Once the best keywords were selected, a filter by location was added, taking the city of Barranquilla as the center and adding a 10 km radius to generate the coverage area.

## 2.3 Data cleaning

Data cleaning is a process that consists of correcting and removing incorrect or duplicated information through computerized methods, as indicated in [11]. In this case, the collected data were stored in a SQL database, for which the Postgresql database engine was used, and the collected information was structured in database tables. For the data cleaning process, the Python programming language was used, as well as the Spacy, Numpy, and Skylearn libraries, which feature functionalities that allow connecting the programming language with the database, obtaining each data stored in the database, and performing data cleaning. In the data cleaning process, the methodology mentioned in [12] and [13] was followed, evaluating the quality of the data. Then, the methodology was replicated, performing a grammatical analysis of the texts contained in each data. Data with special characters, links, and emojis were found and removed using the libraries (data normalization). Finally, the duplicated data found in the database were removed.

The debugging performed on the database made it possible to provide consistency to the information, thus reducing the possible errors generated during training.

### 2.4 Classifying the collected data

Each tweet captured in the search was labeled in three categories set as a classification criteria, *i.e.*, an *event* column containing Y (if the tweet mentions a stream) or N (if it does not mention a stream), referring to whether the tweet indicates stream event risk; a *sarcasm* column containing Y (if it contains sarcasm) or N (if it does not contain sarcasm), referring to phrases with a mocking tone that seek to say the opposite; a *location* column containing Y (if the tweet has an address) or N (if it has no address), referring to the geographic location detected. During manual classification, each tweet was discriminated by a user in charge of labeling each data in its corresponding categories. During this process, the tweet's semantics and context were taken as criteria. This was done manually (*i.e.*, a supervised process). It is important to note that, since the initial training depends on human interaction, there may bias related to the criteria of the person performing the initial classification.

### 2.5 Information processing

#### 2.5.1 Exploratory data analysis and preprocessing

The exploratory analysis consisted of conducting a series of initial studies and tests necessary to obtain basic approximations to the processing of the data [14], using the information previously stored in the database, which had been previously cleaned and normalized. As in data cleaning, the exploratory analysis was carried out using the Python programming language and the Numpy [15], Scikit learn [16], Spacy [17], and Nltk [18] libraries. Then, we counted the number of times that words were repeated (frequency) in the database. Another critical step was filtering the collected data with the one containing the event label equivalent to Y. In this way, the words with the highest frequency during a stream event were found. This information also allowed elaborating histograms and aided in data classification.

Additionally, in the exploratory analysis, the aim was to generate graphs and statistics that would allow determining the behavior of each variable [14], thus obtaining the regression models with the best behavior regarding data trends and behavior [19]. Given the above, the pre-trained algorithm *es_core_news_lg* from the Spacy library [17] was used. It should be noted that the pre-trained algorithm has an accuracy of 100% in tokenization, 99% in parts of speech, and 98% in morphological analysis [17]. Consequently, we vectorized the data to process vectors and not words. We performed mathematical operations on the vectors, as well as regressions, comparing the different statistical models and their behavior. We verified and chose the statistical models with an R2 closer to 1 and the lowest root mean square errors (RMSE).

#### 2.4.2 Training

For the training process, the information was standardized by changing the values of the labels for each data to 1 and 0, with the former corresponding to Y and the latter to N (No). The information obtained was exported in CSV format to train the cleaned data. This copy was made so as not to manipulate the information initially collected and to make it easier to read the corpus file (input data for processing). Considering the data structure and pre-processing shown above, we proceeded to generate a matrix from the corpus. We used the embedding method (vectorization method), which consists of converting the words or sentences (linguistic units) into vectors. To this effect, *es_core_news_lg* was employed, aiming to generate, for the entire corpus, an array of vectors equivalent to a matrix. With the vectorized dataset, this array was processed using the Split method, leaving 70% of the total vectorized data for training and 30% for testing.

After selecting the algorithms that showed the best behavior in the regressions performed for one, two, and three variables (event, sarcasm, and location), they were trained with the information previously divided, classified, and vectorized, thereby obtaining a trained classification machine learning algorithm, which was saved in PKL format and loaded in Python. Different tests were conducted using 30% of the divided data, while the training was tested using the Scikit learn library [16].

https://doi.org/10.14483/23448393.21846

## 3. Results

We obtained 63 259 data. Table 1 shows the data found per year and keyword, where an exponential increase in the amount of information related to streams is observed. Moreover, words such as *arroyo* or *lluvia* have a greater number of coincidences with respect to the other keywords.

**Table I**

**Tweets by year**

| Year | Stream on the avenue | Stream on the street | Streams | Flood | Stream emergency | Rain | *La Felicidad* Stream | Stream | Barranquilla Streams | Country Stream | Streams in Barranquilla | 40th Avenue |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2006 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2007 | 0 | 2 | 6 | 36 | 0 | 209 | 0 | 999 | 0 | 0 | 0 | 5 |
| 2008 | 0 | 1 | 33 | 130 | 0 | 499 | 0 | 500 | 0 | 0 | 0 | 21 |
| 2009 | 0 | 43 | 486 | 494 | 0 | 497 | 0 | 492 | 13 | 0 | 7 | 207 |
| 2010 | 16 | 73 | 489 | 488 | 7 | 490 | 19 | 496 | 492 | 3 | 364 | 496 |
| 2011 | 36 | 493 | 495 | 494 | 14 | 496 | 52 | 495 | 494 | 69 | 492 | 494 |
| 2012 | 112 | 489 | 480 | 471 | 26 | 494 | 124 | 484 | 494 | 28 | 496 | 497 |
| 2013 | 92 | 496 | 484 | 477 | 55 | 487 | 134 | 490 | 495 | 39 | 495 | 498 |
| 2014 | 122 | 484 | 491 | 479 | 35 | 491 | 56 | 478 | 478 | 18 | 494 | 497 |
| 2015 | 187 | 480 | 493 | 488 | 62 | 494 | 174 | 450 | 474 | 65 | 475 | 494 |
| 2016 | 459 | 496 | 481 | 493 | 117 | 490 | 492 | 491 | 495 | 127 | 491 | 488 |
| 2017 | 294 | 475 | 494 | 481 | 68 | 479 | 364 | 477 | 490 | 43 | 490 | 496 |
| 2018 | 488 | 457 | 481 | 495 | 208 | 483 | 404 | 435 | 493 | 71 | 495 | 475 |
| 2019 | 440 | 480 | 462 | 477 | 493 | 489 | 64 | 488 | 485 | 24 | 486 | 496 |
| 2020 | 306 | 493 | 496 | 490 | 252 | 442 | 297 | 496 | 465 | 44 | 465 | 470 |
| 2021 | 400 | 500 | 499 | 495 | 101 | 495 | 352 | 496 | 496 | 22 | 496 | 499 |
| 2022 | 498 | 497 | 500 | 499 | 45 | 500 | 186 | 499 | 493 | 29 | 494 | 497 |
| **Total** | **3450** | **5959** | **6870** | **6987** | **1483** | **7535** | **2718** | **7766** | **6357** | **582** | **6240** | **6630** |

Note: The data were retrieved between 2006 and 2022

During data cleaning, it was found that the initial search criteria (streams, flood, and overflow) obtained similar results to other previously applied criteria, generating duplicated information in the database and implying a considerable increase in information cleaning times. Another critical factor in this section was the removal of special characters and stopwords to obtain texts that could be analyzed and trained (corpus). This reduced the volume of initially obtained information by about 40%, going from 63 259 tweets to 36 720. Additionally, after manual classification, it was found that 8600 were related to a stream event in the city of Barranquilla. Out of these, 1600 had a text that could be associated with an address and could thus be used for geo-coding.

After comparing the different models tested, we noted that the Random Forest model had the best data adjustment; after being adjusted, this regression model exhibited an R2 of 7899, as shown in Table 2, which presents the mean absolute error (MAE), the mean square error (MSE), the RMSE, the coefficient of determination (R2), the root mean log error (RMSLE), and the mean absolute percentage error (MAPE). Additionally, note that this adjusted model exhibits an RMSE of 0.1795, confirming that it has the best statistics for the required adjustment.

**Table II**
**Data regression and results by statistical model**

| Algorithm | MAE | MSE | RMSE | R2 | RMSLE | MAPE |
|---|---|---|---|---|---|---|
| Random Forest | 0,0489 | 0.0323 | 0.1795 | 0.7899 | 0.1257 | 0.1306 |
| Extreme Gradient | 0.0706 | 0.0328 | 0.1807 | 0.7867 | 0.1251 | 0.1951 |
| Extra Trees | 0.0476 | 0.0363 | 0.1903 | 0.7639 | 0.1330 | 0.1255 |
| Light Gradient | 0.0816 | 0.0384 | 0.1959 | 0.7498 | 0.1363 | 0.2221 |
| K Neighbors | 0.0693 | 0.0406 | 0.2015 | 0.7356 | 0.1425 | 0.1743 |
| Decision Tree | 0.0450 | 0.0450 | 0.2119 | 0.7070 | 0.1469 | 0.1213 |
| Gradient Boosting | 0.1197 | 0.0543 | 0.2327 | 0.6474 | 0.1581 | 0.3455 |
| Bayesian Ridge | 0.1821 | 0.0793 | 0.2815 | 0.4844 | 0.1955 | 0.4836 |
| Least Angle | 0.1812 | 0.0794 | 0.2817 | 0.4835 | 0.1953 | 0.4878 |
| Rige | 0.1812 | 0.0795 | 0.2817 | 0.4834 | 0.1953 | 0.4979 |
| AdaBoost | 0.2378 | 0.0996 | 0.3155 | 0.3516 | 0.2385 | 0.3698 |
| Lasso | 0.2998 | 0.1499 | 0.3871 | 0.0247 | 0.2717 | 0.7895 |
| Elastic Net | 0.2998 | 0.1499 | 0.3871 | 0.0247 | 0.2717 | 0.7894 |
| Lasso Least Angle | 0.2998 | 0.1499 | 0.3871 | 0.0247 | 0.2717 | 0.7895 |
| Linear | 0.2994 | 0.1499 | 0.3871 | 0.0246 | 0.2717 | 0.7894 |

*Note*, after performing the corresponding data regressions, the Random Forest statistical model showed a better performance; its R2 was the closest to 1, implying a better fit.

To confirm the fit, the same analysis was validated, considering classification variables such as location and sarcasm, finding similarities in results and the behavior of the data. To this effect, error prediction graphs were also created, as shown in Fig. 3, where the adjusted and predicted errors are denoted, showing no significant variation.

In most cases, the most frequently found words coincided with the initial search criteria, which improved the rate of search and information acquisition during the training of the neural processing algorithm (NPL). Fig. 4 presents a WordCloud showing the words with the most repetitions in the information collected.

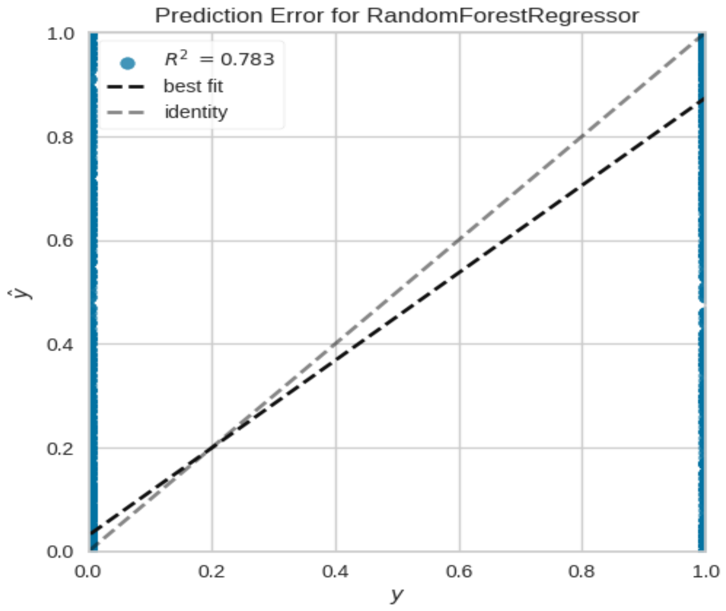

**Figure 3.** Error Prediction with Random Forest Model Fitting (Note, the fitted model does not differ drastically from the identified error).

https://doi.org/10.14483/23448393.21846

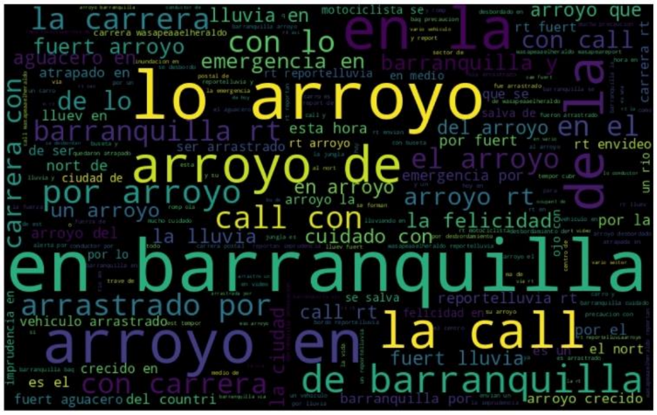

**Figure 4.** WordCloud of words with the highest frequency by geographic area and by keyword

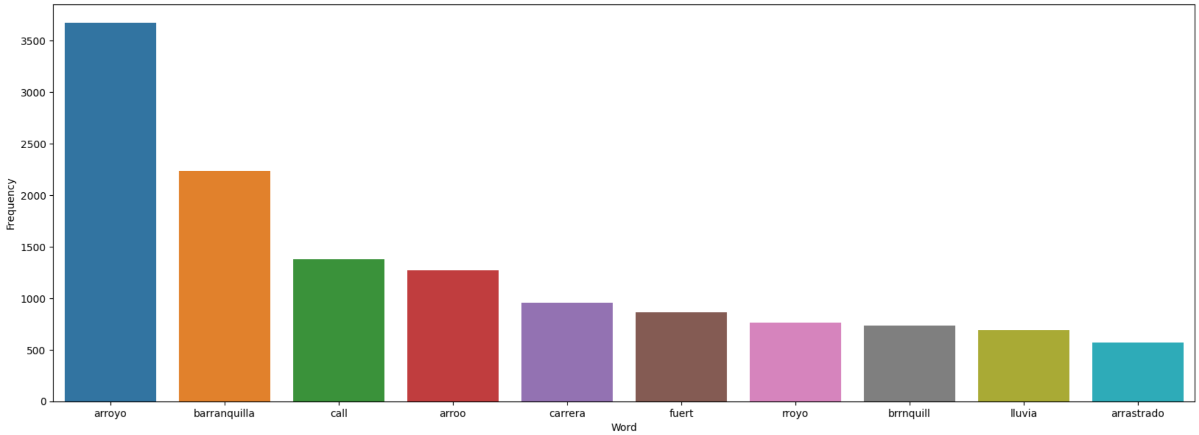

**Figure 5.** Frequency and intensity of word use during a stream (note: there are words that predominantly appear during a stream event, such as stream, Barranquilla, and rainfall report). The units shown above quantity *vs.* item.

The Extra Trees algorithm showed a significant statistical performance, obtaining up to 93.66% accuracy on one classification variable (event). However, it obtained an R2 of 78.99% in the regression. When two variables (event, sarcasm) were analyzed, the algorithm obtained an accuracy of 98.66%, and, with three variables (event, sarcasm, location), it obtained 98.24%. The second best-performing algorithm was the K-Neighbors Classifier, which obtained a 94% accuracy with one variable (event), while the regression showed an R2 close to 73.56%. Under two variables (event, sarcasm), the second algorithm obtained 94.79% accuracy, while, under three variables (event, sarcasm, location), it reported a 93.91% accuracy (Table 3). The two best algorithms were tested using 400 data. The best result was obtained with the K-Neighbors Classifier, which only produced 88 errors out of 400 data.

**Table III**
**Results of testing in the training algorithms**

| Algorithm | Fitted precision | Stream | Sarcasm | Location | Fails |
|---|---|---|---|---|---|
| KNC | 0.9479 | X | X | -- | 88 |
| KNC | 0.9391 | X | X | X | 92 |
| KNC | 0.9400 | X | -- | -- | 94 |
| RF | 0.9822 | X | X | -- | 97 |
| RF | 0.9763 | X | X | X | 104 |
| RF | 0.9771 | X | -- | -- | 102 |
| EXTRA | 0.9366 | X | -- | -- | 125 |
| EXTRA | 0.9866 | X | X | -- | 105 |
| EXTRA | 0.9400 | X | X | X | 125 |

Finally, according to the data collected and processed through the K-Neighbors Classifier algorithm, the process was completed, returning the frequency of tweets for each stream and the location of each tweet related to stream overflows. It was found that the proposed model works with an error rate of 0.22 in the best case, generating an alert to the population according to the location and proximity of these events (Fig. 6).

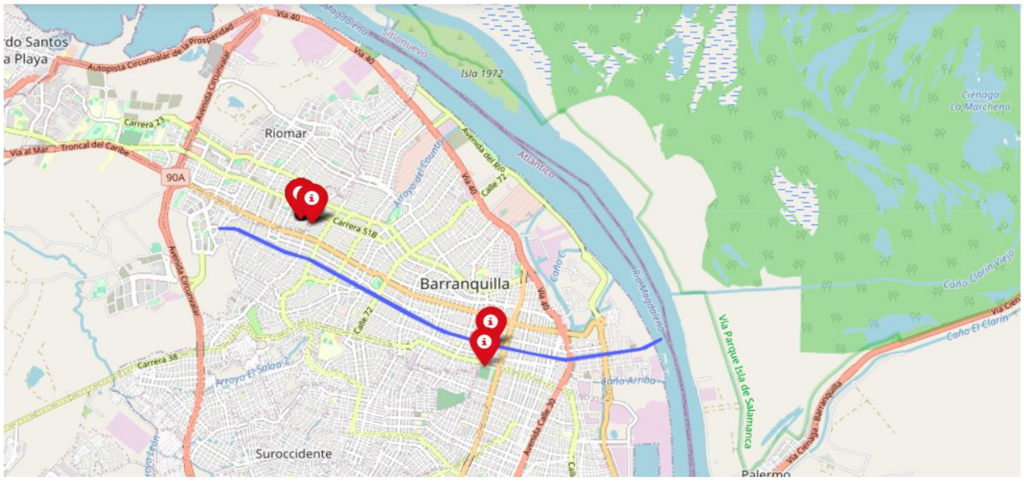

**Figure 6.** Web map generated by the early warning model for stream overflow events in Barranquilla based on community data. The map shows an example of identified overflow events, as they match the location associated with the analyzed Twitter posts.

## 4. Discussion

An early warning model was built from information on stream overflow events in the city of Barranquilla, Colombia, obtained from the X social network. To improve information filtering, a filter by coverage area was added, which allowed obtaining information only from the required area. This search method immensely helped to delimit the collection of required information and reduce the data that generated noise in the cleaning and subsequent training processes. Therefore, data cleaning could be regarded as the whole of the operations performed to eliminate anomalies and obtain an accurate and unique representation [20]. One of the major drawbacks was the high degree of duplicity in the information since, when searching by area of coverage and keywords, we noted that the information contained repeated results. This significantly increased the amount of information in the database. In the same sense, before training the algorithm, it was necessary to label the data obtained. This was done manually since the classification criteria can be subjective.

Some of the most representative diagrams obtained during the exploratory analysis were the bar diagram and scatter, and intensity plots. These allowed to determine the behavior of each variable [14] during the exploratory study and, from the graphs and statistics obtained, to generate initial hypotheses regarding the behavior of each of the tested algorithms, unlike statistical inference, where hypotheses are preliminarily established and subsequently challenged and tested via confirmatory tests [21]. Other exploratory analysis techniques also allow comparing the data distribution by applying a statistical model. Thus, it is possible to determine which statistical model best fits the trend and behavior of the data [19]. In this work, it was possible to obtain different regression graphs and models that enabled the analysis of the behavior of the data and their trends. Likewise, these results allowed identifying the models with the best behavior. However, this preprocessing required a pre-trained algorithm designed for the Spanish language; in this study, an algorithm trained with news in Spanish was used [17].

As for the processing and training of the algorithm, it is critical to highlight the importance of natural language processing, whose objective is the interaction between computers and human language [22]. It can also be indicated that natural language processing is the ability of machines to process information communicated in human language [23], *i.e.*, natural language processing aims to analyze, understand, and generate the language that humans naturally use [24]. Thus, text processing seeks to detect and write the rules that form structural patterns, and then find those patterns in linguistic units such as letters, words, and sentences [22] to be embedded in vectors [24]. Additionally, natural language processing has different levels of complexity, each of which represents a type of analysis to extract specific information at the morphological, lexical, syntactic, semantic, and pragmatic levels [25], [26], [27]. Here, the morphological level corresponds to the composition of words, the linguistic level is related to establishing the individual meaning of each word, the syntactic level deals with the function of each word within a sentence, the semantic level corresponds the meaning of the sentence based on the interaction between words, and the pragmatic level is responsible for analyzing the context of a text [25].

The techniques used in natural language processing are sentence detection, word segmentation and discrimination, grammatical tagging (also known as *parts of speech*, or POS), morphological segmentation, and stopword elimination [28]. Among the best-known applications in the field of natural language processing are content classification and summarization, automatic contextual extraction, sentiment analysis, speech-to-text conversion, and, finally, machine translation [29], [24], [30]. In this regard, the advances in data augmentation and its usefulness in addressing different problems have been reported [31]. This is also useful in text classification through the use of various algorithms [32], with the main challenge being the discrimination of ambiguities [33]. As a result, there are binary answers in the classification.

Natural language processing methods allow identifying data trends, so they are widely used for classification depending on their probability of similarity to a target variable. Among the most used methods to classify texts are K-Neighbors, Logistic Regression, Naive Bayes, SVM, Random Forest [34], and, in some cases, regressions such as the Linear Regression method [35]. Algorithms such as Extra Trees and Random Forest belong to the family of Random Forest methods. This type of algorithm combines the randomness of the subspace and bagging, training multiple decision trees slightly different from the dataset [36]. On the other hand, the K-Neighbors an algorithm is a statistical model that uses Euclidean distance to determine the data closest to those to be classified. Depending on the count, the target data will be classified [37]. In this study, the K-Neighbors classification algorithm (KNC) was selected, since it exhibited the best behavior when analyzing and sorting actual data. In recent years, several research works have focused on this field, which has enabled rapid progress in the subject, to the point that, nowadays, it is possible to find natural language applications in cellphones, as well as virtual assistants or different types of call centers [38], [33] [39].

Finally, this study applied training techniques such as tokenization, which seeks to divide sentences into semantic units; vectorization, which aims to convert the union of semantic units into vectors; and, in some cases, the identification of POS [40], aiming to determine the function of the previously tokenized word in context, *i.e.*, a verb, an adverb, an adjective, a connector, and thus establish its weight within the sentence. This type of preprocessing helps to condition the algorithm's training, so that the necessary coincidences and structures are found, in order to predict the classification variables. With the vectorized information, mathematical operations were performed, seeking vector similarity between the target variable and the text to be classified. Likewise, although the algorithms with the best statistics were analyzed, the ones that showed the best fit were selected, since we observed overfitting in the models, causing unreliable classifications. Given the above, the K-Neighbors algorithm was the one that yielded the best classification results.

## 5. Conclusions

The proposed model enabled the collection of information relevant to the case study, which allowed for the detection and location of possible stream overflow events; by changing the different prioritized filters (location and keywords), it was possible to address a problem unique to the region. On the other hand, an exploratory analysis using regression allowed correctly determining a group of algorithms with good performance, as well as the most frequent words during a stream event. An essential factor to mention was the impact of the *sarcasm* column or variable, as it showed a significant weight during the exploratory analysis and training; when vectorizing the data, the selected text favored the resolution of ambiguities in the tweets collected and filtered. This is due to the semantics of each text analyzed, which are strongly altered by the popular vocabulary of the study area.

## Authors Contributions

All authors contributed equally to the research.

## Acknowledgements

The authors would like to thank the Twitter team for providing a Twitter academic license, which was useful for data search and algorithm training. These processes involved long time intervals, but the collected information constituted a sufficient and necessary input for executing this research project.

https://doi.org/10.14483/23448393.21846

## References

[1] H. D. Van Strahlen Bartel, “Estudio de la problemática de los arroyos urbanos de la cuenca El Rebolo (Barranquilla, Colombia) y propuesta de soluciones,” Master’s thesis, Universitat Politècnica de València, 2017. [Online]. Available http://hdl.handle.net/10251/90068

[2] H. Ávila, “Perspectiva del manejo del drenaje pluvial frente al cambio climático-caso de estudio: ciudad de Barranquilla, Colombia,” *Rev. Ing.*, vol. 36, pp. 54-59, 2012. http://www.scielo.org.co/pdf/ring/n36/n36a11.pdf

[3] J. A. Sepúlveda Ojeda, “Aplicación web para la visualización de sensores del sistema de alertas tempranas de los arroyos de Barranquilla-Colombia,” *Rev. Espacios*, vol. 38, no. 47, p. 17, 2017. http://hdl.handle.net/11323/2024

[4] L. J. Pérez Flórez and J. S. Hernández Miranda, “Diseño del modelo económico energético para un sistema de alerta temprana (MEESAT) para los arroyos de Barranquilla,” undergraduate thesis, Universidad de la Costa, 2015. http://hdl.handle.net/11323/4899

[5] M. Acosta-Coll, F. Ballester-Merelo, and M. Martínez-Peiró, “Early warning system for detection of urban pluvial flooding hazard levels in an ungauged basin,” *Nat. Hazards*, vol. 92, pp. 1237-1265, 2018. https://doi.org/10.1007/s11069-018-3249-4

[6] M. A. Coll, “Sistemas de alerta temprana (SAT) para la reducción del riesgo de inundaciones súbitas y fenómenos atmosféricos en el área metropolitana de Barranquilla,” *Sci. Tech.*, vol. 18, no.º 2, pp. 303-308, 2013. https://revistas.utp.edu.co/index.php/revistaciencia/article/view/8661/5411

[7] A. Chatfield and U. Brajawidagda, “Twitter tsunami early warning network: A social network analysis of Twitter information flows,” in *23rd Australasian Conf. Info. Syst.*, 2012, pp. 1-10. https://core.ac.uk/download/pdf/301388984.pdf

[8] Gobernacion del Atlántico, “Plan departamental de gestión del riesgo Atlántico (Colombia),” 2021. [Online]. Available: http://repositorio.gestiondelriesgo.gov.co/handle/20.500.11762/392?locale-attribute=es

[9] X, “Use Cases, Tutorials, & Documentation,” X Developer Platform. https://developer.twitter.com/en (accessed July 8, 2023).

[10] H. Müller and J. C. Freytag, “Problems, methods, and challenges in comprehensive data cleansing”, 2005. [Online]. Available: https://tarjomefa.com/wp-content/uploads/2015/06/3229-English.pdf

[11] Y. Valdés Hernández and D. Marmol Lacal, “DBAnalyzer 2.0, sistema para analizar bases de datos libre,” undergraduate thesis, Universidad de las Ciencias Informáticas, 2008. https://repositorio.uci.cu/jspui/bitstream/ident/TD_1287_08/1/TD_1287_08.pdf

[12] E. Estoque Cabrera, L. Baró Galán, and M. E. Escobar Pompa, “Implementación de algoritmos para la limpieza de datos,” undergraduate thesis, Universidad de las Ciencias Informáticas, 2015. https://repositorio.uci.cu/jspui/handle/ident/8774

[13] I. Zeroual and A. Lakhouaja, “Data science in light of natural language processing: An overview,” *Procedia Comput. Sci.*, vol. 127, pp. 82-91, 2018. https://doi.org/10.1016/j.procs.2018.01.101

[14] G. D. Buzai and C. A. Baxendale, “Análisis exploratorio de datos espaciales,” *Geogr. Sist. Inf. Geográfica*, no. 1, pp. 1-11, 2009. https://ri.unlu.edu.ar/xmlui/bitstream/handle/rediunlu/702/Buzai_An%C3%A1lisis%20Exploratorio%20de%20Datos%20Espaciales.pdf?sequence=1&isAllowed=y

[15] P. Carranza and J. Fuentealba, “Una introducción al análisis exploratorio de datos por medio de Google Analytics,” *Yupana Rev. Educ. Matemática UNL*, vol. 7, pp. 53-65, 2013. https://doi.org/10.14409/yu.v1i7.4262

[16] Z. Jianqiang and G. Xiaolin, “Comparison research on text pre-processing methods on twitter sentiment analysis,” *IEEE Access*, vol. 5, pp. 2870-2879, 2017. https://doi.org/10.1109/ACCESS.2017.2672677

[17] E. Kouloumpis, T. Wilson, and J. Moore, “Twitter sentiment analysis: The good the bad and the omg!,” in *Proc. Int. AAAI Conf. Web Soc. Media*, 2011, pp. 538-541. https://doi.org/10.1609/icwsm.v5i1.14185

[18] “Spanish · spaCy Models Documentation,” SpaCy. https://spacy.io/models/es (accessed July 16, 2023).

[19] S. Quarteroni, “Natural language processing for industry: ELCA’s experience,” *Inform.-Spektrum*, vol. 41, no. 2, pp. 105-112, 2018. https://doi.org/10.1007/s00287-018-1094-1

[20] A. Yadav, A. Patel, and M. Shah, “A comprehensive review on resolving ambiguities in natural language processing,” *AI Open*, vol. 2, pp. 85-92, Jan. 2021. https://doi.org/10.1016/j.aiopen.2021.05.001

[21] N. Kaur, V. Pushe, and R. Kaur, “Natural language processing interface for synonym,” *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, n.º 7, pp. 638-642, 2014. https://ijcsmc.com/docs/papers/July2014/V3I7201499a13.pdf

[22] M. M. E. Torres and R. Manjarrés-Betancur, “Asistente virtual académico utilizando tecnologías cognitivas de procesamiento de lenguaje natural,” *Rev. Politécnica*, vol. 16, no. 31, pp. 85-96, 2020. https://doi.org/10.33571/rpolitec.v16n31a7

[23] A. Gelbukh, “Procesamiento de lenguaje natural y sus aplicaciones,” *Komputer Sapiens*, vol. 1, pp. 6-11, 2010. https://www.gelbukh.com/CV/Publications/2010/Procesamiento%20de%20lenguaje%20natural%20y%20sus%20aplicaciones.pdf

[24] L. Deng, “Deep learning: from speech recognition to language and multimodal processing,” *APSIPA Trans. Signal Inf. Process.*, vol. 5, no. 1, Jan. 2016. https://doi.org/10.1017/ATSIP.2015.22

[25] F. Ramos and J. Vélez, “Integración de técnicas de procesamiento de lenguaje natural a través de servicios web,” undergraduate thesis, Universidad Nacional del Centro de la Provincia de Buenos Aires, 2016. [Online]. Available: https://www.ridaa.unicen.edu.ar/handle/123456789/644

[26] P. Johri, S. K. Khatri, A. T. Al-Taani, M. Sabharwal, S. Suvanov, and A. Kumar, “Natural language processing: History, evolution, application, and future work,” in *Proc. 3rd Int. Conf. Computing Informatics Networks: ICCIN 2020*, 2021, pp. 365-375. http://dx.doi.org/10.1007/978-981-15-9712-1_31

[27] M. Maldonado, D. Alulema, D. Morocho, and M. Proano, “System for monitoring natural disasters using natural language processing in the social network Twitter,” in *2016 IEEE Int. Carnahan Conf. Sec. Tech. (ICCST)*, 2016, pp. 1-6. https://doi.org/10.1109/CCST.2016.7815686

[28] D. Moreira *et al.*, “Análisis del estado actual de procesamiento de lenguaje natural,” *Rev. Ibérica Sist. Tecnol. Informação*, no. E42, pp. 126-136, 2021. https://dialnet.unirioja.es/servlet/articulo?codigo=8624557

[29] A. Gutiérrez Domínguez, “Aplicación de técnicas de procesamiento de lenguaje natural (NLP) en Twitter para la evaluación de políticas agrarias y del medio rural,” Master’s thesis, 2022. [Online]. Available: http://hdl.handle.net/10251/186767

[30] Z. Zong and C. Hong, “On application of natural language processing in machine translation,” in *2018 3rd Int. Conf. Mech. Control Comp. Eng. (ICMCCE)*, Sep. 2018, pp. 506-510. https://doi.org/10.1109/ICMCCE.2018.00112

[31] B. Li, Y. Hou, and W. Che, “Data augmentation approaches in natural language processing: A survey,” *AI Open*, vol. 3, pp. 71-90, Jan.

2022, https://doi.org/10.1016/j.aiopen.2022.03.001

[32] M. B. Hernández and J. M. Gómez, “Aplicaciones de procesamiento de lenguaje natural,” *Rev. Politécnica*, vol. 32, 2013. https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/article/view/32

[33] A. A. Turdjai and K. Mutijarsa, “Simulation of marketplace customer satisfaction analysis based on machine learning algorithms,” in *2016 Int. Sem. App. Tech. Info. Comm. (ISemantic)*, Aug. 2016, pp. 157-162. https://doi.org/10.1109/ISEMANTIC.2016.7873830

[34] Y. Takefuji and K. Shoji, “Effectiveness of ensemble machine learning over the conventional multivariable linear regression models”. [Online]. Available: https://neuro.musashino-u.ac.jp/publications/pdf/reg_vs_ml.pdf

[35] Y. Al Amrani, M. Lazaar, and K. E. El Kadiri, “Random forest and support vector machine based hybrid approach to sentiment analysis,” *Procedia Comput. Sci.*, vol. 127, pp. 511-520, 2018. https://doi.org/10.1016/j.procs.2018.01.150

[36] D. S. Osorio, J. J. M. Escobar, L. Chanona-Hernández, G. Sidorov, and C. J. Núñez-Prado, “Clasificación bi-clase de canciones infantiles aplicando inteligencia artificial y procesamiento de lenguaje natural,” *Res. Comput. Sci.*, vol. 151, no. 5, pp. 31-38, 2022. https://www.rcs.cic.ipn.mx/2022_151_5/Clasificacion%20bi-clase%20de%20canciones%20infantiles%20aplicando%20inteligencia%20artificial%20y%20procesamiento.pdf

### Iván Andrés Felipe Serna-Galeano

Cadastral and geodetic engineer from Universidad Distrital Francisco José de Caldas and Master's student in Information and Communications Sciences at the Department of Engineering of Universidad Distrital Francisco José de Caldas in Bogotá, Colombia. He can be contacted at iasernag@udistrital.edu.co

### Ernesto Gómez-Vargas

Full professor at the Department of Engineering of Universidad Distrital Francisco José de Caldas in Bogotá, Colombia. Electronics engineer, specialist in Mobile Telecommunications, and master in Teleinformatics from Universidad Distrital Francisco José de Caldas; PhD in Engineering from Pontificia Universidad Javeriana, Bogotá. He can be contacted at egomez@udistrital.edu.co

### Julián Rolando Camargo-López

Full professor at the Department of Engineering of Universidad Distrital Francisco José de Caldas in Bogotá, Colombia. Electronics engineer from Universidad Distrital Francisco José de Caldas, specialist in Design and Construction of Telematic Solutions from Universidad Autónoma de Colombia, and master in Information and Communications Sciences from Universidad Distrital Francisco José de Caldas. He can be contacted at jcamargo@udistrital.edu.co