ISSN 2007-9737

Minimalist Machine Learning with Metaheuristic Optimization for Explainable Spam Filtering

Jorge Alberto Pacheco-Senard[1], Mailyn Moreno-Espino[2,3], Cornelio Yáñez-Márquez[1,*],
Yenny Villuendas-Rey[1], Oscar Camacho-Nieto[1]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Universidad Complutense de Madrid,
Facultad de Informática,
Spain

[3] Universidad Complutense de Madrid,
Instituto de Tecnología del Conocimiento,
Spain

mailymor@ucm.es, {jpachecos2024, cyanez}@cic.ipn.mx, {yvilluendasr, ocamacho}@ipn.mx

Abstract. Spam detection remains a challenging task in text classification due to the dynamic nature of unsolicited messages and the lack of transparency in conventional machine learning models. This paper proposes a family of lightweight and interpretable classifiers based on the *Minimalist Machine Learning* (MML) paradigm integrated with metaheuristic optimization techniques. Three variants (*MML + Random Search*, *MML + Hill Climb*, and *MML + Simulated Annealing*) were implemented and evaluated on the SMS Spam Corpus v.0.1 using a hybrid lexical–semantic representation that combines BM25 and Word2Vec embeddings. Each model was designed to select the most discriminative lexical–semantic features from the feature matrix, optimizing class separability through an objective function based on the Intra-Class Correlation Coefficient (ICC). Experimental results under Leave-One-Out Cross-Validation (LOOCV) demonstrate that the *MML + Simulated Annealing* variant achieved the best overall performance (Balanced Accuracy = 0.9327, F1-score = 0.9014, MCC = 0.8700), yielding results statistically comparable to a linear SVM baseline according to the Wilcoxon paired test. These findings highlight that metaheuristic-enhanced MML models can achieve competitive performance while maintaining full interpretability. Future work will extend these models to sentiment analysis, AI-generated text detection, and hybrid transformer–MML architectures to combine transparency with deep semantic understanding. Given the increasing demand for transparent and responsible AI in communication systems, this study contributes to the development of interpretable and lightweight spam filtering mechanisms.

Keywords. Spam detection, minimalist machine learning, metaheuristic optimization, explainable AI, text classification.

1 Introduction

Electronic communication has become an indispensable component of modern life; however, it faces a critical challenge: the exponential growth of unsolicited and malicious messages, commonly known as spam. Recent estimates indicate that spam messages account for nearly 50% of all global email traffic, causing substantial waste of storage and bandwidth, in addition to serving as a vector for phishing, malware distribution, and identity theft [14]. The increasing sophistication of spam campaigns and the diversity

doi: 10.13053/CyS-29-4-6096

of textual patterns make the task of automatic spam detection a complex and continuously evolving problem.

Over the past two decades, numerous machine learning techniques have been applied to address this issue. Classical approaches such as Naïve Bayes, Decision Trees, k-Nearest Neighbors (KNN), Logistic Regression, Random Forest, and Support Vector Machines (SVM) have demonstrated promising results in the classification of spam and legitimate messages (ham) [13, 4, 15, 14, 5]. These algorithms typically rely on statistical representations of textual data, such as Bag-of-Words or TF-IDF weighting, to model the discriminative patterns within email corpora. More recently, the advent of deep learning and natural language processing (NLP) models (particularly transformers such as BERT) has enabled the automatic extraction of contextual features, achieving near-human performance in text classification tasks [16, 19].

Despite these advances, most high-performing models remain *black boxes*, offering limited transparency into their decision-making processes. This opacity hinders interpretability and reproducibility, both of which are essential for security-sensitive applications such as spam filtering [18]. Furthermore, deep architectures like convolutional or transformer-based networks require substantial computational resources and large labeled datasets, which may not always be available in real-world scenarios. This work therefore explores a minimalist and interpretable approach, supported by metaheuristic optimization.

The *Minimalist Machine Learning* (MML) paradigm, recently proposed by Yáñez-Márquez and colleagues, emerges as a response to the growing complexity and opacity of contemporary algorithms [18]. MML is predicated on the principles of minimal computational complexity and maximal explainability. The central idea of this approach is to transform high-dimensional data into two-dimensional representations through the application of simple statistical measures (e.g., the mean and standard deviation of selected attributes). This process enables geometric visualization and linear separability of classes. The paradigm has exhibited competitive performance while maintaining transparency and ease of interpretation, as evidenced in the extant literature [17].

Concurrently, semantic embedding models and co-occurrence matrices (e.g., Word2Vec and BM25) have emerged as the foundational framework for contemporary textual representation. These models are capable of capturing both statistical and semantic relationships among words; however, they frequently result in high-dimensional feature spaces, which can lead to an increase in computational cost and redundancy [11]. To address this dimensionality issue, metaheuristic optimization techniques (e.g., Particle Swarm Optimization (PSO), Random Search, Hill Climb, or Simulated Annealing) have been widely adopted to perform feature selection and model tuning in an efficient manner [2]. Such approaches enhance generalization by balancing exploration and exploitation of the search space, thereby effectively identifying the most relevant attributes for classification.

In this context, we propose a lightweight and interpretable spam detection framework that integrates lexical and semantic representations (BM25 and Word2Vec) with the Minimalist Machine Learning paradigm, enhanced through metaheuristic optimization based on Simulated Annealing. This approach aims to achieve a transparent and resource-efficient solution while maintaining competitive classification performance in imbalanced datasets.

2 Related Work

Over the last two decades, the problem of spam detection has been extensively studied, giving rise to a wide variety of machine learning approaches ranging from traditional classifiers to deep neural architectures. Early studies applied statistical models such as Naïve Bayes, Decision Trees, k-Nearest Neighbors (KNN), and Logistic Regression, achieving competitive accuracy rates in benchmark datasets such as Spambase and the SMS Spam Corpus [14, 13, 4]. These models operate on handcrafted features extracted through lexical frequency analysis and rely primarily on the probabilistic relationships among words to distinguish spam from legitimate messages.

Among traditional classifiers, Naïve Bayes remains one of the most popular due to its simplicity and robustness under limited data conditions. It models the conditional probability of word occurrences assuming feature independence, making it computationally efficient for text classification tasks [14]. Support Vector Machines (SVM), on the other hand, have demonstrated strong performance in binary classification scenarios by maximizing the separation margin between spam and ham classes. Comparative studies show that SVM and Decision Trees often outperform probabilistic methods in terms of precision and F1-score [4]. Nonetheless, these models depend heavily on the quality of the feature representation and typically lack interpretability from a linguistic perspective.

Feature representation has evolved from simple frequency-based models to more sophisticated lexical and semantic embeddings. The classical TF-IDF (Term Frequency–Inverse Document Frequency) scheme has long been the standard for weighting word importance in documents. However, recent advances introduced the Okapi BM25 model, which improves upon TF-IDF by considering term saturation and document length normalization [11]. BM25 is particularly useful in ranking and retrieval-based tasks but also serves as a solid foundation for spam classification when integrated with machine learning algorithms.

In parallel, distributed word representations (commonly known as embeddings) have revolutionized text processing. Word2Vec [3] and similar models learn vector representations that capture semantic and syntactic relationships between words by analyzing their co-occurrence contexts. Unlike frequency-based approaches, embeddings represent words in continuous vector spaces, enabling the computation of semantic similarity through vector algebra. These representations have been successfully applied to spam filtering, where semantic relationships such as “offer–discount” or “bank–account” help identify deceptive content more effectively than purely lexical features.

More recent works leverage deep learning architectures, particularly recurrent and transformer-based networks, to model long-term dependencies and contextual information in textual data. The introduction of BERT (Bidirectional Encoder Representations from Transformers) marked a paradigm shift by incorporating bidirectional context understanding and transfer learning for text classification [16]. Spam detectors built upon BERT have achieved accuracies exceeding 98% across multiple corpora, including Enron, Ling-Spam, and SpamAssassin. Nevertheless, such models demand extensive computational resources and large annotated datasets, and their internal representations remain opaque, limiting their interpretability in critical applications.

To conclude, the evolution of spam detection methods reveals an ongoing trade-off between accuracy, computational efficiency, and interpretability. Conventional machine learning algorithms are characterized by their expeditious and dependable classification capabilities; however, they frequently encounter challenges in accurately capturing the intricate semantic relationships inherent in textual data. Conversely, neural and transformer-based models have demonstrated remarkable accuracy by leveraging contextual embeddings and extensive corpora. However, their intricate nature often leads to opaque decision boundaries and substantial computational demands. Consequently, the research community continues to seek approaches that balance semantic understanding, scalability, and transparency for more effective and trustworthy spam filtering systems.

3 Minimalist Machine Learning Paradigm

The growing demand for interpretable artificial intelligence has led to the emergence of the *Minimalist Machine Learning* (MML) paradigm, a framework that seeks to balance predictive performance with transparency and computational simplicity. Proposed by Yáñez-Márquez and collaborators [18], MML arises as a response to the increasing complexity of contemporary learning

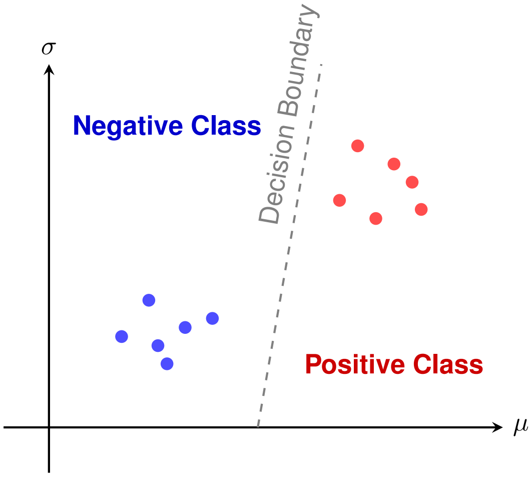


Fig. 1. Conceptual representation of the Minimalist Machine Learning (MML) paradigm

models, such as deep neural networks and large transformer architectures, which, despite their accuracy, often behave as opaque *black boxes*.

Unlike traditional models that rely on high-dimensional representations and numerous hyperparameters, MML advocates a minimalist philosophy: reducing both data dimensionality and algorithmic complexity while maintaining interpretability. The central premise of MML is that any pattern recognition problem can be geometrically represented in a low-dimensional space (typically two dimensions) through the use of simple statistical descriptors such as the arithmetic mean and standard deviation of the feature set [18, 17].

In practice, each instance in a dataset is transformed into a two-dimensional point (μ, σ), where μ represents the mean of the selected features and σ their standard deviation. This transformation allows the classifier to visualize and separate classes directly within a Cartesian plane, producing decision boundaries that are both mathematically simple and intuitively explainable. Figure 1 illustrates this conceptual reduction, in which patterns from different classes can be distinguished through linear or near-linear separations in the two-dimensional space.

The MML paradigm is characterized by its simplicity, a quality that offers several advantages.

First, the explainability of the model stems from the fact that the decision boundary can be visualized and described through straightforward geometric reasoning, which enables users to understand how individual instances are classified.

Second, the computational expense associated with training and inference is significantly reduced, as MML circumvents the complexities inherent in parameter optimization and backpropagation processes.

In conclusion, the paradigm under discussion promotes model transparency and reproducibility. These characteristics are particularly relevant in domains where interpretability and ethical accountability are essential [17].

Despite its simplicity, MML has demonstrated competitive performance across diverse applications, including image analysis, medical diagnosis, and text classification. In each case, the paradigm's minimalist transformation preserves essential class-discriminative information while eliminating redundancy and noise from the original feature space. These results suggest that MML can serve as a foundation for the development of efficient, interpretable, and domain-agnostic classifiers, making it a promising alternative to traditional machine learning models for tasks such as spam detection.

4 Heuristic and Metaheuristic Strategies in Computational Optimization

Decision-making and optimization problems often require finding effective solutions within vast and complex search spaces where traditional analytical or deterministic methods become impractical. To address this challenge, two complementary strategies have emerged: *heuristics* and *metaheuristics*. While both aim to obtain satisfactory solutions within a reasonable computational time, they differ in their levels of generality, adaptability, and scope of application.

4.1 Heuristics

The term *heuristic* originates from the Greek word “heuriskein,” meaning “to discover.” In its broadest sense, a heuristic represents a practical strategy or rule of thumb used to produce sufficiently good solutions with limited time, information, or computational resources [7]. Heuristics are typically problem-specific and exploit domain knowledge to guide the search process toward promising regions of the solution space.

In decision theory and cognitive science, heuristics are often described as mental shortcuts that simplify complex decision-making processes by reducing the amount of information considered [12]. Although they accelerate the search for acceptable solutions, heuristics do not guaranty optimality and may introduce biases or suboptimal outcomes. Nevertheless, they are highly valuable in real-world contexts, where perfect information or exhaustive computation is unattainable.

Algorithmically, heuristics are designed to exploit the structural properties of a given problem. For instance, in combinatorial optimization, a heuristic might iteratively improve an initial solution through local modifications, as in *hill climbing* or *greedy search*. These algorithms are simple, efficient, and effective for specific problem types, but they often risk becoming trapped in local optima. Hence, while heuristics provide a foundation for efficient problem solving, they may lack the flexibility needed to adapt to broader classes of optimization problems.

4.2 Metaheuristics

The concept of *metaheuristic* (literally “beyond heuristic”) refers to a higher-level search strategy designed to guide and control subordinate heuristics in exploring complex search spaces [1]. Metaheuristics provide a general framework applicable to a wide range of problems, including those that are non-linear, non-differentiable, or NP-hard. They are characterized by stochastic mechanisms that balance *exploration* (searching for new regions of the solution space) and *exploitation* (refining known good solutions) [12].

Unlike conventional heuristics, metaheuristics are not tied to a specific problem domain. Instead, they rely on abstract principles (often inspired by nature, physics, or human behavior) to iteratively improve candidate solutions. Examples of well-known metaheuristics include Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Tabu Search (TS), and Simulated Annealing (SA) [1, 8, 9]. Each of these algorithms employs unique mechanisms of adaptation, cooperation, or randomization to escape local optima and approximate the global optimum.

Metaheuristics can be broadly categorized according to several criteria:

– **Nature-inspired vs. Non-nature-inspired:** Nature-inspired algorithms mimic biological or physical processes (e.g., GA, PSO, ACO), while non-nature-inspired methods, like Tabu Search and Simulated Annealing rely on abstract optimization principles.

– **Single-solution vs. Population-based:** Single-solution approaches, such as Simulated Annealing or Hill Climbing, iteratively refine one candidate solution, whereas population-based algorithms (GA, PSO) evolve multiple candidates simultaneously.

– **Deterministic vs. Stochastic:** Deterministic algorithms follow fixed rules, whereas stochastic algorithms introduce random variation to enhance exploration.

The power of metaheuristics lies in their capacity to handle uncertainty, non-linearity, and multi-modality. They have been successfully applied in diverse fields, including engineering design, scheduling, financial optimization, and machine learning [1]. In the context of artificial intelligence, metaheuristics have also been employed for feature selection, hyperparameter tuning, and model optimization, often achieving superior results compared to traditional gradient-based methods.

Heuristics and metaheuristics represent complementary paradigms for solving optimization problems. Heuristics provide domain-specific efficiency and fast convergence, while metaheuristics offer robustness and adaptability across problem types. Their hybridization (combining the speed of heuristics with the global search ability of metaheuristics) has become a central research trend, enabling the design of flexible, general-purpose algorithms capable of tackling complex real-world optimization challenges.

5 Experimental Setup

This section details the experimental framework designed to evaluate the performance of the proposed models under the Minimalist Machine Learning (MML) paradigm. The experiments were conducted using a publicly available benchmark corpus of short message service (SMS) texts.

The evaluation procedure follows a reproducible design, comprising five stages: (i) corpus description, (ii) preprocessing and feature representation, (iii) Baseline Models, (iv) Proposed Models, and (v) Evaluation Metrics. Each step is described in the following subsections.

5.1 Corpus Description

The experiments were carried out using the *SMS Spam Corpus v.0.1* [6], a well-established corpus for research on spam filtering in short message communications. The corpus consists of a collection of English SMS messages manually tagged as legitimate (*ham*) or unsolicited (*spam*). It integrates three primary data sources, all publicly available and curated for academic research purposes.

- **Jon Stevenson Corpus (JSC):** A set of 202 legitimate messages obtained from a publicly accessible online collection attributed to Jon Stevenson. These messages contain only text content and represent informal, user-generated communication.

- **NUS SMS Corpus (NSC):** A subset of approximately 800 legitimate messages originally collected at the Department of Computer Science, National University of Singapore [10]. The messages were voluntarily contributed by university students and primarily reflect the linguistic patterns of everyday English communication in Singapore.

- **Grumbletext Collection:** A compilation of 322 SMS spam messages extracted manually from the *Grumbletext* forum (UK), a platform where users report unsolicited SMS advertising or scams. The extraction required a detailed manual review to isolate the actual spam text from user comments.

The resulting corpus comprises a total of 1,324 messages, including 1,002 legitimate (ham) and 322 spam instances. This corresponds to a class distribution of approximately 75.6% ham and 24.4% spam, yielding an *Imbalance Ratio (IR)* of 3.11. Such a level of imbalance is moderate and closely reflects real-world communication patterns, where legitimate messages vastly outnumber spam. All messages are stored as plain text, with no supplementary metadata, making the corpus particularly suitable for both lexical and semantic text analysis.

The *SMS Spam Corpus v.0.1* has been widely employed in the literature as a benchmark for short-text spam filtering. Cormack et al. [6] first introduced the corpus in their seminal work on SMS spam detection, highlighting its compactness and linguistic diversity. It has since been adopted in various studies on natural language processing, feature engineering, and machine learning for text classification tasks.

Given its accessibility, balanced complexity, and representativeness of authentic user communications, this corpus was selected as the experimental foundation for evaluating the proposed MML-based spam detection models.

5.2 Preprocessing and Feature Representation

The preprocessing stage plays a crucial role in preparing raw textual data for subsequent feature extraction and model training. Given the informal and heterogeneous nature of SMS communication, a customized preprocessing pipeline was implemented to normalize text, reduce noise, and enhance semantic coherence. The process is illustrated in Figure 4 and comprises the following sequential steps:

- **Lowercasing:** All messages were converted to lowercase to eliminate case sensitivity and unify word representations, ensuring that identical words (e.g., *Free* and *free*) are treated equivalently.
- **Tokenization:** A customized tokenizer was developed to handle short messages and contractions common in SMS texts. This tokenizer segments each message into discrete tokens, while preserving punctuation-based semantics relevant for spam detection (e.g., “win!!!” or “call now”).
- **Contraction Expansion:** Common contractions (e.g., *don’t* → *do not*, *I’ll* → *I will*) were expanded to restore complete lexical forms, thus improving semantic clarity and supporting more accurate vector representations.
- **Stopword Removal:** Non-informative words such as articles, prepositions, and common auxiliaries were removed using a domain-adapted stopword list. This step reduces the dimensionality of the feature space by eliminating terms with low discriminative power.
- **Lemmatization:** Tokens were transformed into their canonical (lemma) forms using a lemmatizer based on part-of-speech tagging. Lemmatization contributes to the semantic normalization of words, grouping different inflections under a single base concept.
- **Stemming:** The Porter stemming algorithm was subsequently applied to reduce word variants that may not be covered by lemmatization (e.g., *connect*, *connected*, *connection* → *connect*). The combined use of lemmatization and stemming increases linguistic compactness.
- **Filtering:** Non-alphabetic characters, numeric values, and single-character tokens were removed to further reduce noise and ensure a clean textual representation.

Each preprocessing component was encapsulated as a modular function within the pipeline, ensuring reproducibility and flexibility for future corpus extensions. The processed tokens were then transformed into numerical feature vectors using a hybrid lexical–semantic representation that combines BM25 and Word2Vec embeddings.

5.2.1 Lexical and Semantic Representation

Following text normalization, two complementary feature extraction techniques were applied:

- **BM25 Weighting:** The Okapi BM25 model [11] was employed to assign statistical relevance scores to each term based on its frequency and inverse document frequency across the corpus. Unlike static TF-IDF, BM25 introduces saturation parameters that mitigate bias from excessively frequent terms and normalize for message length. Figure 2 illustrates a co-ocurrence matrix of BM25.
- **Word2Vec Embeddings:** Semantic information was captured using Word2Vec [3], which generates distributed word representations by learning contextual co-occurrence relationships. Each token was mapped to a dense vector in a continuous semantic space, allowing the capture of latent similarities between words (e.g., *“prize”* and *“reward”*). In Figure 3 show an embedding created by this algorithm.

To combine both perspectives, the BM25-weighted term scores were concatenated with their corresponding Word2Vec embeddings, forming a unified feature matrix. This fusion enriched the representation by integrating

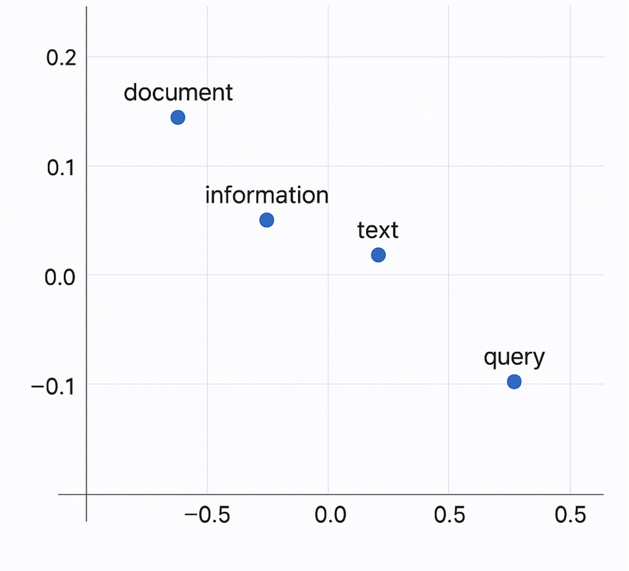
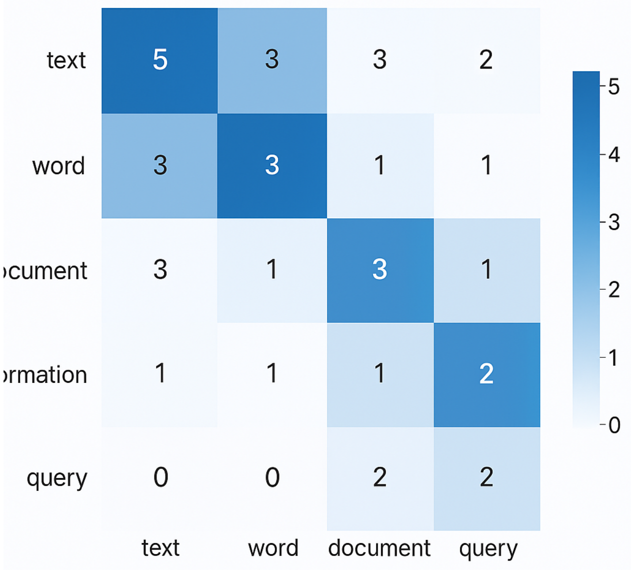


Fig. 2. Co-ocurrence matrix of BM25 algorithm



	text	word	document	query
text	5	3	3	2
word	3	3	1	1
ocument	3	1	3	1
ormation	1	1	1	2
query	0	0	2	2

Fig. 3. Embeddings of Word2Vec algorithm

statistical relevance with semantic context, thereby enhancing the discriminative power of the subsequent Minimalist Machine Learning classifiers.

5.3 Baseline Models

To provide a comparative benchmark for the proposed Minimalist Machine Learning (MML) models, three well-established algorithms from the state of the art were implemented. These baseline models are widely recognized for their effectiveness in text classification tasks, including spam detection [14, 4, 13]. All models were trained and evaluated under identical experimental conditions and using the same feature representation (BM25–Word2Vec fusion) to ensure fairness and reproducibility.

- **Logistic Regression (LR):** A linear probabilistic classifier that models the likelihood of message labels using a logistic function. The model was trained with a learning rate of 0.0001 and optimized using stochastic gradient descent. Logistic Regression is known for its interpretability and robustness in high-dimensional spaces, making it a standard baseline for text-based binary classification problems.

- **Support Vector Machine (SVM):** A margin-based classifier that seeks the optimal hyperplane separating spam and ham messages. A linear kernel was used to accommodate the sparsity of the text vectors, following established best practices in text categorization [4]. The SVM model is particularly effective in handling non-probabilistic decision boundaries and has shown consistent performance across spam filtering benchmarks.

- **Naïve Bayes (NB):** A probabilistic classifier based on Bayes' theorem with the assumption of conditional independence between features. The Multinomial variant was applied, as it is well suited for word frequency data and short text messages [14]. Naïve Bayes serves as a lightweight yet competitive baseline, offering high training speed and robustness against noisy data.

Each baseline model was tuned using Leave-One-Out Cross-Validation (LOOCV) to minimize overfitting and to ensure a robust and

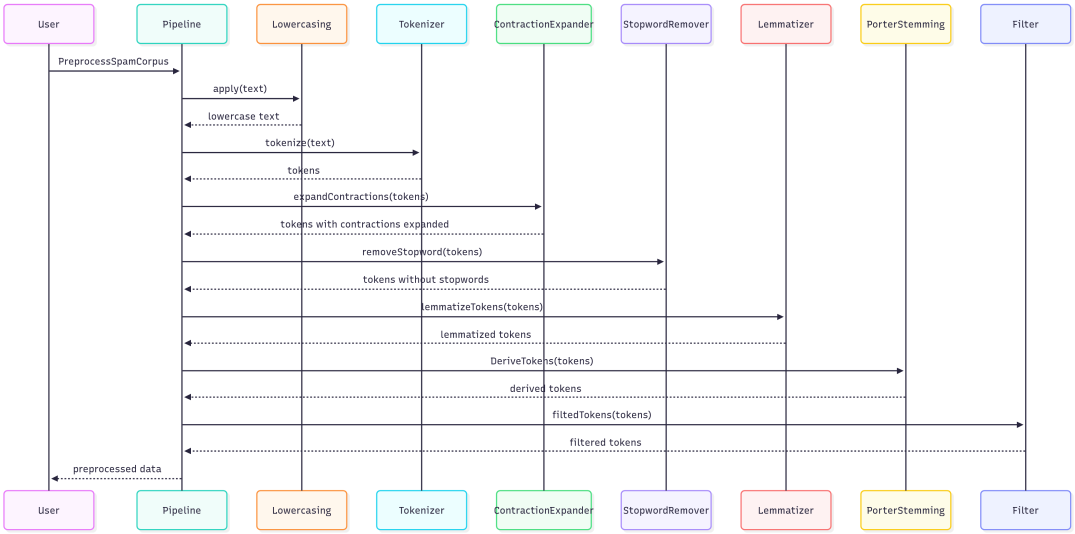


Fig. 4. Preprocessing pipeline for the SMS Spam corpus

unbiased generalization across all samples. This exhaustive validation strategy evaluates each message as a test instance exactly once, providing a precise estimation of model performance on small and moderately imbalanced datasets. The performance of these algorithms served as a benchmark to assess the effectiveness of the proposed MML-based models introduced in the subsequent section.

5.4 Proposed Models

Building upon the theoretical foundation of the Minimalist Machine Learning (MML) paradigm, three generalized variants were developed to evaluate the integration of different metaheuristic optimization techniques. These variants aim to identify the most informative subset of features from the hybrid BM25–Word2Vec representation while preserving the interpretability and efficiency that characterize MML.

5.4.1 Objective and Design

The central goal of the proposed models is to find the smallest subset of features that maximizes class separability and overall classification accuracy. From the combined feature matrix, the ten most relevant attributes were selected through a filtering process based on their discriminative power. Each proposed variant applies a distinct metaheuristic search strategy (Random Search, Hill Climb, and Simulated Annealing) to optimize feature selection and model parameters.

Unlike the original formulation of MML, which performs classification using centroid means, the generalized MML version proposed here employs an objective function derived from intra-class variance. The model seeks to minimize intra-class dispersion while maximizing inter-class separability, thus enhancing robustness in noisy and imbalanced datasets. The classification decision is guided by the *Intra-Class Correlation Coefficient (ICC)*, a measure that quantifies the homogeneity of instances within the same class:

$$\text{ICC} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{\text{class}})^2, \tag{1}$$

where x_i represents each sample feature within a given class, μ_{class} is the mean vector of that class, and n is the number of instances. Lower ICC values indicate greater compactness within classes and, consequently, higher discriminability between categories.

Each model transforms the text to be classified into a representative vector summarizing its central idea. This representation enables precise and interpretable classification decisions without relying on complex deep learning architectures. The process is illustrated in Figure 5

5.4.2 Model Variants

- **Version 1 (MML + Random Search):** In this baseline variant, the Random Search algorithm explores the solution space by randomly sampling subsets of features and evaluating their objective scores using the ICC-based criterion. Although it lacks memory of previous iterations, Random Search provides a stochastic reference point that establishes the lower bound for optimization performance.

- **Version 2 (MML + Hill Climb):** The Hill Climb variant performs iterative local optimization starting from a randomly initialized feature subset. At each iteration, a random neighbor is generated by slightly modifying the current subset, and the move is accepted if it improves the ICC score. To avoid stagnation in local optima, the neighborhood function introduces controlled randomness in feature exchanges between iterations. This approach achieves fast convergence and provides a solid balance between exploration and exploitation.

- **Version 3 (MML + Simulated Annealing):** The third variant incorporates the Simulated Annealing (SA) algorithm to overcome the limitations of local search. Inspired by the metallurgical annealing process, SA introduces a probabilistic acceptance criterion that occasionally allows worse solutions to escape local minima. The acceptance probability is governed by the Boltzmann distribution:

  $$P(\Delta E) = e^{-\frac{\Delta E}{T}}, \tag{2}$$

  where ΔE denotes the change in the objective function and T represents the system temperature. The initial temperature was set to $T_0 = 10.0$, with a decay factor of 0.99 applied at each iteration, progressively reducing randomness as the algorithm converges. This mechanism ensures a balanced trade-off between global exploration and fine-grained local refinement.

5.4.3 Discussion

All three variants share the same minimalist classification foundation but differ in their search dynamics. Random Search establishes a performance baseline through stochastic exploration, Hill Climb refines local solutions efficiently, and Simulated Annealing introduces a probabilistic acceptance mechanism that facilitates escaping local optima and approaching near-global solutions. Each optimization strategy is evaluated not only by its performance but also by its impact on interpretability and feature compactness.Together, these variants demonstrate that incorporating metaheuristic optimization into MML can significantly enhance its capacity to model high-dimensional lexical–semantic data while maintaining full explainability and low computational cost.

5.5 Evaluation Metrics

In binary classification problems such as spam detection, model performance is typically summarized using a *confusion matrix*. This matrix (Figure 6) compares the predicted labels with the true class labels, providing four fundamental outcomes:

- **True Positives (TP):** spam messages correctly identified as spam,

- **True Negatives (TN):** legitimate messages correctly classified as non-spam,

- **False Positives (FP):** legitimate messages incorrectly classified as spam, and

- **False Negatives (FN):** spam messages misclassified as legitimate.

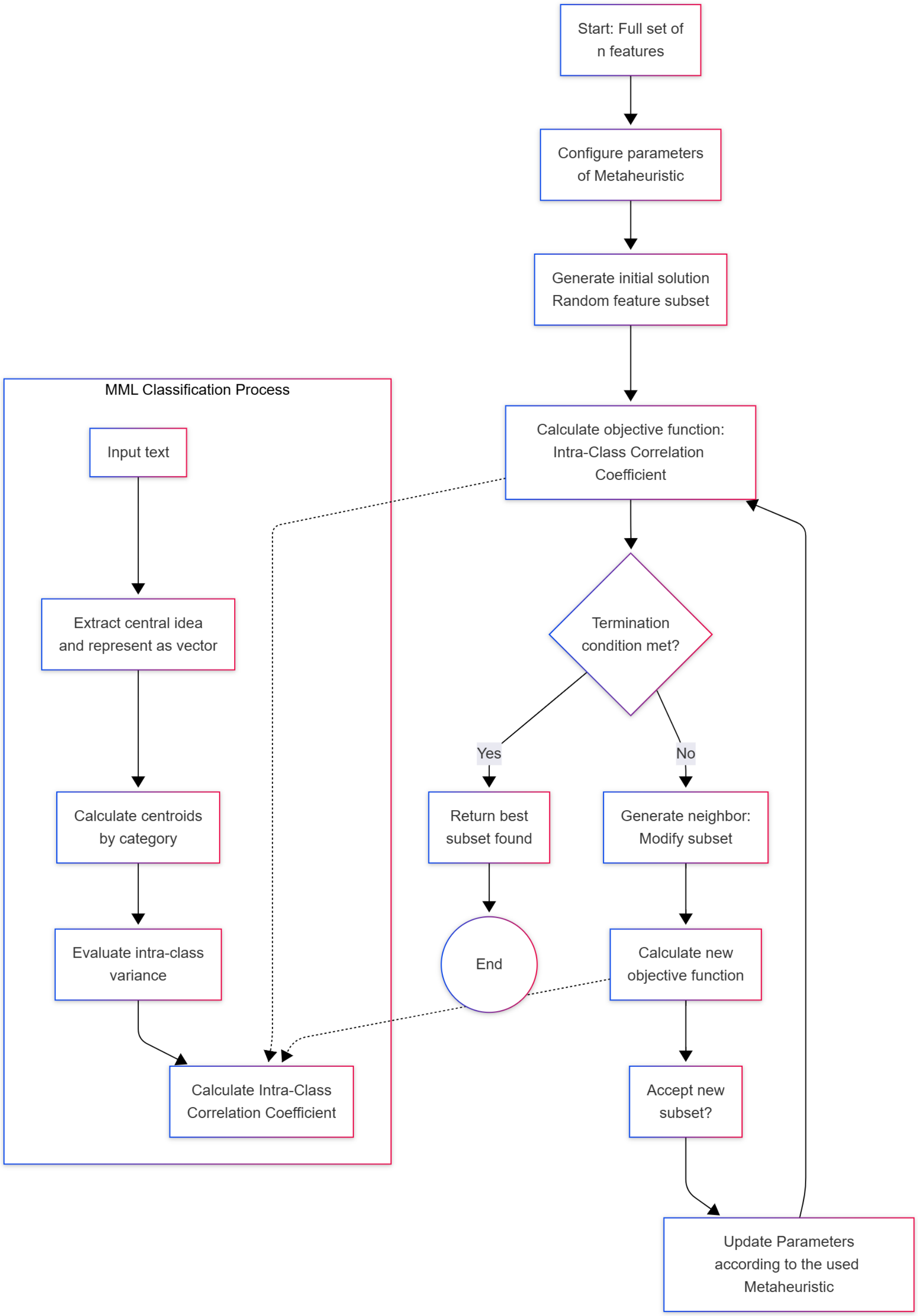

Start: Full set of n features
Configure parameters of Metaheuristic
Generate initial solution Random feature subset
MML Classification Process
Input text
Extract central idea and represent as vector
Calculate centroids by category
Evaluate intra-class variance
Calculate Intra-Class Correlation Coefficient
Calculate objective function: Intra-Class Correlation Coefficient
Termination condition met?
Yes
No
Return best subset found
End
Generate neighbor: Modify subset
Calculate new objective function
Accept new subset?
Update Parameters according to the used Metaheuristic

Fig. 5. Generalized workflow of the proposed MML-based models integrated with metaheuristic optimization

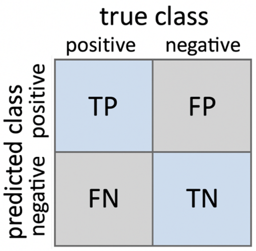


Fig. 6. Binary confusion matrix

Given the class imbalance present in the SMS Spam Corpus (*Imbalance Ratio* = 3.11), the overall accuracy metric was not used, as it can be misleading when one class dominates the dataset. Instead, a set of complementary metrics that capture sensitivity, specificity, precision, balance, and correlation were employed to provide a more reliable assessment of classification performance. All metrics were computed under the Leave-One-Out Cross-Validation (LOOCV) validation method.

5.5.1 Recall (Sensitivity)

Recall measures the model's ability to correctly identify spam messages among all actual spam instances:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3}$$

5.5.2 Specificity

Specificity quantifies the proportion of legitimate (ham) messages correctly identified as non-spam. It complements Recall by focusing on the true negative rate:

$$\text{Specificity} = \frac{TN}{TN + FP}. \tag{4}$$

5.5.3 Precision

Precision, or Positive Predictive Value, measures the proportion of messages predicted as spam that are actually spam. It reflects the classifier's reliability in producing positive predictions:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{5}$$

5.5.4 F1-Score

The F1-Score represents the harmonic mean between Precision and Recall, providing a single metric that balances false positives and false negatives. The harmonic mean formulation prevents dominance by large individual values and emphasizes the joint performance of both metrics:

$$\text{F1-Score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}. \tag{6}$$

5.5.5 Balanced Accuracy

Balanced Accuracy accounts for the uneven class distribution by averaging the recall of both positive and negative classes. This metric ensures an unbiased measure of overall classification capability in imbalanced datasets:

$$\text{Balanced Accuracy} = \frac{\text{Recall} + \text{Specificity}}{2}. \tag{7}$$

5.5.6 Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (MCC) provides a robust statistical evaluation of binary classification quality, considering all four elements of the confusion matrix. It returns a value in the range $[-1, +1]$, where $+1$ indicates perfect classification, 0 random prediction, and -1 complete disagreement between prediction and reality:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{8}$$

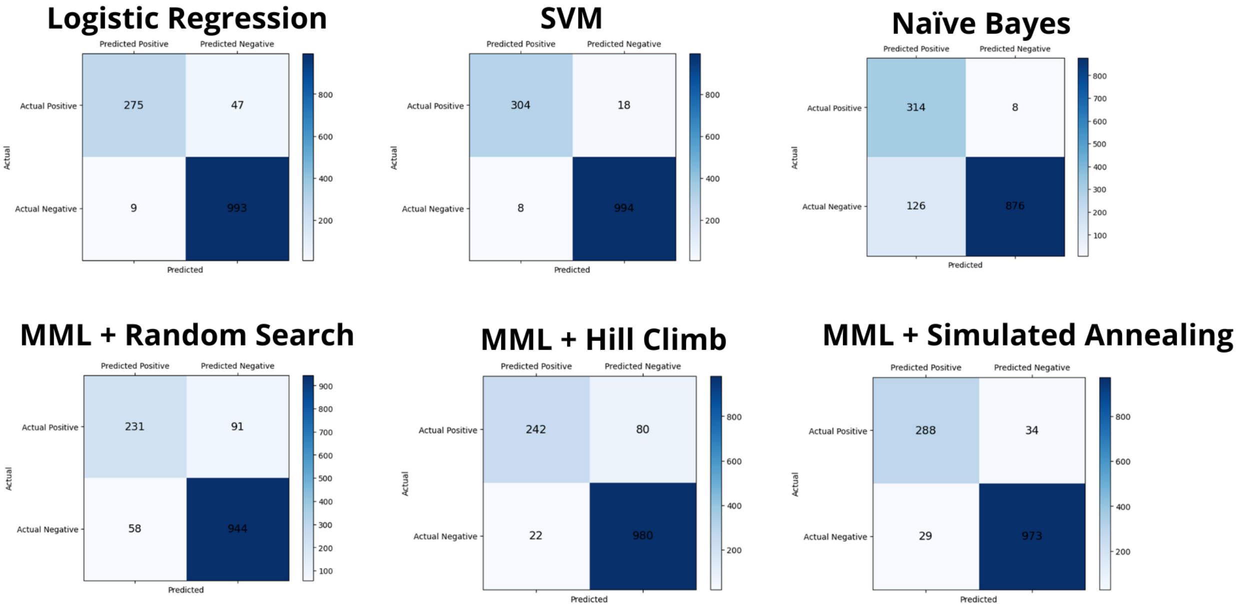


Fig. 7. Binary confusion matrix

6 Results and Discussion

This section reports the performance of baseline classifiers and the proposed MML-based variants under the LOOCV validation method, using spam as the positive class. To assess the effectiveness of the proposed approach, several baseline and optimized configurations were evaluated using standard performance metrics.

Since the dataset is moderately imbalanced (IR = 3.11), the analysis focuses on Recall, Specificity, Precision, F1-score, Balanced Accuracy, and MCC rather than overall accuracy. Figure 7 shows all the confusion matrices (baseline and proposed models).

6.1 Baseline Models

Tables 1, 2 and 3 summarizes the results for Logistic Regression (LR), linear SVM, and Multinomial Naïve Bayes (NB). SVM exhibits consistently strong behavior across all metrics, with the highest Balanced Accuracy (0.9680), F1-score (0.9589), and MCC (0.9589). NB attains the best Recall (0.9751) but at the expense of Precision (0.7136) and Specificity (0.8742), indicating a tendency to overflag ham as spam. LR offers a balanced trade-off, yet remains below SVM in every metric.

Table 1. Performance of logistic regression model under loocv

metric	logistic regression
Recall	0.8540
Specificity	0.9910
Balanced Accuracy	0.9225
Precision	0.9683
F1-score	0.9075
MCC	0.8832

Table 2. Performance of svm model under loocv

metric	SVM
Recall	0.9440
Specificity	0.9920
Balanced Accuracy	0.9680
Precision	0.9743
F1-score	0.9589
MCC	0.9589

Table 3. Performance of naïve bayes models under loocv

metric	Naïve Bayes
Recall	0.9751
Specificity	0.8742
Balanced Accuracy	0.9247
Precision	0.7136
F1-score	0.8241
MCC	0.7736

Table 4. Performance of proposed mml and random search variant under loocv

metric	MML + Random Search
Recall	0.7173
Specificity	0.9421
Balanced Accuracy	0.8297
Precision	0.7993
F1-score	0.7561
MCC	0.6849

6.2 Proposed MML Variants

All three proposed MML variants were executed using a reduced feature space consisting of the ten most discriminative attributes selected from the combined BM25–Word2Vec representation. This feature subset was determined to provide an optimal balance between dimensionality reduction and classification accuracy. For the *Hill Climb* and *Simulated Annealing* implementations, the optimization process was configured to perform 30 random restarts, ensuring adequate exploration of the search space and minimizing sensitivity to initial conditions.

Table 5. Performance of proposed mml and hill climb variant under loocv

metric	MML + Hill Climb
Recall	0.7515
Specificity	0.9780
Balanced Accuracy	0.8647
Precision	0.9166
F1-score	0.8259
MCC	0.7834

Table 4, 5, and 6 present the results for the three MML variants. As expected, *MML + Random Search* provides a stochastic baseline with modest performance (Balanced Accuracy = 0.8297, MCC = 0.6849). *MML + Hill Climb* improves all metrics through iterative local refinement (Balanced Accuracy = 0.8647, MCC = 0.7834). The best-performing variant is *MML + Simulated Annealing*, which achieves the highest scores among the proposed models across every metric, with Balanced Accuracy = 0.9327, F1-score = 0.9014, and MCC = 0.8700. Notably, the Simulated Annealing configuration attains high Recall (0.8944) *and* Specificity (0.9711), indicating a well-balanced control of both false negatives and false positives while maintaining computational efficiency.

6.3 Comparative Analysis

Compared to SVM (the strongest baseline) *MML + Simulated Annealing* narrows the gap substantially in Balanced Accuracy (0.9327 vs. 0.9680) and MCC (0.8700 vs. 0.9589), while delivering competitive Precision (0.9085) and a high F1-score (0.9014). Although SA does not surpass SVM overall, it offers an excellent trade-off between performance and model transparency, in line with the minimalist-explainable design goals.

The improvement from Random Search to Hill Climb and finally to Simulated Annealing

Table 6. Performance of proposed mml and simulated annealing variant under loocv

metric	MML + Simulated Annealing
Recall	0.8944
Specificity	0.9711
Balanced Accuracy	0.9327
Precision	0.9085
F1-score	0.9014
MCC	0.8700

Table 7. Wilcoxon paired test comparing mml variants and svm based on f1-score and mcc

Model comparison	Statistic	p-value
SVM vs. MML + RS	0.0000	0.5000
SVM vs. MML + HC	0.0000	0.5000
SVM vs. MML + SA	0.0000	0.5000

confirms that guided exploration with probabilistic acceptance is effective for feature-subset optimization under MML. These results support the claim that lightweight, interpretable models can approach the effectiveness of strong linear baselines on imbalanced, short-text spam detection tasks.

6.4 Statistical Analysis

To assess whether the observed performance differences between the proposed MML variants and the strongest baseline (SVM) were statistically significant, a paired Wilcoxon signed-rank test was conducted using the F1-score and MCC values obtained under the LOOCV scheme. This non-parametric test was selected due to its robustness and lack of assumptions regarding the normality of performance metric distributions.

Table 7 summarizes the results of the statistical comparison between each MML variant and the SVM baseline. In all cases, the Wilcoxon test yielded p-values greater than 0.05, indicating that there were no statistically significant differences in performance between the models.

Although no statistically significant differences were detected, the results are noteworthy: all proposed MML variants achieved F1 and MCC scores comparable to those of SVM, despite relying on a much simpler and fully interpretable architecture. This outcome supports the premise that minimalist and metaheuristic-driven models can reach competitive effectiveness in spam detection without the opacity and computational complexity inherent to conventional machine learning methods.

These findings demonstrate that metaheuristic-tuned minimalist models can achieve competitive performance while maintaining interpretability, a crucial balance in real-world spam filtering systems.

7 Conclusion and Future Work

This study presented a family of explainable and resource-efficient models for spam detection based on the Minimalist Machine Learning (MML) paradigm integrated with metaheuristic optimization techniques. Three variants (*MML + Random Search*, *MML + Hill Climb*, and *MML + Simulated Annealing*)were developed and evaluated on the SMS Spam Corpus v.0.1, using a hybrid BM25–Word2Vec feature representation. The results demonstrate that metaheuristic optimization significantly enhances the discriminative capacity of MML while maintaining its inherent simplicity and interpretability.

Among the proposed approaches, *MML + Simulated Annealing* achieved the best overall performance, reaching a Balanced Accuracy of 0.9327, F1-score of 0.9014, and MCC of 0.8700. Despite its minimal computational footprint, this configuration yielded results statistically comparable to the state-of-the-art SVM baseline under the Wilcoxon paired test. These findings confirm that MML, when coupled with adaptive search strategies, can serve as a viable and transparent alternative to traditional black-box classifiers in spam filtering and other short-text classification tasks.

Beyond spam detection, the principles of Minimalist Machine Learning hold promise for broader applications. Future research will explore extending the proposed MML variants to other text classification domains, such as sentiment analysis and the detection of AI-generated text. Additionally, an important avenue for future work involves the development of a hybrid architecture that combines the interpretability of MML with the representational power of transformer-based models. Such integration may yield models that are both semantically rich and fully explainable, bridging the gap between transparency and deep contextual understanding.

This research advances the goal of sustainable and interpretable AI by demonstrating that minimalist, metaheuristic-driven approaches can compete with complex models without compromising transparency. This work demonstrates that simplicity and explainability need not be sacrificed for performance. By leveraging metaheuristic optimization within a minimalist learning framework, it is possible to build models that are accurate, efficient, and transparent—an increasingly vital combination in modern artificial intelligence.

References

1. **Almufti, S. M., Shaban, A. A., Ali, R. I., Dela Fuente, J. A. (2023).** Overview of metaheuristic algorithms. Polaris Global Journal of Scholarly Research and Trends, Vol. 2, No. 2, pp. 10–32.

2. **Aprilianto, D. (2020).** SVM optimization with correlation feature selection based binary particle swarm optimization for diagnosis of chronic kidney disease. Journal of Soft Computing Experiments, Vol. 1, No. 1.

3. **Balfagih, A. M., Keselj, V., Taylor, S. (2022).** N-gram and word2vec feature engineering approaches for spam recognition on some influential twitter topics in saudi arabia. Journal of Advances in Information Technology.

4. **Budiman, D., et al. (2021).** Email spam detection: A comparison of SVM and naïve Bayes. Journal of Software Research and Engineering, Vol. 2, No. 1.

5. **Cepero-Pérez, N., Moreno-Espino, M., García-Borroto, M., Morales, E. F. (2023).** Progressive forest: Un criterio de detención temprana para la construcción de conjuntos. Computación y Sistemas, Vol. 27, No. 1, pp. 89–97.

6. **Cormack, G. V., Gómez Hidalgo, J. M., Puertas Sánchez, E. (2007).** Spam filtering for short messages. Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management (CIKM '07), ACM, Lisbon, Portugal, pp. 313–320.

7. **Dale, S. (2015).** Heuristics and biases: The science of decision making. Business Information Review, Vol. 32, No. 2, pp. 93–99.

8. **Galvis-Chacón, J., Ramos-Soto, Ó., Oliva, D., Valdivia, A., Rostro-González, H., Zapotecas-Martínez, S., Pérez-Cisneros, M. (2025).** Optimización del filtrado de electrocardiogramas para la detección mejorada de enfermedades cardiovasculares: un enfoque metaheurístico. Computación y Sistemas, Vol. 29, No. 1, pp. 77–89.

9. **García-Morales, M. A., Brambila-Hernández, J. A., Fraire-Huacuja, H. J., Frausto-Solís, J., Cruz-Reyes, L., Gómez-Santillán, C. G., Carpio-Valadez, J. M. (2024).** Algoritmo evolutivo multiobjetivo basado en descomposición con ajuste adaptativo de parámetros de control para resolver el problema biobjetivo de optimización de compras por internet (moea/d-aacpbishop). Computación y Sistemas, Vol. 28, No. 2, pp. 727–738.

10. **How, Y., Kan, M.-Y. (2005).** Optimizing Predictive Text Entry for Short Message Service on Mobile Phones. Lawrence Erlbaum Associates, Las Vegas, USA.

11. **Mallampati, D., P. Hegde, N. (2020).** Feature extraction and classification of email spam detection using IMTF-IDF. Journal of Computer Applications.

12. **Methling, F., Abdeen, S., von Nitzsch, R. (2022).** Heuristics in multi-criteria decision-making: The cost of fast and frugal decisions. EURO Journal on Decision Processes, Vol. 10, pp. 100013.

13. **Nadhifa, A., et al. (2021).** Comparison of KNN, naïve Bayes and decision tree methods for email classification. Journal of Software Research and Engineering, Vol. 1, No. 2.

14. **Nandhini, S., Marseline, K. (2020).** Performance evaluation of machine learning algorithms for

email spam detection. International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), IEEE, pp. 1–4.

15. **Rodríguez-González, A. Y., Pérez-Espinosa, H., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., Olvera-López, J. A. (2024).** Sección temática: Avances en reconocimiento de patrones. Computación y Sistemas, Vol. 28, No. 1, pp. 197–198.

16. **Sahmoud, T., Mikki, M. (2021).** Spam detection using BERT. International Journal of Computer Applications.

17. **Solorio-Ramírez, J., Saldana-Perez, M., Lytras, M., Moreno-Ibarra, M., Yáñez-Márquez, C. (2021).** Brain hemorrhage classification in CT scan images using minimalist machine learning. Diagnostics, Vol. 11, No. 8, pp. 1449.

18. **Yáñez-Márquez, C. (2020).** Toward the bleaching of the black boxes: Minimalist machine learning. IEEE IT Professional, Vol. 22, No. 4, pp. 51–56.

19. **Zaman-Khan, H., Naeem, M., Guarasci, R., Bint-Khalid, U., Esposito, M., Gargiulo, F. (2024).** Mejorando la clasificación de texto mediante BERT: un enfoque de aprendizaje por transferencia. Computación y Sistemas, Vol. 28, No. 4, pp. 2279–2296.

Article received on 21/05/2025; accepted on 22/08/2025.
**Corresponding author is Cornelio Yáñez-Márquez.*