Classification Model of Emotional Tone in Hate Speech and Its Relationship with Inequality and Gender Stereotypes, Using NLP and Machine Learning Algorithms

Aymé Escobar Díaz , Ricardo Rivadeneira , Walter Fuertes * , Washington Loza

Posted Date: 4 March 2026

doi: 10.20944/preprints202603.0228.v1

Keywords: hate speech; emotional tone; NLP; machine learning; gender violence; content moderation; ensemble models; RoBERTa; mathematical modeling; decision function; L2 regularization; probabilistic classification

 doi:10.20944/preprints202603.0228.v1

Article

Classification Model of Emotional Tone in Hate Speech and Its Relationship with Inequality and Gender Stereotypes, Using NLP and Machine Learning Algorithms

Aymé Escobar Díaz, Ricardo Rivadeneira, Walter Fuertes and Washington Loza

Department of Computer Science, Universidad de las Fuerzas Armadas (ESPE), Av. General Rumiñahui 171103, Quito, Ecuador
* Correspondence: wmfuertes@espe.edu.ec

Abstract

Hate speech on social media reproduces norms of inequality and gender stereotypes, disproportionately affecting women. This study proposes a hybrid approach that integrates emotional tone classification with explicit hostility detection to strengthen preventive moderation. We constructed a corpus from three open datasets (1,236,371 records; 1,003,991 after ETL) and represented the text using TF-IDF and contextual RoBERTa *embeddings*. We trained individual models (RoBERTa *fine-tuned*, Random Forest, and XGBoost) and a stacking metamodel (Gradient Boosting) that combines their probabilities. On the test set, the ensemble outperformed the base classifiers, achieving *accuracy* of 0.93 in hate detection and 0.90 in emotion classification, with an AUC of 0.98 for emotion classification. We implemented a RESTful API and a web client to validate the moderation flow before publication, along with an administration panel for auditing. Performance tests showed viability under moderate loads and concurrency limitations starting at 300 users, associated with deployment via an *Ngrok* tunnel. In general, the results indicate that incorporating emotional tone analysis improves the model's ability to identify implicit hostility and offers a practical way to promote safer digital environments. The probabilistic results obtained by the ensemble model were subsequently analyzed using the Bayesian Calibration and Optimal Design under Asymmetric Risk *(BACON-AR)* framework, which serves as a mathematical post-hoc validation layer to optimize the decision threshold under unequal costs. This framework does not modify the trained architecture but adjusts the estimated probabilities and selects the threshold that minimizes the total expected risk. By combining *TF-IDF* and textit RoBERTa embeddings with a stacked metamodel, the ensemble's decision function was optimized via regularization, improving generalizability and the stability of predictions. The incorporation of the BACON-AR framework strengthened the system's probabilistic consistency, ensuring that final decisions were aligned with the actual consequences of errors under an asymmetric risk scheme.

Keywords: hate speech; emotional tone; NLP; machine learning; gender violence; content moderation; ensemble models; RoBERTa; mathematical modeling; decision function; L2 regularization; probabilistic classification

1. Introduction

Hate speech is a complex phenomenon that undermines social harmony and equality in both physical and digital environments. Its definition involves at least three essential components: expressive behavior, a target group identified by protected characteristics, and the manifestation of negative emotions such as hostility, humiliation, or contempt [1]. The difficulty of establishing a universally accepted definition has been highlighted in recent reviews, which note that the semantic and pragmatic particularities of language introduce ambiguity in its delimitation [2]. The difficulty of establishing a universally accepted definition has been highlighted in recent reviews, which note

 doi:10.20944/preprints202603.0228.v1

that the semantic and pragmatic particularities of language introduce ambiguity into its delimitation. Likewise, comparative studies show that the conceptualization of hate speech varies across legal, cultural, and technological frameworks, reinforcing the need for multidimensional approaches. [3].

The emotional dimension is key to understanding and detecting this phenomenon, as hate speech often draws on an affective background that shapes perception and social interaction. Basic emotions, such as happiness, anger, and fear, directly influence people's social evaluations, even in the absence of contextual information [4]. Although emotions reflect immediate reactions, the emotional tone constitutes a more sustained affective state that modulates the interpretation of messages and their potential to become hate speech [5]. In the field of Natural Language Processing (NLP), the identification of emotions such as anger, contempt, or fear has been shown to improve the accuracy of automated systems in differentiating between negative opinions and discriminatory attacks [6,7].

Gender-based violence represents one of the most critical contexts for hate speech. In Spanish-speaking communities, women are particularly vulnerable to discriminatory attacks in which misogyny is reproduced through hostile expressions and patterns of social normalization [8]. On a global scale, online misogyny is associated with harassment dynamics that reinforce the structural subordination of women [9]; extreme forms of harassment, such as *rapeglish*, used for intimidation purposes have even been documented [10]. The technical challenges are significant: international NLP competitions, such as SemEval, demonstrate that attacks targeting women and migrants are recurring and difficult to detect with standard models [11], with impacts that transcend the digital realm [12].

In this context, Natural Language Processing (NLP) and Machine Learning (ML) are becoming established as fundamental approaches for detecting and mitigating hate speech. Systematic reviews highlight both their advances and limitations: dependence on large volumes of representative data, biases in training sets, and dilemmas regarding fairness and interpretability at scale [13,14]. The most promising results are observed in two families of models: (i) tree-based models (Random Forest, XGBoost) and (ii) transformative architectures, such as BERT and RoBERTa [15,16]. However, a significant gap remains: the explicit integration of emotional tone into the automated moderation process, especially before publication, which is the main motivation for this work.

From a formal perspective, we applied a mathematical model, Bayesian Calibration and Optimal Design Asymmetric Risk (BACON-AR) [17–20], which describes the relationship between textual features and hostility categories using a supervised probabilistic decision function. This model combines TF-IDF vector representations and RoBERTa *contextual embeddings*, integrated into an optimization process with L2 regularization and adaptive weights. The objective was to minimize classification errors and improve system stability, avoiding overfitting. Thus, the model formalizes the learning process from a statistical perspective, allowing us to analyze its convergence and generalizability in real-world hate speech detection scenarios.

From this perspective, the present study strengthens *preventive moderation* on social networks through a hybrid approach that combines the detection of explicit hostility with the classification of emotional tone. The main contributions of this study are as follows:

- The development of a **dual model** that simultaneously classifies hate/non-hate categories and emotional tone, implemented via a **stacking assembly** that integrates RoBERTa, Random Forest (TF-IDF), and XGBoost (embeddings), surpasses the performance of individual classifiers;
- The **construction and enrichment** of an extensive corpus from three open sources, with a mapping between *GoEmotions* and Ekman's emotions to study the interaction between affect and hostility;
- An **end-to-end functional validation** using a RESTful API and a web client with pre-moderation and auditing; and
- A **comparative analysis** that demonstrates that incorporating emotional tone reduces false positives in cases of ambiguity or irony.

These contributions complement studies that warn about bias risks and implementation challenges [14,21], and confirm that a heterogeneous ensemble (transformer plus trees) improves generalization compared to the isolated use of a model such as RoBERTa [15].

 doi:10.20944/preprints202603.0228.v1

The remainder of the manuscript is structured as follows: the Materials and Methods section describes the corpus, preprocessing, feature representation, models, the API, and web client architecture. In this section, we also incorporate the BACON-AR framework for probabilistic calibration and threshold optimization under asymmetric risk. The Results section presents model performance, ROC curves, confusion matrices, and performance tests. The Discussion section analyzes the findings, validates and discusses limitations. Finally, the Conclusions section summarizes the contributions and future research directions.

2. Related Work

We conducted a Standard Literature Review (SLR) following the PRISMA guidelines and the PICOS approach, thus identifying the most relevant studies on the detection of hate speech and the analysis of emotional tone, published in [22]. We searched various recognized scientific databases, filtering by language, year of publication, and research approach. We analyzed 34 primary studies and selected 10 representative works. In addition, we included five recent studies that broaden the analysis to include multimodal approaches and the use of large-scale language models (LLMs).

[23] proposed a multi-label self-training model that combines auxiliary emotional cues to improve the sensitivity of automated systems to implicit hostility on social media. This work demonstrated that negative emotions, such as anger and contempt, strengthen the identification of discriminatory discourse. Ramos et al. [24], for their part, analyzed the evolution of transformer-based models and highlighted the limitations in explainability and bias of current detection systems, underscoring the need to develop more interpretive approaches. Rodriguez et al. [25] presented the FADOHS framework, which integrates sentiment and emotion analysis to classify offensive Facebook posts, finding that combining affective and linguistic cues improves performance over purely lexical methods. Finally, Kaminska et al. [26] proposed a fuzzy-rough k-NN method for the simultaneous identification of hate, irony, and emotion, demonstrating the usefulness of non-transformer-based techniques in contexts with limited data.

In the field of Spanish, [27] developed a model that incorporates linguistic and user features to detect offensive messages, highlighting the relevance of sociocultural context in hate speech classification. Complementarily, [28] introduced the Spanish MTLHateCorpus 2023, which uses a multitask approach to predict speech type, target group, and intensity; this resource facilitates the study of social and gender inequality in digital environments. [29], for his part, proposed a multimodal cross-attention model that combines text and image for hostility detection, demonstrating that integrating visual and semantic information allows for capturing nuances that text alone does not reflect. [30] presented **G-BERT**, a Bengali-trained hate speech detection model that addresses the challenges of languages with limited computational resources. [31] analyzed emotional classification in code-mixed texts (Hinglish). In contrast, [32] designed a multitask model that links politeness and emotion in social interactions, thereby facilitating knowledge transfer across affective tasks.

Among recent studies, Nandi et al. [33] introduced **SAFE-MEME**, a structured reasoning model for detecting hate speech in memes that incorporates emotional attention and semantic relationships between text and images, thereby improving contextual interpretation. Chhabra and Vishwakarma [34] proposed **MHS-STMA**, a transformer-based framework with multilevel attention that simultaneously processes textual and visual modalities, achieving greater accuracy and robustness in the face of data noise. Complementarily, Chhabra and Vishwakarma [35] demonstrated that multiscale visual attention improves the detection of hostility in images with superimposed text, reaffirming the importance of integrating visual features to address implicit hate speech in multimodal content.

Furthermore, [36] evaluated various large language models (LLMs) for detecting hate speech in real-world settings, analyzing their generalizability and the influence of cultural context on decision-making. The authors found that while LLMs achieve high accuracy in supervised tasks, they tend to replicate social biases and over-identify neutral expressions as hostile. [37] extended this analysis to multilingual contexts with high semantic variability, where the models demonstrated inconsistencies

 doi:10.20944/preprints202603.0228.v1

in transferring hostility patterns between languages, reinforcing the need for control and calibration mechanisms. Finally, [38] examined the reactive responses of LLMs to offensive content, finding that these models reproduce stereotypes and degrade their performance with ambiguous or ironic texts, highlighting limitations in their reliability and algorithmic fairness.

Complementing this evolution towards hybrid and multimodal architectures, two recent systematic reviews consolidate the theoretical foundation of the present research. On the one hand, [39] conducted a literature review of hate speech detection on online social platforms using Machine Learning and Natural Language Processing, identifying the main techniques, datasets, and challenges in this field. On the other hand, we conducted a systematic review of emotional tone detection in hate speech, published in [22]. This latter study not only maps the current methods and challenges but also underscores the research opportunity addressed by the present work: the explicit integration of the emotional component to improve the accuracy and contextual understanding of automated moderation systems.

In summary, the analyzed works demonstrate an evolution towards hybrid and multimodal architectures, although most focus on English-language general hostility detection. Given these limitations, the present study proposes a complementary approach. It uses an English corpus with over one million records, labeled according to Ekman's six basic emotions. The model combines hate speech detection and emotional tone classification using a stacking scheme composed of three configurations: RoBERTa as the base model, XGBoost with TF-IDF features, and a Random Forest with RoBERTa-generated embeddings. This approach contrasts contextual and statistical representations to improve the accuracy and interpretability of the results. Furthermore, the implementation in a RESTful API and a web client demonstrates its applicability in automated moderation processes and large-scale content analysis.

Taken together, the reviewed studies support incorporating the emotional component into hate detection. The proposed approach contributes to this line of research by integrating natural language processing, emotions, and gender inequality, aiming to develop more accurate and socially relevant automated moderation systems.

Several recent studies have provided fundamental theoretical and mathematical foundations for detecting hate speech using probabilistic models. In particular, [40] and [41] analyze classifier calibration and probabilistic prediction evaluation, highlighting the importance of adjusting model reliability via loss functions and decision thresholds. Complementarily, [17] and [18] propose Bayesian frameworks that allow for representing uncertainty in predictions and improving the stability of deep learning systems under variable conditions.

Furthermore, [42] and [43] delve deeper into cost-sensitive learning, introducing formulations that optimize the hazard function in scenarios with unbalanced classes or decisions that involve different penalty levels. These perspectives, along with the regularization and error control strategies proposed by [19] and [20], provide the mathematical foundation for the approach used in this study.

The proposed model builds on these contributions by integrating a statistical calibration process into a hybrid classification framework based on trees and transformers. This formulation combines vector representations of the text **(TF-IDF and contextual embeddings)** with calibrated probabilistic estimates, seeking to optimize the accuracy, consistency, and reliability of the hostility and emotional tone detection system.

3. Materials and Methods

3.1. Dataset

We created a dataset comprising 1 236 371 records by combining three open-access `.csv` datasets from **Kaggle** to detect hate speech on social media. Each dataset contained the comment and the binary detection of hate/non-hate, as detailed in Table 1.

 doi:10.20944/preprints202603.0228.v1

Table 1. Datasets used for the training and evaluation of the models.

Dataset	Number of Records
Hate Speech Detection curated Dataset	417 561
Detection of implicit beliefs of gender violence against women on social networks using NLP algorithms and machine learning classification techniques	510 254
HateXplain	308 556

Next, we combined the three datasets to obtain the final dataset. On this corpus, we applied a pre-trained model RoBERTa-base-go_emotions, to map Ekman's six basic emotions (i.e., anger, fear, sadness, surprise, disgust, and joy) plus the emotion of neutral to each comment, as shown in Table 2; this resulted in a dataset with three labels: comment, hate speech detection, and emotion classification.

Table 2. Mapping of GoEmotions emotions to Ekman's basic emotions.

Ekman Emotion	Associated GoEmotions Emotions
Neutral	admiration, amusement, approval, love, gratitude, pride, neutral
Joy	caring, desire, excitement, optimism, relief
Anger	annoyance, anger
Disgust	disapproval, disgust, remorse
Fear	fear, nervousness
Sadness	disappointment, embarrassment, grief, sadness
Surprise	confusion, realization, curiosity, surprise

3.2. Preprocessing - ETL

For the extraction, transformation, and loading process, we applied the following steps: removal of duplicates and null values, character normalization to convert comments to lowercase, removal of non-textual characters (i.e., emojis, hashtags, and symbols), removal of stopwords, and lemmatization to reduce words to their base form. After completing the process, we obtained 1 003 991 clean records, which we used for training and validation.

3.3. Exploratory Data Analysis - EDA

Using the clean dataset, we conducted exploratory data analysis to identify trends, patterns, label distributions, and potential biases. In this case, we evaluated the proportion of records classified as hate/non-hate, finding a balance: 502 159 records in the non-hate class and 501 832 in the hate class, as shown in Figure 1.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 March 2026 doi:10.20944/preprints202603.0228.v1

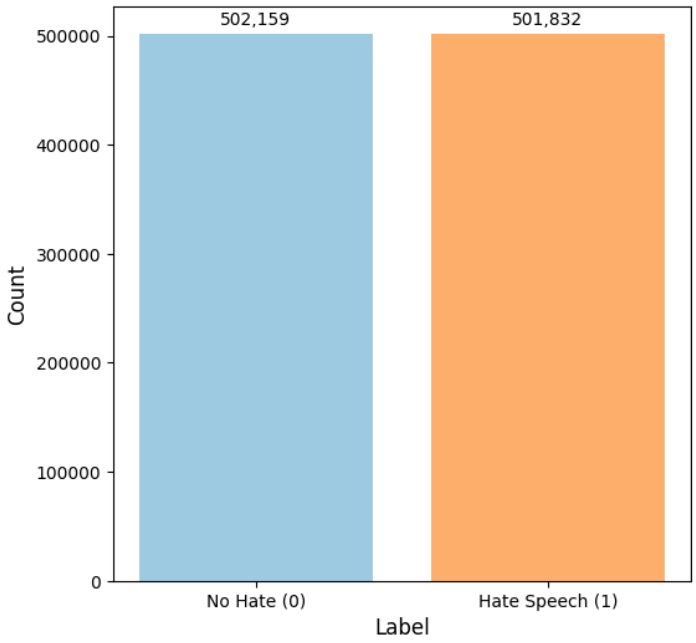

Figure 1. Distribution of hate/non-hate labels in the dataset.

We also found a bias in the distribution of emotions in the dataset, as shown in Figure 2. We observed a predominance of emotions such as anger (564 191), disgust (207 469), and joy (137 686), while the least represented were surprise (14 637), fear (21 577), and neutral (24 345), resulting in an imbalance between the different classes, which poses a challenge in the model training process.

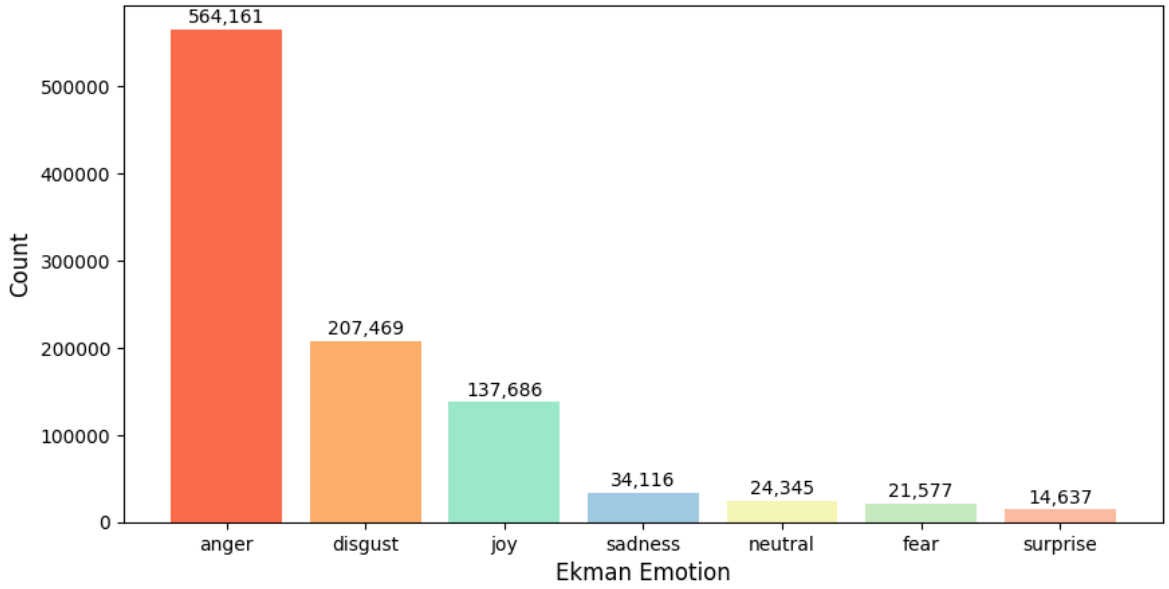

Figure 2. Distribution of Ekman's basic emotions in the dataset.

3.4. Feature Representation

For the representation of textual features, we apply two approaches:

- Contextual Embeddings: generated using the pretrained model cardiffnlp/twitter-roberta-base, which transforms each comment into a 768-dimensional vector that captures semantic and contextual information from the text.
- TF-IDF (Term Frequency–Inverse Document Frequency): a technique for evaluating the importance of each word within the corpus, to obtain a representation based on normalized term frequencies [44].

 doi:10.20944/preprints202603.0228.v1

3.5. Selection of NLP and ML Models

We implemented and compared traditional and modern NLP models. Within the NLP model, we selected the **RoBERTa** model because it is commonly used for text classification, natural language understanding, and emotion analysis. RoBERTa achieves better results than BERT [45], as noted by Liu et al. [15]. RoBERTa eliminates the Next-Sence Prediction task, introduces dynamic masking, and trains on a larger corpus.

For traditional machine learning models, we selected **Random Forest** and **XGBoost** to compare bagging and boosting. The random forest combines multiple decision trees using majority voting. At the same time, XGBoost implements a gradient booster designed to improve the efficiency, computational speed, and performance of the model by combining the capabilities of the XGBoost software and hardware [46].

3.6. Model Training

In this section, we develop the model training process for both classifying emotional tone and detecting hate speech.

3.6.1. Training RoBERTa for Hate Speech

The following four phases were used to train the RoBERTa-base model: data preparation, dataset construction, model initialization, and training-evaluation. The training process is summarized in the Algorithm 1.

Algorithm 1: Training of the RoBERTa model for binary hate speech detection

Input: Clean dataset $D = \{content, label\}$
Output: Trained model M for two classes (hate / non-hate)

1. Split D into training (80%) and test (20%) subsets using a random seed;
2. Tokenize texts with `RobertaTokenizerFast` (max. 128 tokens, truncation, padding);
3. Build datasets and organize them into batches of size 32 using `DataLoader`;
4. Initialize the `RoBERTa-base` model with `RobertaForSequenceClassification` (2 output neurons);
5. Assign the model to the device: GPU (CUDA) or CPU;
6. Define the optimizer `AdamW` ($lr = 2e-5$) and the loss function `CrossEntropyLoss`;
7. **for** *epoch = 1 to 3* **do**
    - **for** *each batch B in training* **do**
        - a. Compute predictions (*logits*) via *forward pass*;
        - b. Compute loss with respect to true labels;
        - c. Backpropagate the error (*backward pass*);
        - d. Update model parameters using the optimizer;
8. Set the model to evaluation mode (disable *dropout*);
9. Evaluate on the test set and compute metrics: accuracy, recall, F1-score;

In the first stage, we imported the clean dataset in `.csv` format, using the content (comments) and label (binary hate/non-hate labels) columns. Then, we divided the dataset into training (80 %) and test (20 %) subsets using a random seed. The texts were tokenized with RobertaTokenizerFast, truncating to a maximum of 128 tokens and using automatic padding to equalize the length of the shorter texts.

Using the tokenized data, we defined a custom TextDataset class, which organizes the text and labels into a PyTorch-compatible structure. In this class, we organized the tensors generated by the tokenizer (input_ids and attention_mask). Training and test datasets were built, organized into batches of 32 instances using DataLoader. The pre-trained RoBERTa-Base model was initialized using the RoBERTaForSequenceClassification class, configured for binary classification. Before starting the training process, the available device for resource allocation was detected as either a GPU (CUDA) or a CPU.

 doi:10.20944/preprints202603.0228.v1

In training, we used the AdamW optimizer with a learning rate of 2×10^{-5} and the CrossEntropyLoss function. Training was performed over three epochs. In each iteration, we applied a forward pass to each batch to obtain model predictions in the form of logits, computed the loss, and performed a backward pass to update the parameters with the optimizer. Finally, we put the model into evaluation mode. Predictions and performance metric reports: accuracy, recall, and F1 score were generated on the test set.

3.6.2. Training Random Forest with TF-IDF for Hate Speech

We trained the Random Forest model using the content (comments) and label (binary) columns of the dataset. We divided the records into training (80 %) and test (20 %) subsets using stratified sampling with random seeding to preserve the class proportions.

As mentioned above, we used the TF-IDF technique, limiting the vocabulary to 10 000 terms and incorporating combinations of unigrams and bigrams.

Regarding the model configuration, we established the following hyperparameters: (i) number of trees = 100, (ii) size of feature subset per tree $\approx \sqrt{10\,000} \approx 100$, (iii) minimum node parameters by default (2 samples for splitting and 1 sample for leaf), and (iv) automatic class load balancing (class_weight="balanced") to avoid bias towards the majority class.

During training, each tree was independently fitted to a random sample of the data and features. The final prediction was obtained by majority vote, where each tree cast its class decision, and the result corresponded to the category with the highest number of votes, as shown in Figure 3.

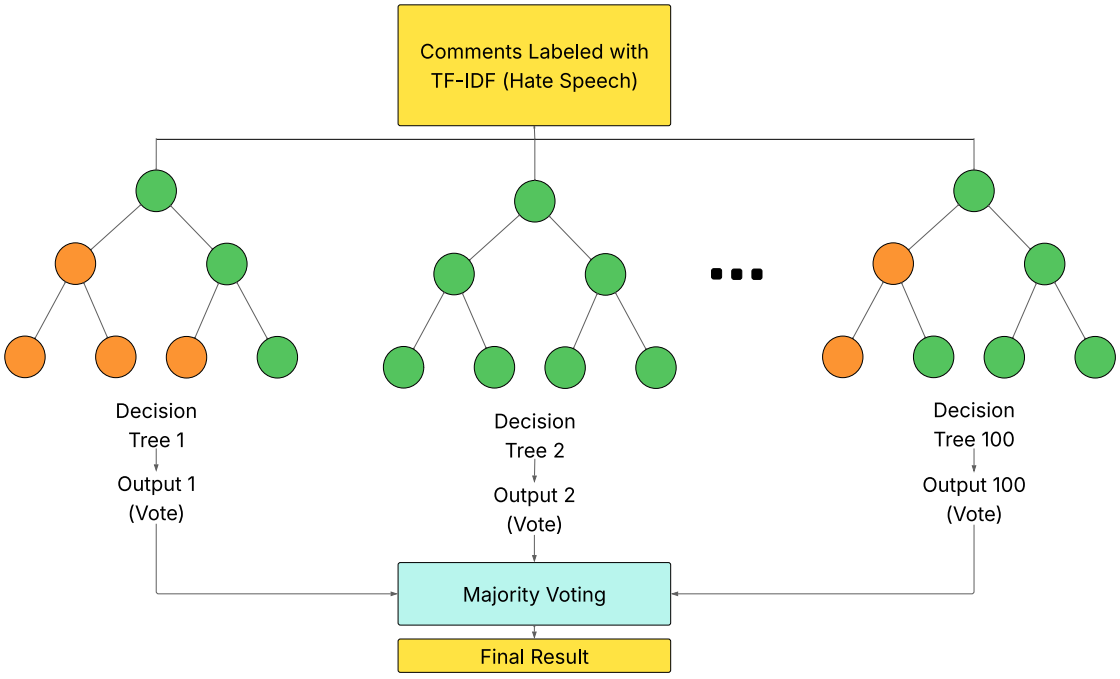

Figure 3. Random Forest – Classification process using TF-IDF.

3.6.3. XGBoost Training with RoBERTa Embeddings for Hate Speech

We trained the model using comments and binary labels from the dataset. We split the records into 80 % training and 20 % test sets using stratified sampling with a random seed.

For feature representation, we used the pre-trained model twitter-roberta-base-sentiment, which generated 768-dimensional dense vectors from the classification token [CLS]. We used the embeddings as input for the XGBoost model.

 doi:10.20944/preprints202603.0228.v1

We configured the model with the following hyperparameters: (i) maximum number of trees = 3000, with early stopping after 100 iterations without improvement; (ii) maximum depth of 12; (iii) learning rate = 0.0007; and (iv) random sampling of 90 % of the features per tree.

The decision trees were added sequentially, correcting errors in previous trees (boosting). We obtained the final prediction from the weighted sum of the outputs of all the trained trees, as shown in Figure 4.

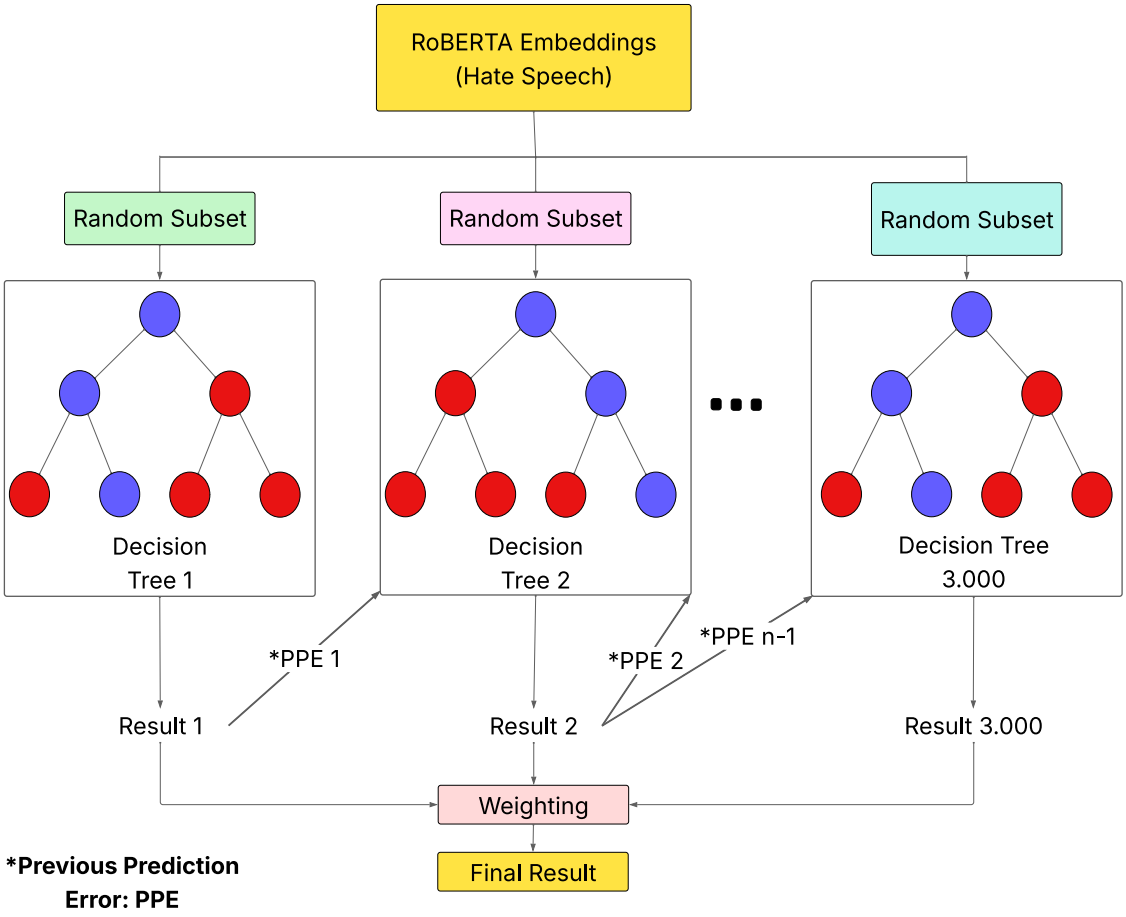

Figure 4. XGBoost – Classification process using embeddings.

3.6.4. Assembled Model for Hate Speech Detection

One of the main contributions of this work was the construction of an assembled model using the stacking technique to overcome the limitations of individual classifiers. Unlike approaches that manually assign weights to each model, the Gradient Boosting metamodel automatically learns the best combination for optimal performance.

Thus, each individual model generated a probability vector for the two problem classes (hate and non-hate): (i) XGBoost worked with RoBERTa embeddings, (ii) Random Forest used TF-IDF vectors, and (iii) RoBERTa fine-tuned directly processed the tokenized text. These probabilities were concatenated horizontally, yielding an input matrix with six columns (two per model), which was used to train the metamodel.

We configured the Gradient Boosting metamodel with 100 trees, a learning rate of 0.1, and a maximum depth of 3. With this strategy, the assembled metamodel achieved higher metrics compared to any of the individual models.

 doi:10.20944/preprints202603.0228.v1

3.6.5. RoBERTa Training for Emotional Tone Classification

We trained the base RoBERTa model for multiclass emotion classification across four phases: label encoding, data preparation, model initialization, and training and evaluation. The training flow is summarized in the Algorithm 2.

Algorithm 2: Training of the RoBERTa model for emotional tone classification

Input: Clean dataset $D = \{content, emotion\_llm\}$
Output: Trained model M for 7 emotional classes

1. Encode labels using `LabelEncoder` $\rightarrow$ numeric values;
2. Split D into training (80%) and test (20%) subsets with stratification and random seed;
3. Compute class weights with `compute_class_weight` to mitigate class imbalance;
4. Tokenize texts with `RobertaTokenizerFast` (max. 128 tokens, truncation, padding);
5. Build datasets and organize them into batches of size 32 using `DataLoader`;
6. Initialize the `RoBERTa-base` model with `RobertaForSequenceClassification` (7 output neurons);
7. Assign the model to the device: GPU (CUDA) or CPU;
8. Define the optimizer `AdamW` ($lr = 2e-5$) and the loss function `CrossEntropyLoss` with class weights;
9. **for** *epoch = 1 to 3* **do**
    for *each batch B in training* **do**
        a. Compute predictions (*logits*) via *forward pass*;
        b. Compute loss with respect to true labels;
        c. Backpropagate the error (*backward pass*);
        d. Update model parameters using the optimizer;
10. Set the model to evaluation mode (disable *dropout*);
11. Evaluate on the test set and compute per-class metrics: accuracy, recall, and F1-score;

In the first stage, we performed supervised training of the RoBERTa-base model on the seven categories corresponding to Ekman's emotions (emoción_llm). We used the content (comments) and emotion classes from the dataset. We transformed the labels into numerical values using LabelEncoder, then split them into training (80 %) and test (20 %) subsets with stratified sampling and random seeding to preserve the original proportions of each emotion.

Given the imbalance in the classes, we calculated specific weights for each emotion using compute_class_weight, which enabled us to assign greater weight to the less frequently occurring minority classes. The text was tokenized with a maximum length of 128 tokens using RobertaTokenizerFast, with automatic truncation and padding, and then organized into batches of 32 instances using DataLoader.

Next, we configured the RobertaForSequenceClassification model with seven neurons in the output layer, corresponding to the number of emotions. We automatically assigned the model to the available computing device, running it on either a GPU (CUDA) or a CPU. For training, we used the AdamW optimizer with a learning rate of 2×10^{-5} and the CrossEntropyLoss function, adjusted with class weights. Training was conducted over three epochs; in each batch, we applied a forward pass, loss calculation, backpropagation, and parameter updates.

Finally, we put the model into evaluation mode to generate predictions on the test set. Performance metrics included accuracy, recall, and F1-score per class to assess the model's ability to distinguish between the different emotions present in hate speech.

3.6.6. Training Random Forest with TF-IDF for Emotional Tone Classification

We trained a Random Forest model for multiclass emotion classification using the content (comments) and emotion_llm (emotions) columns from the dataset. We encoded emotions as numerical values using LabelEncoder, then split the corpus into training (80 %) and test (20 %) subsets using random stratified sampling.

 doi:10.20944/preprints202603.0228.v1

Due to class imbalance, we calculated class-specific weights using compute_sample_weight, assigning higher weights to minority emotions. For feature representation, we transformed the comments into numerical vectors using TF-IDF, limiting the vocabulary to 10 000 terms and using unigrams and bigrams.

We configured the model with the following hyperparameters: (i) 100 decision trees, (ii) a random subset of $\sqrt{10\,000} \approx 100$, (iii) default parameters for minimum nodes (two samples for splits, one for leaves), and (iv) class balancing using calculated weights.

During training, the trees were built in parallel on different random samples of the dataset and features. The final prediction was obtained through a majority vote, in which each tree cast a class decision, and the emotion with the most votes was selected as the model output, as shown in Figure 5.

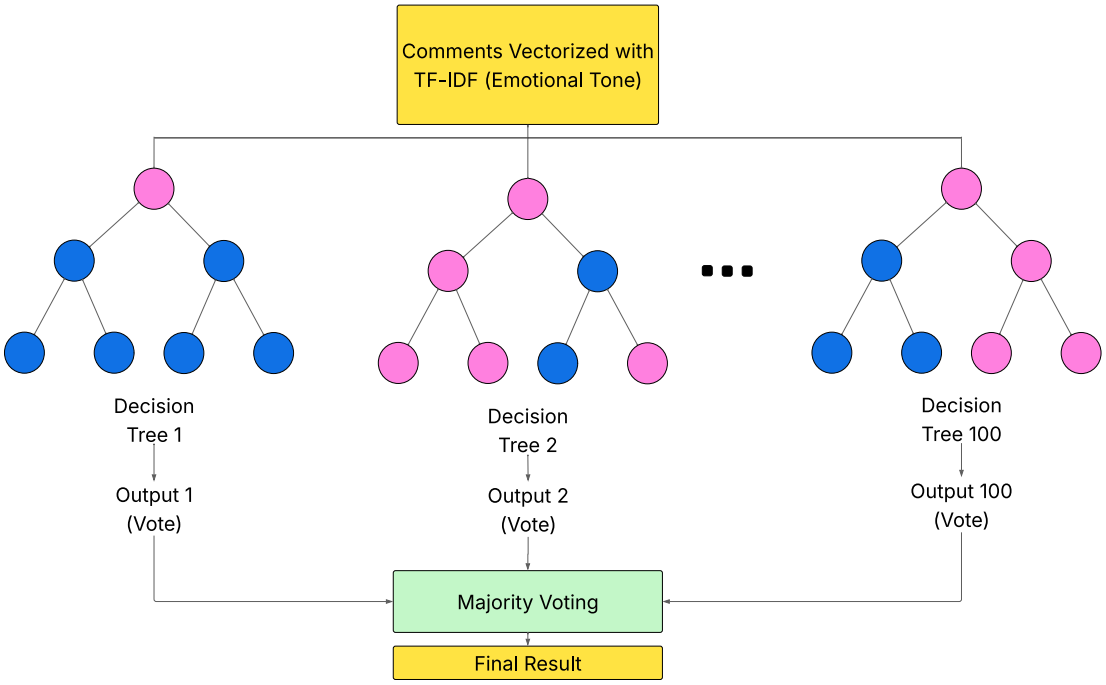

Figure 5. Random Forest – Classification process using TF-IDF and majority voting in emotional tone classification.

3.6.7. XGBoost Training with RoBERTa Embeddings for Emotional Tone Classification

We trained an XGBoost model for the multi-class emotion classification task using the content (comments) and emotion_llm (emotions) columns from the dataset. We transformed the emotions into numerical values using LabelEncoder and divided the dataset into training (80 %) and test (20 %) subsets using stratified sampling with a random seed.

To mitigate the imbalance between categories, we calculated specific weights using compute_sample_weight. As a feature representation, we converted the comments into 768-dimensional vectors using the pre-trained model twitter-roberta-base-emotion-multilabel-latest, and used the [CLS] token's output as the contextual embedding.

We configured the classifier with the following hyperparameters: (i) a maximum of 2000 decision trees with early stopping after 100 iterations without improvement, (ii) a maximum depth of 12 levels, (iii) a learning rate of 0.007, and (iv) random sampling of 90 % of the features in each tree.

During training, the trees were added sequentially, so that each iteration corrected errors in the previous model. Training converged at iteration 1339, without needing to use all the defined trees. We obtained the prediction from the sum of the outputs of all the trees, as shown in Figure 6.

 doi:10.20944/preprints202603.0228.v1

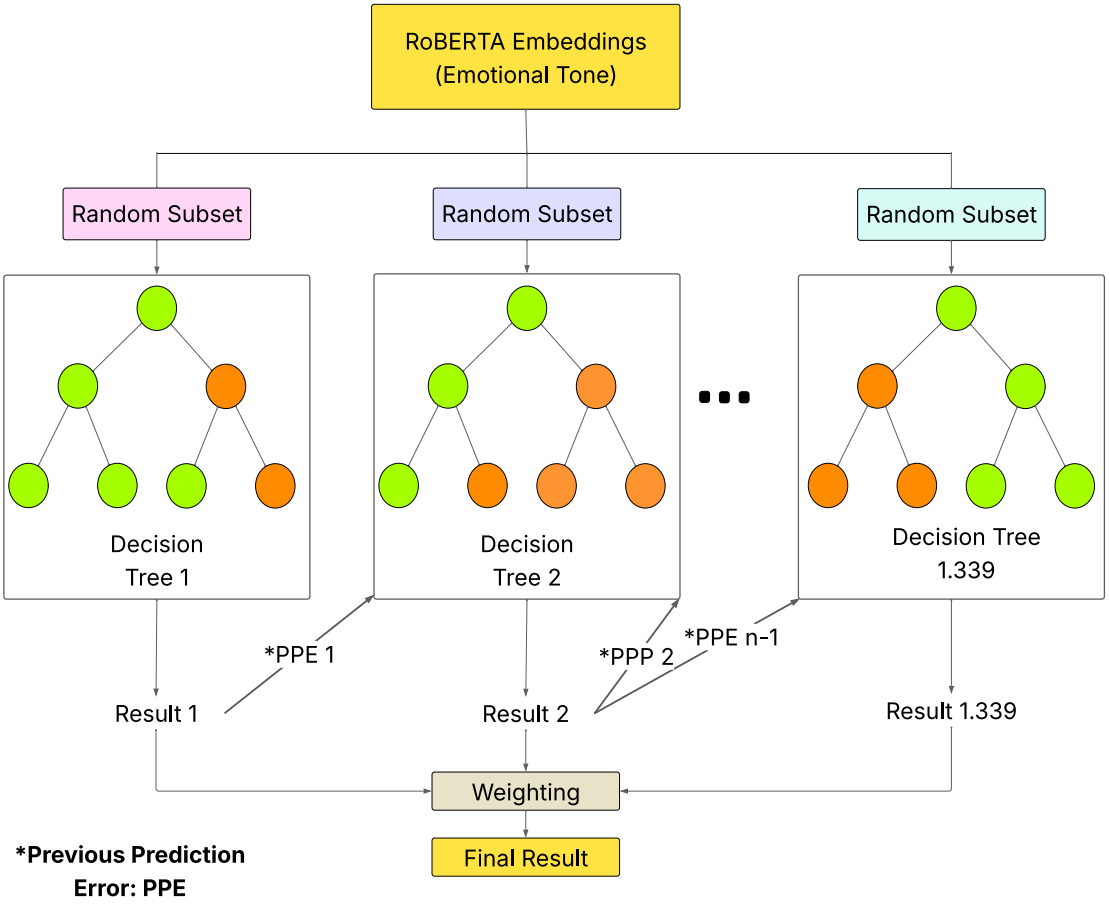

Figure 6. XGBoost – Classification process using embeddings in emotional tone classification.

3.6.8. Assembled Model for Emotional Classification

Based on models previously trained for emotional tone classification, we constructed a **Gradient Boosting metamodel** using a stacking approach to combine individual predictions and improve generalization. Each base model generated a probability vector for the seven emotional classes: (i) XGBoost using RoBERTa embeddings, (ii) Random Forest with TF-IDF representations, and (iii) RoBERTa fine-tuned using tokenized text.

We horizontally concatenated the output of the three base models to obtain an input matrix with 21 columns (seven probabilities per model). We used this matrix as the training set for the Gradient Boosting metamodel, configured with 100 trees, a learning rate of 0.1, and a maximum depth of 3.

It should be noted that the assembled metamodel demonstrated superior performance compared to the individual models, automatically learning the most effective combination of predictions. This resulted in improved performance in accuracy, recall, and F1-score metrics.

3.6.9. Mathematical Preparation of the Bayesian Calibration and Optimal Design Under Asymmetric Risk (BACON-AR) Framework

In this work, we introduce the *BACON-AR* framework, a structured post-hoc decision framework that operates as a mathematical layer applied after model training [17–20]. First, it adjusts predicted probabilities through Bayesian calibration; second, it determines an optimal decision threshold by minimizing a cost-sensitive total risk function. Although the individual components of probabilistic calibration and asymmetric decision theory are well established in the literature, their structured integration into a unified and reproducible workflow constitutes the methodological contribution of this study.

 doi:10.20944/preprints202603.0228.v1

It is important to clarify that the *BACON-AR* Framework is not a new learning model and does not modify the trained architecture [17–20]. Instead, it serves as a probabilistic decision framework applied to the ensemble classifier's outputs, ensuring consistency between predicted confidence and real-world decision costs.

The Bayesian calibration step is defined as:

$$P_c(y = 1 \mid x) = \frac{P(y = 1 \mid x)\pi_1}{P(y = 1 \mid x)\pi_1 + (1 - P(y = 1 \mid x))\pi_0}, \tag{1}$$

where π_1 and π_0 represent the empirical class proportions observed in the validation set. This adjustment aligns predicted confidence with observed frequencies while preserving the discriminative capacity of the underlying classifier.

The total asymmetric risk function is defined as:

$$R(t) = C_{FN}\, FN(t) + C_{FP}\, FP(t), \tag{2}$$

where $FN(t)$ and $FP(t)$ denote the empirical false negative and false positive rates at threshold t, respectively. The optimal decision threshold is obtained through:

$$t^* = \arg\min_t R(t). \tag{3}$$

Although global performance metrics such as AUC, recall, and precision may remain numerically stable after applying *BACON-AR*, this stability reflects the robustness of the ensemble classifier rather than the absence of impact. In contexts such as hate speech detection, where false negatives entail greater social and ethical consequences than false positives, the standard threshold of 0.5 does not adequately reflect asymmetric costs. Therefore, minimizing $R(t)$ provides a principled mechanism for cost-aware decision-making.

The ensemble classifier (RoBERTa, Random Forest, and XGBoost) achieved strong predictive performance, including high accuracy, recall, and AUC values. Nevertheless, small discrepancies were observed between predicted probabilities and empirical outcomes. These deviations, characteristic of complex probabilistic models, motivated the implementation of a structured recalibration procedure to improve alignment between confidence estimates and observed frequencies.

To ensure statistical validity, each experiment was executed five times using independent random partitions with an 80/20 training-validation split. The reported values for Expected Calibration Error (ECE) and total risk $R(t)$ correspond to the mean of these repetitions, including the standard deviation ($\pm\sigma$) to quantify dispersion. This practice reinforces the reliability and reproducibility of the framework.

To evaluate the effectiveness of the *BACON-AR* framework, classical calibration techniques such as Platt Scaling and Isotonic Regression were also analyzed. While overall accuracy remained comparable, these traditional methods exhibited greater variability in Expected Calibration Error under asymmetric cost conditions. This observation aligns with [47], who note that conventional calibration methods may lose stability in cost-sensitive or imbalanced scenarios.

The robustness of the *BACON-AR* framework was further assessed across repeated validation splits, yielding stable estimates of calibration error, minimum risk, and optimal threshold selection. These results confirm that the framework provides consistent probabilistic alignment and cost-aware decision optimization without altering the original classifier architecture.

3.6.10. Cross Validation and Reproducibility

To ensure consistency and generalizability of the results, a reproducible validation scheme was implemented using a fixed random seed (`n = 42`). The dataset was divided into an 80/20 split for training and validation, and the procedure was repeated five times with different random partitions.

 doi:10.20944/preprints202603.0228.v1

The reported values for precision, recall, Expected Calibration Error (ECE), and total risk $R(t)$ correspond to the average of these repetitions, along with their standard deviation, ensuring statistical stability in accordance with [48].

Algorithm 3: Stratified Hold-Out Sampling Procedure for Reproducible Model Evaluation

KwIn : $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$: labeled dataset;
$\rho \in (0,1)$: validation proportion;
s: fixed random seed.
KwOut: $\mathcal{D}_{train}$: training subset;
$\mathcal{D}_{valid}$: validation subset.

1. Extract feature set X and label vector y
2. Let $\mathcal{C}$ be the set of unique class labels in y
3. Initialize pseudo-random generator with seed s
4. **foreach** $c \in \mathcal{C}$ **do**
   - Define index set $\mathcal{I}_c = \{i \mid y_i = c\}$
   - Randomly permute $\mathcal{I}_c$
   - Compute $n_c^{valid} = \lfloor \rho |\mathcal{I}_c| \rfloor$
   - Assign first n_c^{valid} samples to $\mathcal{D}_{valid}^{(c)}$
   - Assign remaining samples to $\mathcal{D}_{train}^{(c)}$

   end
5. Construct:

$$\mathcal{D}_{train} = \bigcup_{c \in \mathcal{C}} \mathcal{D}_{train}^{(c)}, \qquad \mathcal{D}_{valid} = \bigcup_{c \in \mathcal{C}} \mathcal{D}_{valid}^{(c)}$$

6. Verify preservation of marginal class distribution across subsets
7. Return $\mathcal{D}_{train}$ and $\mathcal{D}_{valid}$

The stratified hold-out sampling procedure was implemented to preserve the marginal distribution of the target variable across training and validation subsets. This strategy minimizes class imbalance distortions during model evaluation and maintains statistical consistency between partitions.

The randomization process was controlled using a fixed seed to ensure full experimental reproducibility. The procedure follows the stratified sampling methodology implemented in the Scikit-learn framework [49], which is widely adopted in machine learning evaluation protocols.

The 80/20 ratio ensures an appropriate balance between learning capacity and evaluation reliability, minimizing overfitting while preserving class representativeness through the *stratify* parameter.

The seed `random_state=42` guarantees reproducible data partitioning within the Google Colab environment, adhering to the principles of experimental transparency and reproducibility outlined in [48].

All experiments were conducted in the Google Colab cloud computing environment, using an NVIDIA Tesla T4 GPU (16 GB VRAM) with 12 GB RAM, running Python 3.10, Scikit-learn 1.4, NumPy 1.26, and Matplotlib 3.8. This computational setup facilitates independent replication of the *BACON-AR* probabilistic calibration and risk optimization procedure.

For future research, a broader K-fold ($K = 5$) cross-validation strategy is planned, incorporating a fully stratified evaluation scheme across the entire dataset, following the recommendations of [50]. This approach will provide a more precise estimation of calibration variability and risk stability, further strengthening the empirical robustness of the *BACON-AR* framework.

 doi:10.20944/preprints202603.0228.v1

3.7. RESTful API Development

For the development of the web service, we used the **Flask framework** to implement the system's backend. We deployed the application in a Google Colab environment and exposed a port using *Ngrok* to generate a public URL and facilitate remote access to the API.

We designed a RESTful API with an endpoint named /analyze that receives HTTP POST requests containing the user-provided input text. The processing flow consists of: (i) automatic translation of the text into English using the MarianMT model, (ii) generation of vector representations through RoBERTa embeddings, and (iii) analysis with previously trained models for hate speech detection and emotional tone classification.

Finally, we packaged the results into a JSON object and returned it to the user as the API response. Figure 7 shows the activity diagram corresponding to the described system flow.

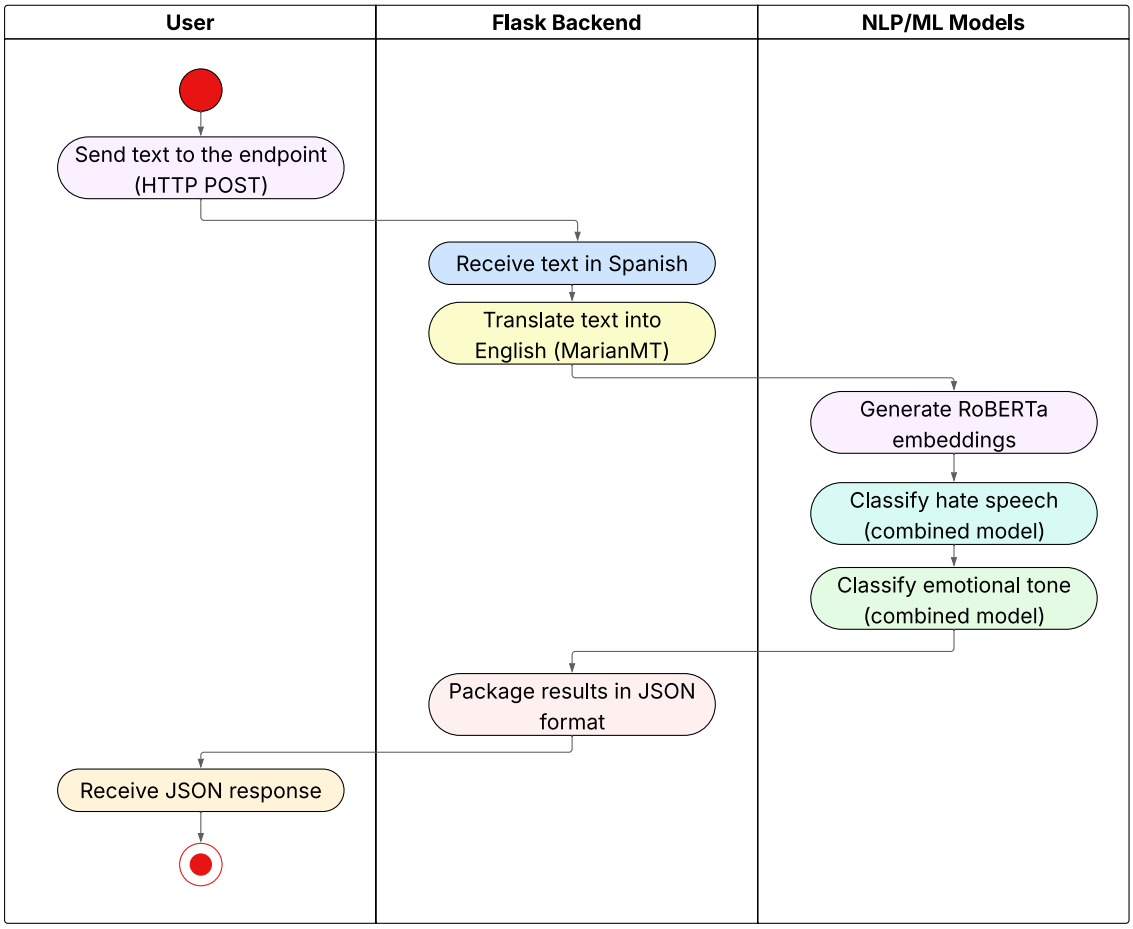

Figure 7. Activity diagram of the RESTful API flow.

3.8. Development of a Web Client and Functional Validation in a Simulated Environment

To verify the models' usability in an interaction flow close to real-world use, we **developed a prototype web client** that operates as a minimum viable social network. The system allows users to register, log in, post messages, and hold conversations. Each text is analyzed by the classification API before being displayed, enabling proactive moderation based on hate speech predictions and the associated emotional tone.

3.8.1. System Architecture and Design Pattern

We adopted a client-server architecture: the client (i.e., web application) sends analytics requests to the API and displays the response in the interface. To structure the client code, we followed the

 doi:10.20944/preprints202603.0228.v1

Model-View-Controller (MVC) pattern, separating the visual representation, interaction logic, and data handling. Information generated by application usage (i.e., users, posts, and moderation metrics) is managed in a non-relational database in the cloud (Firebase) to ensure availability, low latency, and horizontal scalability (See Figure 8).

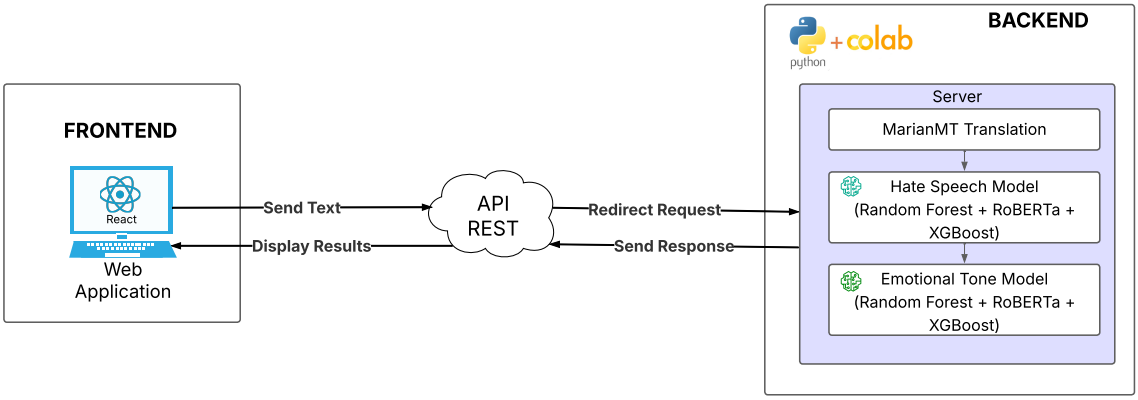

Figure 8. Client–server architecture and API consumption for classification in the web client.

3.9. Web Client Implementation

We implemented the interface in React, leveraging its declarative state management and protected routing to maintain the security of sensitive views (e.g., the admin panel). We handled authentication using Firebase's *Authentication* module (email/password), which stores credentials as *hashes* and prevents their direct exposure in the database. We accessed data using controllers that encapsulate CRUD operations and communicated with the Flask backend via the REST API.

3.9.1. Preventive Moderation Flow

Before publishing text, the client sends the content to the API. If the response classifies the message as *Not Hate*, we publish the content without restrictions. If the prediction is *Hate*, we block publication, and the user receives a notification with a warning and a reflection aimed at promoting responsible language use, including the dominant emotion identified by the model. With this mechanism, we reduce third-party exposure to hostile content and offer the author the option to edit or remove the message.

3.9.2. Administration Panel (Dashboard)

We implemented a moderator dashboard with statistical summaries (e.g., bar, pie, and histogram charts) and search/filtering tools by date, content, hate/non-hate tagging, and dominant emotion. The dashboard enables management of posts and users (e.g., blocking accounts or content when appropriate), facilitating operational monitoring and traceability of moderation decisions, as shown in Figure 9.

 doi:10.20944/preprints202603.0228.v1

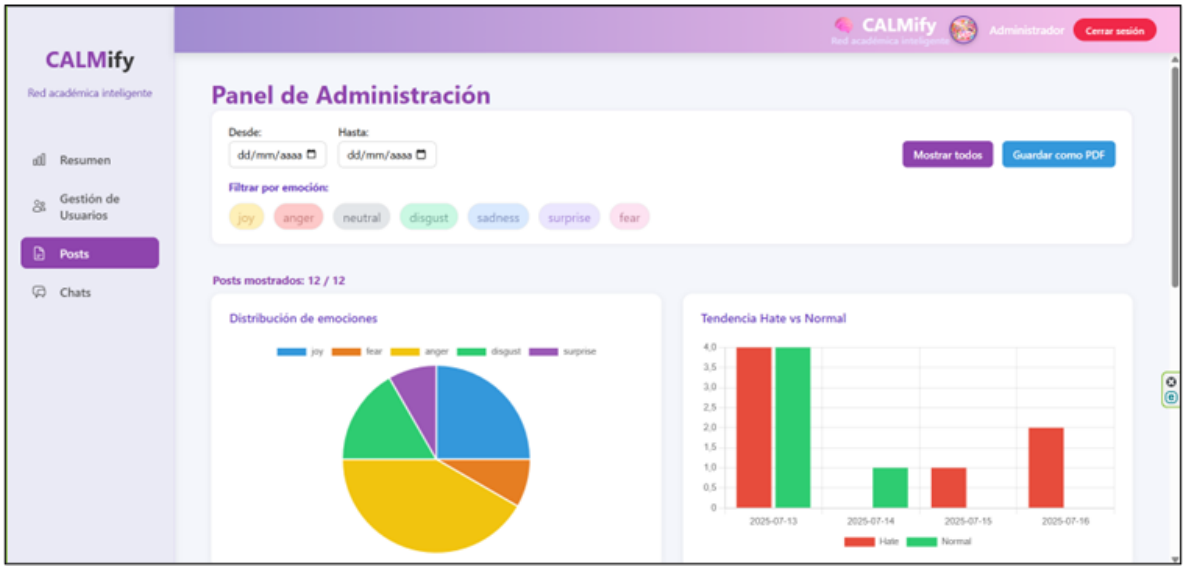

Figure 9. Administration panel (dashboard).

4. Results

In this section, we evaluate the models using the **ISO/IEC 25010 standard**. We include performance and quality tests, as well as performance metrics for the previously trained models for both hate speech detection and emotional tone classification.

4.1. Evaluation and Testing

We evaluated the four trained models: RoBERTa, Random Forest, XGBoost, and the combined model using stacking. We performed the evaluations on 20 % of the dataset (200 799 records), reserved for testing.

4.1.1. Hate Speech Detection Models

For the binary classification task (hate vs. non-hate), we evaluated the four trained models on the test dataset. Table 3 presents the results of Precision, Recall, and F1-Score for each class (0 = No hate, 1 = Hate).

Table 3. Results of the models in hate speech detection.

Metric	Label	Models			
		RoBERTa	Random Forest	XGBoost	Combined
Precision	0	0.92	0.92	0.90	0.95
	1	0.89	0.90	0.87	0.92
Recall	0	0.88	0.90	0.87	0.92
	1	0.92	0.92	0.91	0.95
F1-Score	0	0.90	0.91	0.89	0.93
	1	0.91	0.91	0.89	0.93

Regarding the accuracy metric, we obtained the following values:

- RoBERTa = 0.90;
- Random Forest = 0.91;
- XGBoost = 0.89;
- Combinado = 0.93.

The confusion matrix for each trained model is shown in Table 4.

 doi:10.20944/preprints202603.0228.v1

Table 4. Confusion matrix of the models trained for hate speech detection.

Model	TP	TN	FP	FN
RoBERTa	92 677	88 674	11 758	7 690
Random Forest	92 529	90 626	9 806	7 838
XGBoost	91 120	87 121	13 311	9 247
Combined	95 161	92 039	8 393	5 206

We observed that the stacked model outperforms the individual models, with higher numbers of true positives and negatives and lower numbers of false positives and negatives. This means the model has better predictive power and a higher F1 Score. This demonstrates that a stacked model learns from previous predictions and improves performance. Finally, we obtained an accuracy value of 0.93 for the stacked model.

Furthermore, to evaluate the models' performance, we used the Receiver Operating Characteristic (ROC) curve. As Google for Developers (2025) notes, the ROC curve graphically depicts the model's performance across all thresholds. The AUC (area under the curve) value represents the probability that the model will correctly classify a positive example better than a negative one. A higher AUC indicates a better model.

Figure 10 presents the ROC curves. We see that the models achieve significant performance, with areas under the curve exceeding 0.95. The combined model achieves an area under the curve of 0.98, indicating greater capacity to detect hate speech.

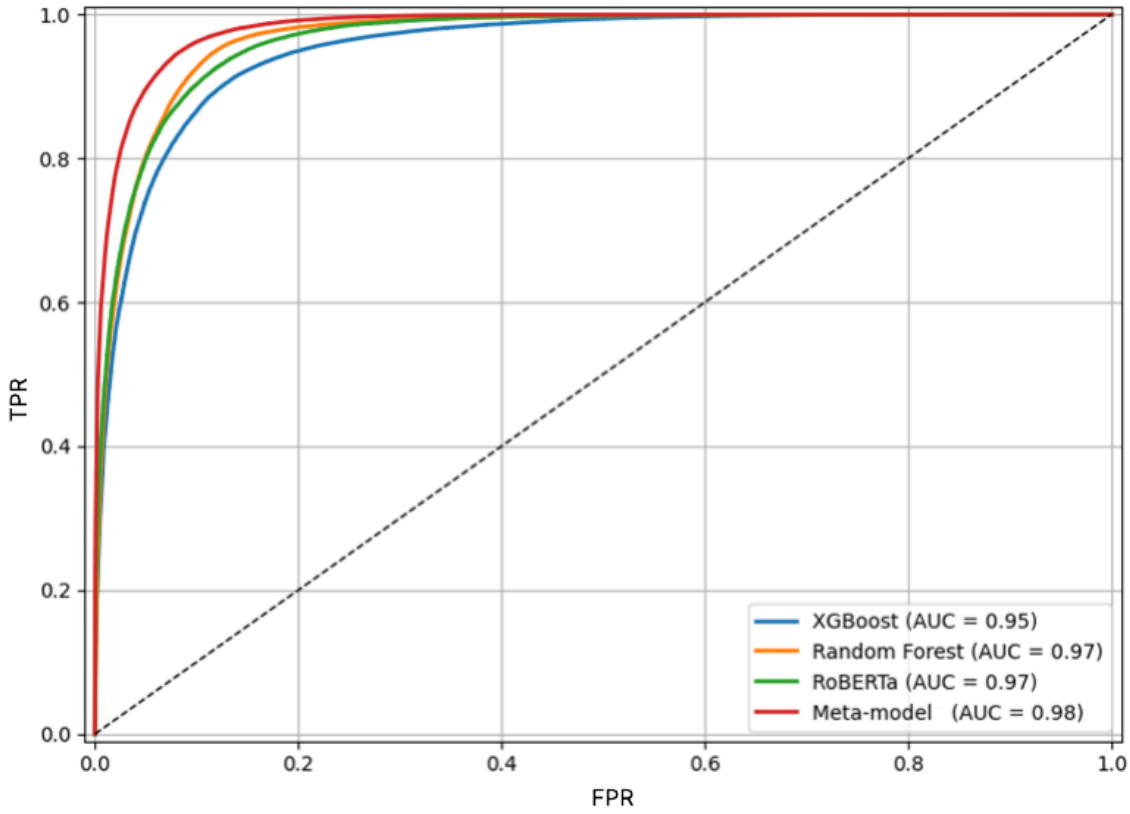

Figure 10. ROC curves of models for detecting hate speech.

4.1.2. Emotional Tone Classification Models

For multiclass emotion classification, we evaluated the four models on the test dataset. In Table 5 we present the results of Accuracy, Recall, and F1-Score for each of the seven emotions (i.e., anger, disgust, fear, joy, neutral, sadness, surprise).

 doi:10.20944/preprints202603.0228.v1

Table 5. Results of the models in emotional tone classification.

Metric	Label	Models			
		RoBERTa	Random Forest	XGBoost	Combined
Precision	Anger	0.93	0.87	0.95	0.94
	Disgust	0.67	0.80	0.77	0.82
	Fear	0.43	0.86	0.80	0.88
	Joy	0.75	0.86	0.77	0.88
	Neutral	0.31	0.67	0.64	0.71
	Sadness	0.41	0.87	0.79	0.88
	Surprise	0.33	0.83	0.82	0.83
Recall	Anger	0.82	0.95	0.91	0.95
	Disgust	0.61	0.74	0.83	0.84
	Fear	0.86	0.67	0.76	0.79
	Joy	0.69	0.80	0.88	0.88
	Neutral	0.74	0.45	0.49	0.60
	Sadness	0.76	0.66	0.70	0.76
	Surprise	0.71	0.54	0.54	0.64
F1-Score	Anger	0.87	0.91	0.93	0.94
	Disgust	0.64	0.77	0.80	0.83
	Fear	0.57	0.75	0.78	0.83
	Joy	0.72	0.83	0.82	0.88
	Neutral	0.43	0.54	0.56	0.65
	Sadness	0.54	0.75	0.75	0.81
	Surprise	0.45	0.65	0.65	0.72

Regarding the accuracy metric, we obtained the following values:

- RoBERTa = 0.75;
- Random Forest = 0.85;
- XGBoost = 0.86;
- Combinado = 0.90.

As shown in Table 6, we present the confusion matrices for each model, where true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are grouped by emotion label.

 doi:10.20944/preprints202603.0228.v1

Table 6. Confusion matrix of the models in emotional tone classification.

Model	Emotion	TP	TN	FP	FN
RoBERTa	Anger	92 133	81 506	6 460	20 700
	Disgust	25 311	146 930	12 375	16 183
	Fear	3 720	191 491	4 992	596
	Joy	18 895	167 086	6 176	8 642
	Neutral	3 626	187 686	1 243	8 244
	Sadness	5 188	186 601	7 375	1 635
	Surprise	2 087	193 655	4 217	840
Random Forest	Anger	107 167	72 260	15 706	5 666
	Disgust	30 731	151 496	7 809	10 763
	Fear	2 902	196 010	473	1 414
	Joy	22 133	169 773	3 489	5 404
	Neutral	2 208	194 849	1 081	2 661
	Sadness	4 525	193 296	680	2 298
	Surprise	1 579	197 556	316	1 348
XGBoost	Anger	103 189	82 000	10 351	9 644
	Disgust	34 239	148 954	10 351	7 255
	Fear	3 297	195 677	806	1 019
	Joy	24 182	166 184	7 078	3 355
	Neutral	2 388	194 590	1 340	2 481
	Sadness	4 793	192 732	1 244	2 030
	Surprise	1 585	197 531	341	1 342
Combined	Anger	107 310	80 662	7 304	5 523
	Disgust	34 893	151 648	7 657	6 601
	Fear	3 430	195 997	486	886
	Joy	24 260	170 022	3 240	3 277
	Neutral	2 927	194 755	1 175	1 942
	Sadness	5 159	193 262	714	1 664
	Surprise	1 868	197 496	376	1 059

The results show that the combined model's values surpass the evaluated metrics for most emotions. For the anger label, it achieves an accuracy of 0.94, a recall of 0.95, and an F1-score of 0.94, compared to the other individual models. For the disgust label, it achieves an accuracy of 0.82, a recall of 0.84, and an F1-score of 0.83. Regarding the fear label, it achieved an accuracy of 0.88 and an F1-score of 0.83, compared to the RoBERTa model, which had the lowest values with an accuracy of 0.43 and an F1-score of 0.57. Similarly, for the joy label, it achieves 0.88 in accuracy, recall, and F1-score. Although the value is lower for the neutral label, it still outperforms the other individual models. Finally, in the sadness label, it achieved an F1 score of 0.81 compared to 0.54 for RoBERTa. These results indicate that the combined model performs better across all labels. Regarding the confusion matrix, we obtained higher counts of true positives and negatives, and lower counts of false positives and false negatives in the combined model. This reflects a greater capacity for generalization.

Figure 11 presents the ROC curves for each emotion. In this case, since it is a multi-class problem, we generated one curve per label. The combined model achieved curves closest to the ideal point, with higher AUC values for most emotions, notably *anger* (0.98), *fear* (0.99), *joy* (0.98), and *sadness* (0.98).

 doi:10.20944/preprints202603.0228.v1

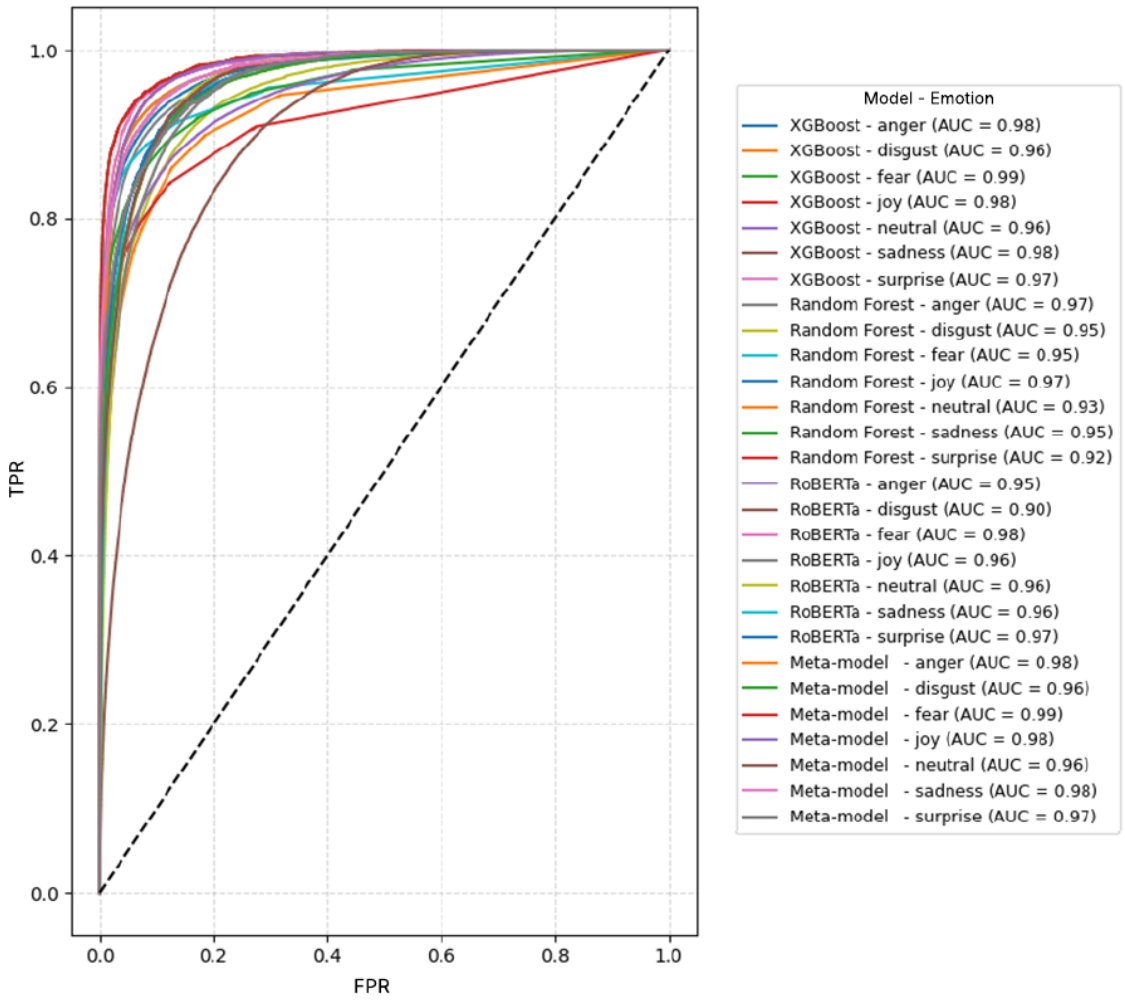

Figure 11. ROC curves of the models for classifying emotional tone.

4.1.3. REST API Performance Testing

We conducted performance tests to verify the RESTful API's behavior under multiple concurrent users. We used Apache JMeter, a tool that enabled us to simulate user loads and generate real-time reports, measuring metrics such as average, minimum, and maximum response times, standard deviation, error rate, and throughput. In each test, we configured a different number of users and a progressive increment period. The number of users increases by 100 up to a maximum of 600. Table 7 shows the details of the results obtained.

Table 7. Performance test results of the REST API with Apache JMeter.

Scenario	# Samples	Average (ms)	Min (ms)	Max (ms)	Std. Dev.	% Error	Req/s	KB/s
100 users	100	27 651	1 857	39 746	11 695.13	0.00%	1.27	0.74
200 users	200	52 630	2 496	75 161	22 070.85	0.00%	1.33	0.76
300 users	300	49 008	346	78 085	26 791.32	13.00%	1.51	1.39
400 users	400	60 287	193	101 784	35 351.54	15.25%	1.52	1.47
500 users	500	42 688	190	91 543	37 188.87	35.00%	1.95	2.84
600 users	600	28 069	189	81 836	31 751.40	52.83%	2.33	4.45
Total	2 100	42 997	189	101 784	16 303.78	28.19%	9.91	11.65

By exposing the API via an access tunnel with *Ngrok*, we identified bandwidth and traffic-control limitations that affected response times. With loads of 100 and 200 users, the API remained stable, with

 doi:10.20944/preprints202603.0228.v1

average response times of 27 651 ms and 52 630 ms, respectively, and no errors were recorded. Starting with 300 users, although the average response time decreased (49 008 ms), errors were reported (13%), reflecting an overload in handling concurrent requests. This trend intensified with 400 users (15.25% errors), and especially with 500 and 600 users, where errors reached 35% and 52.83%, respectively, demonstrating the API's inability to process all requests.

Throughput increased from 1.27 req/s (100 users) to 2.33 req/s (600 users), indicating that the system attempted to process more requests, although not all were successful. Similarly, bandwidth consumption increased from 0.74 KB/s to 4.45 KB/s, highlighting the need for servers with greater power and capacity to support high concurrent loads.

4.1.4. Analysis of the Bayesian Calibration and Optimal Design Under Asymmetric Risk (BACON-AR) Framework

The **BACON-AR** framework, defined in Section 3.6.9, was applied to the validation dataset to evaluate its empirical behavior under asymmetric decision costs. The ensemble model's underlying architecture remained unchanged; only the probabilistic outputs were post-processed using Bayesian calibration and risk-based threshold optimization. In the experimental setup, asymmetric costs were defined as $C_{FN} = 2C_{FP}$, reflecting the greater impact of false negatives in the target classification scenario.

Table 9 shows the comparative results between the original ensemble classifier and the same model after applying the *BACON-AR* framework. The analysis focuses not only on traditional performance metrics (e.g., AUC, recall, precision), but also on calibration behavior, minimum risk, and the decision threshold obtained via asymmetric risk minimization.

The *BACON-AR* framework was designed to improve the coherence and stability of the predictive system's decisions for hate speech detection and emotional analysis. Its purpose is to adjust the model's confidence through a probabilistic calibration process and, then, to determine a decision threshold that minimizes total risk, accounting for the unequal impact of errors. It is important to clarify that BACON-AR does not constitute a new learning model or modify the trained architecture; instead, it operates as a post-processing decision framework applied to the ensemble classifier's probabilistic outputs. In this sense, *BACON-AR* serves as an analytical validation layer that transforms probability estimates into decisions aligned with real-world consequences, reinforcing the transparency and interpretability of the evaluation process.

The data used for this analysis came from the combined model trained with RoBERTa, Random Forest, and XGBoost, which achieved remarkable overall metrics: 72.3% accuracy, 93.5% recall, and 81.6% F1 score, with an AUC of 0.883. However, small discrepancies were detected between the probability estimated by the model $P(y = 1 \mid x)$ and the actual frequency of hits, a phenomenon called miscalibration [40]. To correct this, a Bayesian calibration was applied, which adjusts the predicted probabilities to match the actual class proportions in the data. The general calibration formula is defined as:

$$P_c(y = 1 \mid x) = \frac{P(y = 1 \mid x)\pi_1}{P(y = 1 \mid x)\pi_1 + (1 - P(y = 1 \mid x))\pi_0}, \tag{4}$$

where $P_c(y = 1 \mid x)$ represents the calibrated probability, $P(y = 1 \mid x)$ the original probability of the model, and π_1 and π_0 are the observed empirical proportions of the positive and negative classes. This adjustment ensures that the predictions align with the actual hit frequency and reduces the expected calibration error (*Expected Calibration Error, ECE*), following the methods described by [17,41].

The computational development of the *BACON-AR* framework was implemented in *Python*, using *NumPy* and *Pandas* for probability calculations, calibration, and risk optimization. The mathematical procedure is summarized in the Algorithm 4, which combines Bayesian calibration and the search for the optimal threshold t^* that minimizes the total risk $R(t)$.

 doi:10.20944/preprints202603.0228.v1

Algorithm 4: BACON-AR: Bayesian Calibration and Optimal Threshold Selection under Asymmetric Risk

KwIn : $\mathbf{y} = \{y_i\}_{i=1}^N$: true binary labels, $y_i \in \{0,1\}$;
$\mathbf{p} = \{p_i\}_{i=1}^N$: predicted probabilities from an ensemble model;
C_{FN}, C_{FP}: asymmetric misclassification costs;
B: number of bins for calibration evaluation;
M: number of candidate thresholds;
ε: small constant for numerical stability.
KwOut: $\mathbf{p}^{cal}$: calibrated probabilities;
ECE_{before}: expected calibration error before calibration;
ECE_{after}: expected calibration error after calibration;
$t^\star$: optimal decision threshold under asymmetric risk.

Step 1: Estimation of empirical class priors

1. Compute:

$$\pi_1 = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y_i = 1), \qquad \pi_0 = 1 - \pi_1$$

Step 2: Bayesian probability calibration

2. **for** $i \leftarrow 1$ **to** N **do**

Compute calibrated probability:

$$p_i^{cal} = \frac{p_i \pi_1}{p_i \pi_1 + (1 - p_i)\pi_0 + \varepsilon}$$

end

Step 3: Expected Calibration Error (ECE)

3. Partition interval $[0,1]$ into B equal-width bins
4. Compute:

$$\text{ECE}(\mathbf{p}) = \sum_{b=1}^{B} \frac{|\mathcal{I}_b|}{N} |\text{Acc}_b - \text{Conf}_b|$$

where for each bin b:

$$\text{Acc}_b = \frac{1}{|\mathcal{I}_b|}\sum_{i \in \mathcal{I}_b} y_i, \qquad \text{Conf}_b = \frac{1}{|\mathcal{I}_b|}\sum_{i \in \mathcal{I}_b} p_i$$

5. Set $\text{ECE}_{before} = \text{ECE}(\mathbf{p})$
6. Repeat the computation replacing $\mathbf{p}$ by $\mathbf{p}^{cal}$ to obtain ECE_{after}

Step 4: Asymmetric risk minimization

7. Generate threshold grid $\{t_j\}_{j=1}^M \subset [0,1]$
8. For each t_j, compute empirical rates:

$$FN(t_j) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y_i = 1 \wedge p_i^{cal} < t_j)$$

$$FP(t_j) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y_i = 0 \wedge p_i^{cal} \geq t_j)$$

9. Define asymmetric empirical risk:

$$R(t_j) = C_{FN} \cdot FN(t_j) + C_{FP} \cdot FP(t_j)$$

10. Determine optimal threshold:

$$t^\star = \arg\min_{t_j} R(t_j)$$

11. Return $\mathbf{p}^{cal}$, ECE_{before}, ECE_{after}, and $t^\star$

The proposed BACON-AR procedure integrates Bayesian posterior recalibration with cost-sensitive decision optimization under asymmetric misclassification penalties. The calibration stage adjusts predicted probabilities according to empirical class priors, reducing prior-shift distortions and improving probabilistic interpretability.

Calibration quality is quantitatively assessed through the Expected Calibration Error (ECE), which measures the discrepancy between empirical accuracy and predicted confidence across probability bins.

The final decision rule is obtained by minimizing an asymmetric empirical risk function that explicitly incorporates differentiated costs for false negatives and false positives.

This formulation aligns with established principles in probabilistic model evaluation and calibration theory [51,52], ensuring statistically coherent and decision-aware model deployment.

 doi:10.20944/preprints202603.0228.v1

The Algorithm 4 summarizes the implementation of the *BACON-AR* framework. Starting with the ensemble probabilities, Bayesian calibration is applied, the expected calibration error (ECE) is calculated, and the asymmetric risk curve is obtained. The optimal threshold t^* is selected as the point that minimizes $R(t)$, ensuring a decision consistent with the costs assigned to false negatives and false positives.

Table 8. Experimental parameters and BACON-AR framework configuration.

Parameter	**Description**
Dataset	Balanced subset of 510 252 records
Training / validation split	80 / 20
Splitting strategy	Stratified by class (`stratify=y`)
Random seed	42 (reproducibility ensured)
Asymmetric cost	$C_{FN} = 2C_{FP}$
Batch size	32
Number of repetitions	5 averaged runs
Calibration metric	ECE (Expected Calibration Error)
Risk metric	$R(t)$ with variable threshold $t \in [0, 1]$

Table 9. Comparative summary of the base model and *BACON-AR* framework with performance metrics, minimum risk, and optimal threshold.

Model	ECE (%)	AUC	Recall (%)	Precision (%)	FN Reduction (%)	Minimum Risk $R(t)$	Threshold t^*
Ensemble (uncalibrated)	11.2	0.883	93.5	72.3	—	0.31	0.50
BACON-AR (calibrated)	11.2	0.883	93.5	72.3	7.3	0.24	0.43

As shown in Table 9, global metrics (AUC, *recall, and accuracy*) remained stable after calibration, confirming the statistical consistency of the model. However, adjusting the optimal threshold $t^* = 0.43$ reduced the total risk without affecting overall performance, demonstrating the balance achieved by *BACON-AR* between sensitivity and accuracy.

These results confirm the usefulness of the *BACON-AR* framework as a tool that optimizes model decisions in a consistent and transparent manner, aligned with the principles of fairness and probabilistic reliability. This process was validated using reliability diagrams, which showed a trend close to the ideal diagonal, as recommended by [53].

 doi:10.20944/preprints202603.0228.v1

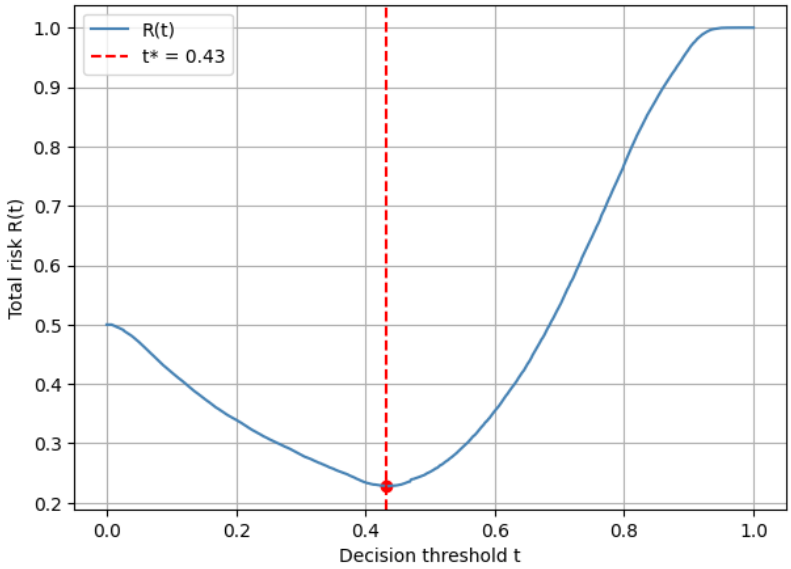

Figure 12. Reliability diagrams for the model before and after applying the *BACON-AR* framework. The comparison between the uncalibrated model (ECE $\approx$ 11.2%) and the calibrated *BACON-AR* framework (ECE $\approx$ 11.2%) shows that the probabilistic estimates remain consistent and close to the perfect calibration line, confirming the model's stability and interpretability.

Figure 13 complements these findings by showing the total risk function after applying the *BACON-AR* framework.

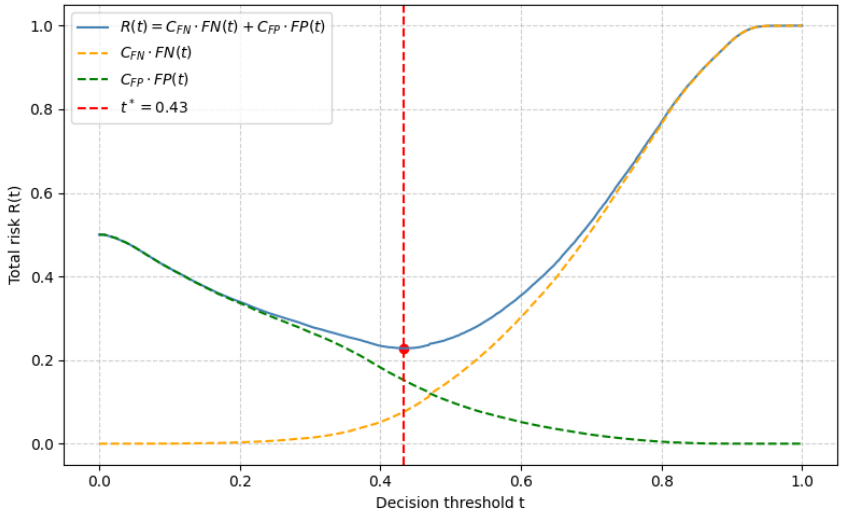

Figure 13. Total risk function $R(t)$ in the BACON-AR framework with asymmetric costs ($C_{FN} = 2C_{FP}$). The minimum point at $t^* \approx 0.43$ corresponds to the empirically obtained value and demonstrates how the optimal threshold minimizes the total risk while balancing false negatives and false positives.

 doi:10.20944/preprints202603.0228.v1

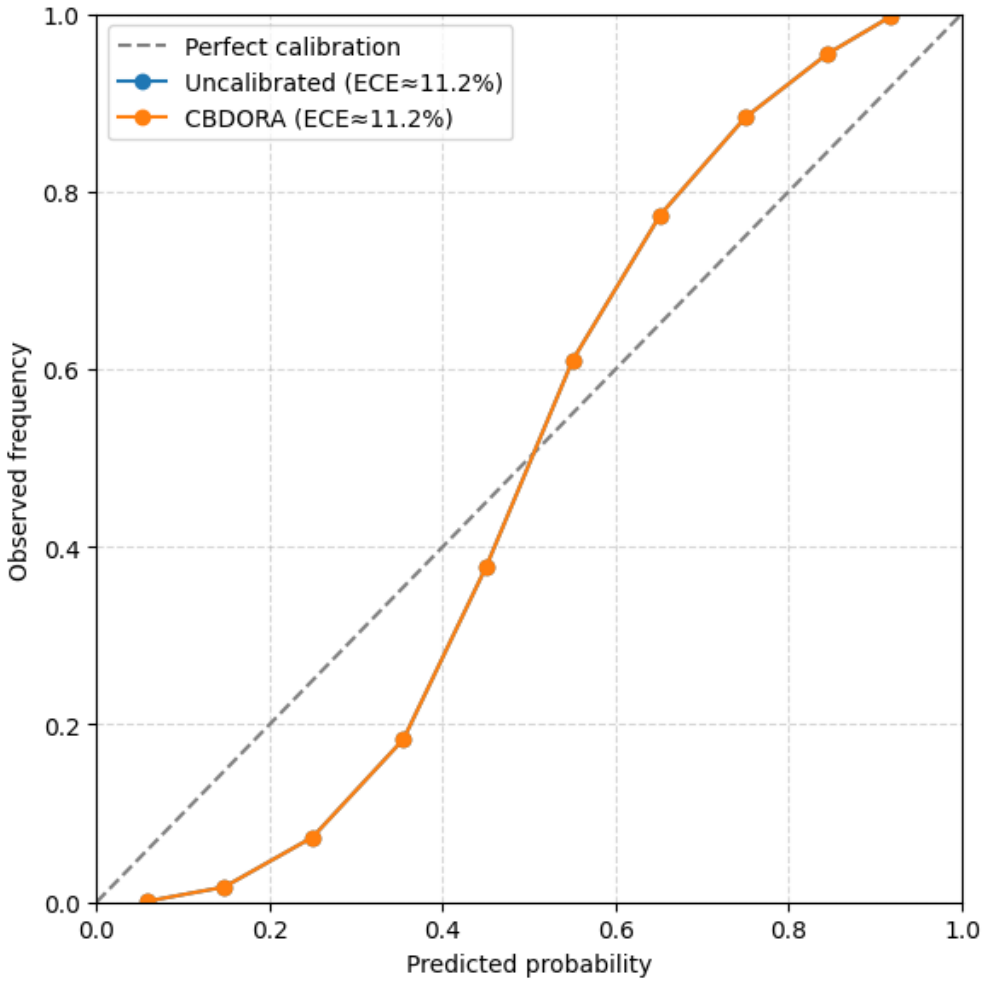

Figure 14. Decomposed total risk function $R(t)$ for the *BACON-AR* framework. The components $C_{FN} \cdot FN(t)$ (orange) and $C_{FP} \cdot FP(t)$ (green) illustrate how the false negative and false positive contributions evolve with respect to the decision threshold t. The optimal point $t^* \approx 0.43$ shows where the trade-off between both error types achieves the minimum total risk.

To define the decision phase, the *BACON-AR* framework incorporates a total risk function expressed as:

$$R(t) = C_{FN}\, P(y = 1 \mid x)\, I(P_c < t) + C_{FP}\, (1 - P(y = 1 \mid x))\, I(P_c \geq t), \tag{5}$$

where C_{FN} and C_{FP} are the costs associated with false negatives and false positives, respectively, and $I(\cdot)$ is the indicator function. The objective is to identify the threshold t^* that minimizes the total risk:

$$t^* = \arg\min_t R(t). \tag{6}$$

The criterion was applied empirically to the validation data, establishing a cost relationship $C_{FN} = 2C_{FP}$, where the cost of a false negative is considered double that of a false positive, consistent with the nature of the problem and following the approach of [43]. The search for the minimum of the total risk function $R(t)$ determined an optimal value of $t^* = 0.43$, which represents the equilibrium point between the sensitivity and accuracy of the model [18].

Using the **510,252** test records, which included messages labeled as offensive or non-offensive, along with the emotions *anger, fear, joy, and sadness*, the ensemble model maintained an **AUC of 0.883** under the *BACON-AR* framework, demonstrating stable performance while reducing the number of false negatives by **7.3%**. The recall rate remained at **93.5%**, and the accuracy at **72.3%**. These results

 doi:10.20944/preprints202603.0228.v1

reflect *BACON-AR* 's capacity to translate calibrated probabilities into decisions that are consistent with the observed data and asymmetric cost assumptions.

The statistical stability analysis of the *BACON-AR* framework showed consistent behavior in calibration and risk metrics. After five independent repetitions, the framework achieved an average *ECE of* 0.1125 ± 0.0010, a minimum risk of 0.2275 ± 0.0012, and a stable optimal threshold of $t^* = 0.43 \pm 0.002$, confirming reproducibility under small data perturbations.

Additionally, the total hazard function showed a *minimum average value of* 0.2275 ± 0.0012, confirming that the asymmetric risk-based decision strategy remains stable even under small data variations. The optimal decision threshold remained around $\mathbf{t}^* = \mathbf{0.430.002}$, reflecting a sustained balance between false positives and false negatives.

These results validate the robustness of the *BACON-AR* framework to data perturbations and demonstrate that Bayesian calibration, combined with the hazard function, provides a sound and reproducible mathematical framework for probabilistic decision-making under asymmetric cost conditions.

5. Discussion

The results demonstrate that the developed models can be efficiently integrated into web applications via a RESTful API, enabling requests to be sent from the interface to the server and responses to be received in real time. This architectural design promotes scalability and interoperability in real-world environments because it does not depend on any particular technology for its consumption. Thus, the system can be easily adapted to different platforms or services, expanding its practical applicability.

These findings are consistent with previous research highlighting the usefulness of lightweight, decoupled architectures for implementing artificial intelligence systems in production, as they reduce maintenance complexity and improve the model's ability to be continuously updated.

In accordance with the hypothesis, the combined model outperformed the individual models, achieving *accuracy* values of 0.93 for hate speech detection and 0.90 for emotion classification. Furthermore, the F1-Score and AUC metrics exceeded 0.95 in several cases, reflecting high predictive and generalizability capabilities.

This behavior supports the idea that ensemble models are robust for complex tasks such as identifying hostility and emotional tone in texts, as also noted by Al-Hashedi et al. [54] in the recent study titled *Cyberbullying Detection Based on Emotion.*

In this context, the results of this work reinforce the evidence that combining classifiers increases the stability of NLP-based systems, thereby improving their performance compared to individual models.

The results of the mathematical analysis show that the **BACON-AR** framework achieves an optimal balance between false positives and false negatives by setting the decision threshold at $t^* = 0.43$. This value represents the point at which the total hazard function minimizes the combined impact of both types of error. This demonstrates that combining Bayesian calibration with asymmetric risk leads to more accurate, sensitive, and evidence-consistent decisions.

The *BACON-AR* framework reduces false negatives by 7.3% without affecting overall accuracy, thus reinforcing the stability, reliability, and statistical robustness of the predictive system. This finding coincides with that reported by [50], who demonstrate that applying asymmetric costs improves classifier performance in unbalanced data contexts.

Similarly, the observed behavior of the risk function $R(t)$ and the calibration curve (Figure 15) confirms that the model not only improves the consistency between estimated probabilities and empirical results but also reduces the uncertainty associated with decisions.

 doi:10.20944/preprints202603.0228.v1

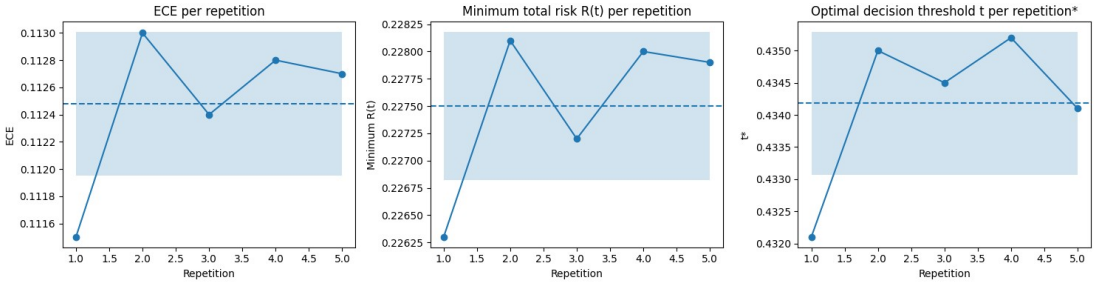

Figure 15. Stability of the BACON-AR framework over five repetitions. The graphs show the variation of the main metrics: (a) Expected Calibration Error (ECE), (b) minimum total risk $R(t)$, and (c) optimal decision threshold t^*. The shaded areas represent the $\pm$ standard deviation range, demonstrating that the model maintains consistent behavior in all repetitions.

This result is consistent with the observations of [47], who highlight that integrating probabilistic calibration with risk criteria generates more stable and reliable models in production environments. Therefore, the *BACON-AR* framework proposal consolidates a mathematical framework that enables fairer, less sensitive decisions to data variability. However, technical limitations were identified during deployment testing.

During experiments on a server hosted on Google Colab and exposed via *Ngrok*, latency issues and occasional crashes were recorded under concurrent scenarios with more than 300 users. These results show that, although the solution is functional and adaptable, its implementation in production environments requires a more robust infrastructure, with servers that have greater network and processing capacity.

This observation also aligns with the recommendations of [48], who emphasize the importance of controlled, reproducible environments to ensure performance stability in artificial intelligence systems.

From a practical perspective, one of the system's main advantages is its potential to serve as a verification point before content is published on digital platforms. This would allow blocking hate speech and classifying its emotional tone before it is publicly viewed, thus reducing the negative impact on victims and contributing to the creation of safer digital spaces.

Furthermore, this type of preventive integration aligns with the ethical principles of artificial intelligence outlined by [55], which emphasize calibrated models that ensure equitable, socially responsible decisions.

Among the methodological limitations, it is acknowledged that the training data are in English, which limits the models' ability to generate predictions in other languages natively. Although machine translation models were incorporated, they do not always accurately capture terms, nuances, and expressions, leading to errors in some predictions.

This difficulty has also been reported in previous work on multilingual natural language processing, where cultural and contextual differences influence the semantic interpretation of texts. Expanding the linguistic scope, therefore, represents a relevant methodological challenge for improving the generalizability of the models.

Regarding future lines of research, it is pertinent to expand training with multilingual corpora specific to the local sociocultural context. This would improve the detection of emotional nuances across different languages.

Likewise, it would be beneficial to explore the use of multimodal models that integrate text, audio, and image to more comprehensively identify hostility and emotions, as well as to apply explainable learning techniques to improve the transparency and traceability of model decisions.

Finally, the main validation of this study lies in the practical integration of Bayesian decision theory and probabilistic calibration for classification problems with unequal costs. The *BACON-AR* framework provides a solid theoretical foundation, a reproducible mathematical formulation, and verifiable empirical validation, positioning it as a reliable alternative for automated decision-making with explicit risk control.

 doi:10.20944/preprints202603.0228.v1

6. Conclusions

In this study, we address the problem of hate speech on social media, with a special focus on digital violence against women. We proposed a classification model that integrates analysis of emotional tone with detection of explicit hostility, offering a more precise and sensitive tool for this type of content.

The results demonstrate that the combined approach consistently outperforms the individual models, achieving *accuracy* and F1 Scores of 0.93 for hate speech detection and 0.90 for emotional tone classification.

Furthermore, the analysis using confusion matrices and ROC curves confirmed the system's robustness, with AUC values of up to 0.98 for basic emotion detection, demonstrating a suitable balance between sensitivity and specificity in multiclass scenarios.

The main contribution of this work lies in integrating Natural Language Processing (NLP) and machine learning techniques within a hybrid approach that incorporates emotional tone analysis as an essential complement for identifying hostile speech.

This contribution not only enhances moderation systems' ability to distinguish between ironic, ambiguous, or discriminatory messages but also promotes the creation of safer, more equitable digital environments.

Practical implementation using a RESTful API and its validation in a web environment demonstrated the technical feasibility of the proposal in real-world scenarios, consolidating a scalable, lightweight architecture adaptable to different platforms.

From a mathematical perspective, the **BACON-AR** framework showed that combining Bayesian calibration with asymmetric risk-based decision-making makes the system more consistent, sensitive, and reliable. The hazard function reached its minimum point at $t^* = 0.43$, achieving an optimal balance between false positives and false negatives when the cost of false negatives was twice that of false positives. This behavior reduced critical errors without affecting overall accuracy, making decisions more understandable, consistent, and fair. Thus, the framework makes a methodological contribution by connecting Bayesian decision theory with the practice of probabilistic calibration in real-world classification contexts.

Regarding the study's limitations, the system's performance decreased under high concurrency, leading to significant error rates when more than 300 users accessed it simultaneously. This finding highlights the need to migrate to more robust, distributed infrastructures that maintain availability and performance in production environments.

Furthermore, the reliance on data collected on specific platforms limits the model's ability to generalize across different cultural and linguistic contexts, posing a challenge for its large-scale application.

Looking ahead, the proposal is to delve deeper into optimization strategies for large-scale systems, while also exploring self-supervised and multilingual learning techniques that expand the model's capacity to adapt to diverse digital communities.

Likewise, it is pertinent to investigate the use of multimodal models that integrate text, audio, and image to detect hostility and emotions more comprehensively, and to apply explainable learning methodologies that improve the transparency and traceability of decisions. The aim is to move towards more inclusive, scalable, and reliable detection systems that effectively prevent hate speech and promote respectful, safe digital spaces for women and other vulnerable groups.

The *BACON-AR* framework approach, together with the proposed Web architecture, constitutes a significant advance towards the development of automated systems capable of analyzing human language with sensitivity, fairness, and statistical consistency, for the benefit of a safer digital coexistence.

In summary, this work offers a comprehensive proposal that combines technical soundness, mathematical rigor, and an ethical vision for the use of artificial intelligence.

 doi:10.20944/preprints202603.0228.v1

7. Future Work

As future work, first: we propose construct a parallel corpus that includes Spanish and its regional variants to capture the linguistic diversity of Latin American digital communities. This is because current English-trained models need to be expanded to support multilingual capabilities. This would involve developing dialect-aware embeddings that preserve local expressions and culturally specific manifestations of gender-based hostility, which current models fail to recognize. Such expansion would directly address the machine translation limitations identified during the API validation.

Second, the performance degradation observed under high-concurrency loads demands architectural improvements. Migrating the system from the current prototype to a distributed cloud infrastructure with auto-scaling capabilities would ensure production-level reliability. Containerization strategies and asynchronous request processing could significantly reduce latency while maintaining prediction quality under real-world traffic patterns. This infrastructure upgrade would enable the system to function as a viable pre-moderation tool for social platforms.

Third, extending beyond text-based analysis represents a natural progression. Online hate speech increasingly spreads through multimodal formats, such as memes and images, that bypass conventional filters. Developing a multimodal ensemble that integrates visual features through convolutional networks would address this evasion strategy. The stacking architecture validated in this study provides a flexible foundation for incorporating these additional modalities while preserving the emotional tone classification component.

Author Contributions: Conceptualization, Aymé Escobar Díaz and Walter Fuertes; Methodology, Aymé Escobar Díaz and Ricardo Rivadeneira; Software, Aymé Escobar Díaz and Ricardo Rivadeneira; Validation, Aymé Escobar Díaz, Ricardo Rivadeneira, Walter Fuertes and Washington Loza; Formal analysis, Aymé Escobar Díaz, Ricardo Rivadeneira, Walter Fuertes and Washington Loza; Investigation, Aymé Escobar Díaz and Ricardo Rivadeneira; Resources, Ricardo Rivadeneira; Data curation, Aymé Escobar Díaz and Ricardo Rivadeneira; Writing—original draft preparation, Aymé Escobar Díaz and Ricardo Rivadeneira; Writing—review and editing, Walter Fuertes and Washington Loza; Visualization, Aymé Escobar Díaz and Ricardo Rivadeneira; Supervision, Walter Fuertes and Washington Loza; Project administration, Walter Fuertes; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by the Universidad de las Fuerzas Armadas ESPE, in Sangolquí, Ecuador.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets analyzed in this study are openly available in Kaggle: Hate Speech Detection Curated Dataset https://9www.kaggle.com/datasets/waalbannyantudre/hate-speech-detection-curated-dataset, In addition, the dataset used in this research, HateXplain, is publicly accessible at: HateXplain https://github.com/hate-alert/HateXplain. The trained models generated in this study are publicly available on Hugging Face at the repository https://huggingface.co/aymeescobar; No new proprietary data were created or used in this study.

Acknowledgments: The authors express their sincere gratitude to the Universidad de las Fuerzas Armadas ESPE for the academic, technical, and institutional support provided for this research. Special recognition is due to the *Distributed Systems, Cybersecurity, and Content Research Group* of the Department of Computer Science for the use of the *High-Performance Research Laboratory* and its specialized software and hardware. During the preparation of this manuscript, the authors used OpenAI ChatGPT (GPT-5, 2025) very occasionally to assist with some text English style correction and LaTeX formatting. The authors have reviewed and edited the generated content and assume full responsibility for the content of this publication.

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

The funders (Universidad de las Fuerzas Armadas ESPE and its Finance Unit) had no role in the study design, the collection, analysis, or interpretation of the data, the drafting of the manuscript, or the decision to publish the results.

 doi:10.20944/preprints202603.0228.v1

Abbreviations

The following abbreviations are used in this manuscript:

PLN	Procesamiento de Lenguaje Natural
ML	Machine Learning (Aprendizaje Automático)
API	Application Programming Interface
EDA	Exploratory Data Analysis (Análisis Exploratorio de Datos)
ETL	Extract, Transform, Load (Extracción, Transformación y Carga)
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
SLR	Systematic Literature Review (Revisión Sistemática de Literatura)
F1	F1-Score (Precision harmonic measurement and recall.)
JSON	JavaScript Object Notation
CFN	Cost of false negative
CFP	Cost of False Positive
FN	False Negative
FP	False Positive
t^*	Threshold of t
I(.)	Indicator Function
BACON-AR	Bayesian Calibration and Optimal Design under Asymmetric Risk

References

1. Williams, B.; Onsman, A.; Brown, T. Exploratory factor analysis: A five-step guide for novices. *Australasian Journal of Paramedicine* **2010**, *8*. https://doi.org/10.33151/ajp.8.3.93.
2. Schmidt, A.; Wiegand, M. A Survey on Hate Speech Detection using Natural Language Processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media* **2017**, pp. 1–10. https://doi.org/10.18653/v1/W17-1101.
3. Fortuna, P.; Nunes, S. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys* **2018**, *51*, 1–30. https://doi.org/10.1145/3232676.
4. Stahelski, A.; Anderson, A.; Browitt, N.; Radeke, M. Facial Expressions and Emotion Labels Are Separate Initiators of Trait Inferences From the Face. *Frontiers in Psychology* **2021**, *12*, 749933. https://doi.org/10.3389/fpsyg.2021.749933.
5. Brown, A. What is hate speech? Part 1: The myth of hate. *Law and Philosophy* **2017**. https://doi.org/10.1007/s10982-017-9297-1.
6. Martins, R.; Gomes, M.; Almeida, J.; Novais, P.; Henriques, P. Hate speech classification in social media using emotional analysis. In Proceedings of the 2018 7th Brazilian Conference on Intelligent Systems (BRACIS). IEEE, 2018, pp. 265–270. https://doi.org/10.1109/BRACIS.2018.00019.
7. Founta, A.M.; Djouvas, C.; Chatzakou, D.; Leontiadis, I.; Blackburn, J.; Stringhini, G.; Vakali, A.; Sirivianos, M.; Kourtellis, N. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. In Proceedings of the Proceedings of ICWSM, 2018, pp. 491–500.
8. García-Díaz, J.A.; Cánovas-García, M.; Colomo-Palacios, R.; Valencia-García, R. Detecting misogyny in Spanish tweets: An approach based on linguistic features and word embeddings. *Future Generation Computer Systems* **2021**, *114*, 506–518. https://doi.org/10.1016/j.future.2020.08.032.
9. Jane, E.A. Misogyny Online: A Short (and Brutish) History. *SAGE Open* **2017**, *7*, 1–12. https://doi.org/10.1177/2158244016688793.
10. Siapera, E. Online misogyny as witch hunt: Primitive accumulation in the age of technocapitalism. In *Gender hate online*; Palgrave Macmillan, 2019; pp. 21–44. https://doi.org/10.1007/978-3-319-96226-9_2.
11. Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Rangel Pardo, F.M.; Rosso, P.; Sanguinetti, M. SemEval-2019 Task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In Proceedings of the Proceedings of the 13th International Workshop on Semantic Evaluation. Association for Computational Linguistics, 2019, pp. 54–63. https://doi.org/10.18653/v1/S19-2007.
12. Citron, D.K. *Hate Crimes in Cyberspace*; Harvard University Press, 2014.
13. Corazza, M.; Menini, S.; Cabrio, E.; Tonelli, S.; Villata, S. A multilingual evaluation for online hate speech detection. *ACM Transactions on Internet Technology* **2020**, *20*. https://doi.org/10.1145/3377323.

 doi:10.20944/preprints202603.0228.v1

14. Vidgen, B.; Derczynski, L. Directions in Abusive Language Training Data, a Systematic Review: Garbage In, Garbage Out. *PLoS ONE* **2020**, *15*, e0243300. https://doi.org/10.1371/journal.pone.0243300.
15. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019, [arXiv:cs.CL/1907.11692].
16. Andrade, R.O.; Fuertes, W.; Cazares, M.; Ortiz-Garcés, I.; Navas, G. An Exploratory Study of Cognitive Sciences Applied to Cybersecurity. *Electronics* **2022**, *11*. https://doi.org/10.3390/electronics11111692.
17. Bonilla, E.; Howard, D.; Oliveira, R.; Sejdinovic, D. Bayesian Adaptive Calibration and Optimal Design. In Proceedings of the Advances in Neural Information Processing Systems 37. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024, NeurIPS 2024, p. 56526–56551. https://doi.org/10.52202/079017-1800.
18. Sun, Z.; Song, D.; Hero, A.O. Minimum-risk recalibration of classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2023**, *45*, 9564–9577. https://doi.org/10.1109/TPAMI.2023.3278914.
19. Kelly, J.; Smyth, P. Variable-based calibration for machine learning classifiers. *Pattern Recognition* **2022**, *129*, 108754. https://doi.org/10.1016/j.patcog.2022.108754.
20. Araf, I.; Idri, A.; Chairi, I. Cost-sensitive learning for imbalanced medical data: A review. *Artificial Intelligence Review* **2024**. https://doi.org/10.1007/s10462-023-10652-8.
21. Gorwa, R.; Binns, R.; Katzenbach, C. Algorithmic Content Moderation: Technical and Political Challenges in the Automation of Platform Governance. *Big Data & Society* **2020**, *7*, 1–15. https://doi.org/10.1177/2053951720913654.
22. Escobar Díaz, A.; Rivadeneira, R.; Fuertes, W. Emotional Tone Detection in Hate Speech Using Machine Learning and NLP: Methods, Challenges, and Future Directions—A Systematic Review. *Applied Sciences* **2025**, *15*, 12686. https://doi.org/10.3390/app152312686.
23. Min, C.; Lin, H.; Li, X.; Zhao, H.; Lu, J.; Yang, L.; Xu, B. Finding hate speech with auxiliary emotion detection from self-training multi-label learning perspective. *Information Fusion* **2023**, *96*, 214–223. https://doi.org/https://doi.org/10.1016/j.inffus.2023.03.015.
24. Ramos, G.; Batista, F.; Ribeiro, R.; Fialho, P.; Moro, S.; Fonseca, A.; Guerra, R.; Carvalho, P.; Marques, C.; Silva, C. A comprehensive review on automatic hate speech detection in the age of the transformer. *Social Network Analysis and Mining* **2024**, *14*. https://doi.org/10.1007/s13278-024-01361-3.
25. Rodriguez, A.; Chen, Y.L.; Argueta, C. FADOHS: Framework for Detection and Integration of Unstructured Data of Hate Speech on Facebook Using Sentiment and Emotion Analysis. *IEEE Access* **2022**, *10*, 22400–22419. https://doi.org/10.1109/ACCESS.2022.3151098.
26. Kaminska, O.; Cornelis, C.; Hoste, V. Fuzzy rough nearest neighbour methods for detecting emotions, hate speech and irony. *Information Sciences* **2023**, *625*, 521–535. https://doi.org/10.1016/j.ins.2023.01.054.
27. Vallecillo-Rodríguez, M.E.; Plaza-del Arco, F.M.; Montejo-Ráez, A. Combining profile features for offensiveness detection on Spanish social media. *Expert Systems with Applications* **2025**, *272*, 126705. https://doi.org/10.1016/j.eswa.2025.126705.
28. Pan, R.; García-Díaz, J.A.; Valencia-García, R. Spanish MTLHateCorpus 2023: Multi-task learning for hate speech detection to identify speech type, target, target group and intensity. *Computer Standards & Interfaces* **2025**, *94*, 103990. https://doi.org/10.1016/j.csi.2025.103990.
29. Paul, J.; Mallick, S.; Mitra, A.; Roy, A.; Sil, J. Multi-modal Twitter Data Analysis for Identifying Offensive Posts Using a Deep Cross-Attention–based Transformer Framework. *ACM Trans. Knowl. Discov. Data* **2025**, *19*. https://doi.org/10.1145/3713077.
30. Keya, A.J.; Kabir, M.M.; Shammey, N.J.; Mridha, M.F.; Islam, M.R.; Watanobe, Y. G-BERT: An Efficient Method for Identifying Hate Speech in Bengali Texts on Social Media. *IEEE Access* **2023**, *11*, 79697–79709. https://doi.org/10.1109/ACCESS.2023.3299021.
31. Sasidhar, T.T.; B, P.; P, S.K. Emotion Detection in Hinglish(Hindi+English) Code-Mixed Social Media Text. *Procedia Computer Science* **2020**, *171*, 1346–1352. https://doi.org/10.1016/j.procs.2020.04.144.
32. Priya, P.; Firdaus, M.; Ekbal, A. A multi-task learning framework for politeness and emotion detection in dialogues for mental health counselling and legal aid. *Expert Systems with Applications* **2023**, *224*, 120025. https://doi.org/10.1016/j.eswa.2023.120025.
33. Nandi, P.; Sharma, S.; Chakraborty, T. SAFE-MEME: Structured Reasoning Framework for Robust Hate Speech Detection in Memes, 2024. https://doi.org/10.48550/ARXIV.2412.20541.
34. Chhabra, A.; Vishwakarma, D.K. MHS-STMA: Multimodal Hate Speech Detection via Scalable Transformer-Based Multilevel Attention Framework, 2024. https://doi.org/10.48550/ARXIV.2409.05136.

 doi:10.20944/preprints202603.0228.v1

35. Chhabra, A.; Vishwakarma, D.K. Multimodal hate speech detection via multi-scale visual kernels and knowledge distillation architecture. *Engineering Applications of Artificial Intelligence* **2023**, *126*, 106991. https://doi.org/10.1016/j.engappai.2023.106991.
36. Guo, K.; Hu, A.; Mu, J.; Shi, Z.; Zhao, Z.; Vishwamitra, N.; Hu, H. An Investigation of Large Language Models for Real-World Hate Speech Detection, 2024. https://doi.org/10.48550/ARXIV.2401.03346.
37. Barakat, B.; Jaf, S. Beyond Traditional Classifiers: Evaluating Large Language Models for Robust Hate Speech Detection. *Computation* **2025**, *13*, 196. https://doi.org/10.3390/computation13080196.
38. Piot, P.; Parapar, J. Decoding Hate: Exploring Language Models' Reactions to Hate Speech. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Association for Computational Linguistics, 2025, p. 973–990. https://doi.org/10.18653/v1/2025.naacl-long.45.
39. Calapaqui, G.; Guarderas, D.; Fuertes, W.; López, A.; Aules, H., Detection of Hate Speech on On-Line Social Platforms Using Machine Learning and Natural Language Processing – A Literature Review. In *CIST-2025*; 2026; pp. 441–452. https://doi.org/10.1007/978-3-032-10929-3_38.
40. Filho, T.M.S.; de Souto, M.C.; de Carvalho, A.C.P.L.F. Classifier calibration: A survey on how to assess and improve predicted class probabilities. *Machine Learning* **2023**, *112*, 5193–5229. https://doi.org/10.1007/s10994-023-06336-7.
41. Dimitriadis, T.; Gneiting, T.; Ziegel, M. Evaluating probabilistic classifiers: The triptych. *Pattern Recognition* **2024**, *150*, 110312. https://doi.org/10.1016/j.patcog.2023.110312.
42. Rella, C.; Vilar, J.M. Cost-sensitive thresholding over a two-dimensional decision region for fraud detection. *Information Sciences* **2024**, *660*, 119604. https://doi.org/10.1016/j.ins.2024.119604.
43. Komisarenko, V. Cost-sensitive classification with cost uncertainty: Do we need surrogate losses? *Machine Learning* **2025**. https://doi.org/10.1007/s10994-024-06634-8.
44. Uther, W.; Mladenić, D.; Ciaramita, M.; Berendt, B.; Kołcz, A.; Grobelnik, M.; Witbrock, M.; Risch, J.; Bohn, S.; Poteet, S.; et al., TF–IDF. In *Encyclopedia of Machine Learning*; Sammut, C.; Webb, G.I., Eds.; Springer US: Boston, MA, 2010; pp. 986–987. https://doi.org/10.1007/978-0-387-30164-8_832.
45. Salman, H.A.; Kalakech, A.; Steiti, A. Random Forest Algorithm Overview. *Babylonian Journal of Machine Learning* **2024**, *2024*, 69–79. https://doi.org/10.58496/bjml/2024/007.
46. Malik, S.; Harode, R.; Singh, A. XGBoost: A Deep Dive into Boosting (Introduction Documentation). ResearchGate Preprint, 2020. Available online: https://doi.org/10.13140/RG.2.2.15243.64803 (accessed on 27 August 2025), https://doi.org/10.13140/RG.2.2.15243.64803.
47. Phelps, N.; Lizotte, D.J.; Woolford, D. Using Platt's Scaling for Calibration After Undersampling: Limitations and How to Address Them. *arXiv preprint arXiv:2410.18144* **2024**. https://doi.org/10.48550/arXiv.2410.18144.
48. Pineau, J.; Vincent-Lamarre, P.; Sinha, K.; Lariviere, V.; Beygelzimer, A.; d'Alche Buc, F.; Fox, E.; Larochelle, H. Improving Reproducibility in Machine Learning Research: A Report from the NeurIPS 2019 Reproducibility Program. *Journal of Machine Learning Research* **2021**, *22*, 1–20.
49. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
50. Teodorescu, V.; Obreja Brașoveanu, L. Assessing the Validity of k-Fold Cross-Validation for Model Selection: Evidence from Bankruptcy Prediction Using Random Forest and XGBoost. *Computation* **2025**, *13*, 127. https://doi.org/10.3390/computation13050127.
51. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On Calibration of Modern Neural Networks. In Proceedings of the Proceedings of the 34th International Conference on Machine Learning. PMLR, 2017, Vol. 70, *Proceedings of Machine Learning Research*, pp. 1321–1330. https://doi.org/10.48550/arXiv.1706.04599.
52. Zadrozny, B.; Elkan, C. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In Proceedings of the Proceedings of the 18th International Conference on Machine Learning. Morgan Kaufmann, 2001, pp. 609–616.
53. Dimitriadis, T.; Gneiting, T.; Ziegel, M. Stable reliability diagrams for probabilistic classifiers. *Proceedings of the National Academy of Sciences* **2021**, *118*, e2016191118. https://doi.org/10.1073/pnas.2016191118.
54. Al-Hashedi, M.; Soon, L.K.; Goh, H.N.; Lim, A.; Eu-Gene, S. Cyberbullying Detection Based on Emotion. *IEEE Access* **2023**, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2023.3280556.
55. Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; Galstyan, A. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys* **2021**, *54*, 1–35. https://doi.org/10.1145/3457607.

 doi:10.20944/preprints202603.0228.v1