Ratio Mathematica Volume 56, 2026

Spam Email Detection Using Advanced Machine Learning Techniques

Salvador Cruz Rambaud[*]
Giacomo di Tollo[†]
Arinze Moses Emeti[‡]
Ludovico Mascia[§]
Massimo Squillante[**]

Abstract

The increasing sophistication of unsolicited and malicious email creates a need for spam-detection methods able to identify context-dependent and linguistically complex patterns that may evade conventional filters. This study compares two approaches to spam classification: Bidirectional Encoder Representations from Transformers (BERT) and Self-Organizing Maps (SOM). Generalization is assessed on five publicly available email benchmarks with different sizes and class distributions. For each dataset, thirty independent train-test partitions are generated; preprocessing, class balancing, model fitting, and REVAC hyperparameter tuning are performed using training data only. Performance on previously unseen test partitions is evaluated using precision, recall, accuracy, and F1-score. Both approaches provide effective classification, but BERT consistently outperforms SOM across all five datasets. The results support the value of contextual, sequence-aware representations for sophisticated spam patterns and underline the importance of rigorous, leakage-free evaluation when comparing spam-detection models.

Keywords: Natural Language Processing (NLP), Transformer Model, Binary Classification, BERT (Bidirectional Encoder Representations from Transformers), SOM (Self-Organizing Maps).

2010 AMS subject classification: 49 Calculus of variations and optimal control; optimization; 65 Numerical analysis; 90 Operations research, mathematical programming[††]

[*] University of Almeria, Department of Economics and Business, Almeria, Spain.
[†] Università Politecnica delle Marche, Department of Management, Ancona, Italy; g.ditollo@univpm.it (corresponding author).
[‡] University of Luxembourg, Department of Computer Science, Esch-sur-Alzette, Luxembourg.
[§] Università Telematica "Giustino Fortunato", Benevento, Italy.
[**] Accademia Peloritana dei Pericolanti, Messina, Italy.
[††] Received on April 24, 2026. Accepted on September 18, 2026. Published on September 19, 2026. DOI: 10.23755/rm.v56i0.1749. ISSN: 1592-7415. eISSN: 2282-8214.

1. Introduction

Email is one of the most widely used forms of digital communication because of its accessibility, low cost, and ability to support the rapid exchange of information. For this reason, it represents a major vector for unwanted and potentially malicious content, including unsolicited commercial messages, phishing attempts, fraudulent communications, and malware distribution. Therefore, *spam* represents both a nuisance to users/organizations and a relevant cybersecurity challenge. In particular, *phishing* and other forms of deceptive email can exploit users' behavior and linguistic expectations to obtain sensitive information, facilitate unauthorized access, or distribute malicious content. The scale and diversity of these threats make the reliable automatic identification of unwanted messages an important problem in both information security and artificial intelligence.

Spam detection can be formulated as a binary classification problem in which incoming messages are assigned to one of two classes: legitimate email (*ham*) or *spam*. The effectiveness of a spam detection system depends on its ability to identify linguistic, structural, and semantic characteristics that distinguish the two afore mentioned classes. Early approaches to email filtering relied extensively on manually defined rules, keyword matching, and statistical representations of message content: although such approaches can be effective for relatively simple forms of spam, their performance may deteriorate when messages employ obfuscated terminology, synonyms, deliberately altered spelling, deceptive hyperlinks, or context-dependent language; moreover, the increasing sophistication of spam campaigns makes it difficult to define a static set of rules capable of adapting to the continuously changing characteristics of unwanted email.

The afore mentioned limitations have motivated the adoption of machine-learning and natural language processing (NLP) techniques for spam classification: rather than relying exclusively on predefined rules, machine-learning approaches can learn discriminative patterns directly from collections of labelled or unlabelled email messages. Conventional machine-learning methods, however, often depend on manually engineered textual representations, such as term-frequency or TF-IDF features, which may provide only limited information about the contextual relationships among words. More recent developments in NLP, particularly transformer-based architectures, have substantially expanded the ability of machine-learning models to represent natural language.

Among these developments, Bidirectional Encoder Representations from Transformers (BERT) has become an important architecture for contextual language representation: BERT uses transformer-based self-attention mechanisms to construct representations of tokens by considering their surrounding context in both directions [4]. In this framework, the representation assigned to a word depends not only on the word itself but also on the linguistic context in which it occurs. This property is particularly relevant to spam detection because the distinction between legitimate and unwanted messages may depend on subtle combinations of words, phrases, semantic relationships, and contextual cues that cannot be adequately captured by keyword-based methods. Previous studies have demonstrated the potential of BERT and related deep-learning architectures for email spam detection [2,5,8], suggesting that contextual language representations can provide substantial improvements over traditional approaches.

At the same time, neural approaches can provide complementary perspectives on the organization of textual data. Self-Organizing Maps (SOMs) are unsupervised neural-network models that project high-dimensional observations onto a lower-dimensional topological map while preserving, to some extent, relationships among observations in the original feature space

[13,14]. In a spam-detection setting, SOMs can be used to organize email representations according to their similarity and subsequently associate regions of the resulting map with spam or ham through a post-training classification procedure. Unlike BERT, SOMs do not explicitly model sequential linguistic context or contextual dependencies between tokens. Nevertheless, their ability to represent the structure of high-dimensional data and provide an interpretable visual organization makes them an interesting benchmark against which contextual transformer-based models can be compared.

The comparison between BERT and SOM is therefore relevant because the two approaches embody substantially different principles of representation and learning. BERT explicitly exploits contextual and sequential information through self-attention and supervised fine-tuning, whereas SOM organizes observations according to distances within a predefined feature space through unsupervised learning. Comparing these paradigms on the same spam-classification problem provides an opportunity to assess whether the additional contextual information captured by transformer-based representations translates into more consistent predictive performance across heterogeneous email benchmarks.

A further issue concerns the reliability of performance estimates in empirical spam-classification studies. Reported classification results can be strongly influenced by the choice of training and test partitions, class imbalance, pre-processing procedures, and hyperparameter configurations. In particular, evaluating a model on data that have also data pre-processing, model selection, or parameter optimization can result in information leakage and consequently lead to overly optimistic estimates of predictive performance. A robust evaluation procedure should therefore ensure that all data-dependent operations are performed exclusively on the training portion of each experiment, while the corresponding test observations remain completely unseen until final evaluation.

In this study, we address these issues through a systematic comparison of BERT and SOM for spam email classification: five publicly available email benchmarks are considered, providing heterogeneous sets of data in terms of corpus size and *spam*-to-*ham* ratios. To assess the robustness of the results, the experiments are repeated over thirty independent train-test partitions for each set of data. Within each partition, pre-processing, class balancing, model selection, and parameter estimation are performed using only the training data, while the final performance is measured exclusively on the corresponding held-out test set. Class imbalance is addressed through the Synthetic Minority Over-sampling Technique (SMOTE) [23], applied only to the training partition, thereby preserving the original composition of the test data.

An additional methodological contribution is the use of Relevance Estimation and VAlue Calibration (REVAC) [26] for hyperparameter optimization. REVAC provides a systematic parameter-tuning framework that can be applied to different classes of algorithms. In the present study, it is used to optimize the relevant hyperparameters of both BERT and SOM, allowing the two approaches to be evaluated under a structured and comparable optimization procedure rather than relying exclusively on manually selected parameter values.

The main objective of the study is therefore to determine whether the contextual representations provided by BERT offer a consistent advantage over the topological, distance-based representation of SOM in *spam* email classification, while simultaneously examining the robustness of this comparison across multiple sets of data and train-test partitions. Model performance is evaluated using precision, recall, accuracy, and F1-score, and paired statistical comparisons are performed across the common train-test partitions to determine whether observed differences between the two approaches are statistically significant.

The contribution of this work is threefold. First, it provides an empirical comparison between a contextual transformer-based classifier and an unsupervised topological neural model within the same spam-detection framework. Second, it adopts a repeated, leakage-aware experimental protocol across five heterogeneous public sets of data, providing a more robust assessment of generalization than evaluations based on a single train-test partition. Third, it investigates the use of REVAC as a common hyperparameter optimization framework for substantially different learning architectures. The resulting analysis provides evidence concerning both the predictive advantages of contextual language modeling and the applicability of systematic parameter optimization to spam-classification problems.

The remainder of the paper is organized as follows. Section 2 introduces the BERT, SOM, and REVAC methodologies employed in the study. Section 3 describes the sets of data and pre-processing procedures, including the experimental pipeline used to prevent information leakage. Section 4 presents and discusses the empirical results and statistical comparisons. Finally, Section 5 summarizes the main findings, discusses their implications, and outlines directions for future research.

2. Methods

In what follows we are introducing the main components of our experimental approach: the Bidirectional Encoder Representations from Transformers (BERT) will be introduced in Section 2.1, Self-Organizing Maps (SOM) will be introduced in Section 2.2, and Relevance Estimation and VAlue Calibration (REVAC) will be outlined in Section 2.3.

2.1 BERT

BERT is a natural language processing (NLP) model developed by Google, and it is structured to learn deep bidirectional representations from unlabeled text by simultaneously considering the context on both sides of a token across all layers. This allows the pre-trained model to be fine-tuned with only a minimal output layer, enabling the user to achieve high performance on various tasks—such as question answering and natural language inference—without the need for extensive changes to the model architecture.

BERT is based on the transformer architecture, which employs attention mechanisms to weigh the importance of each word to others in the sentence [4]: this allows BERT to develop a comprehensive understanding of the sentence structure and meaning, which is essential for spam detection, as spam emails often use varied and sophisticated language to avoid detection. This makes BERT exceptionally effective for a variety of NLP tasks, distinguishing it from traditional tools such as VADER, TextBlob, and Naive Bayes classifiers, which often rely on lexicons or rule-based approaches [6]. In spam detection, BERT is typically fine-tuned on a large, labelled sets of data of *spam* and *ham* emails: the fine-tuning process adjusts the pre-trained model's weights based on spam-specific vocabulary, structures, and linguistic cues [4]. By training BERT on *spam* data, the model learns to identify patterns specific to spam emails, such as using certain phrases, unnatural language structures, or links to external sites. Once fine-tuned, BERT can classify incoming emails as *spam* or *ham* with high accuracy.

BERT includes several model variants (e.g., BERT-base, BERT-large), and their choice and fine-tuning have an impact on system performance [5]. In our approach, the BERT model is

initially pre-trained on vast text corpus using unsupervised learning, then we involved additional training on domain-specific labelled data for spam detection (see the training-test partitions introduced in Section 4). We also point out that designing an effective fine-tuning strategy requires setting hyperparameters such as learning rate, batch size, and number of training epochs while applying techniques like early stopping and learning rate scheduling to prevent overfitting and optimise the model's performance. In our approach, we have resorted to REVAC[12] to set the hyperparameters of the model, that will be outlined in Section 2.3.

For completeness and reproducibility, we detail below the mathematical formulation of the components of the BERT pipeline actually used in this contribution [5,8].

Let an input email, after tokenization, be represented as a sequence $x = (x_1, \ldots, x_n)$, obtained via the model's WordPiece tokenizer together with the special tokens [CLS] and [SEP]. Each token x_i is mapped to an embedding vector $e_i = E\_tok(x_i) + E\_pos(i) + E\_seg(x_i) \in \mathbb{R}^d$, combining token, positional, and segment embeddings, where d is the hidden dimension of the chosen BERT variant.

The embedded sequence is processed by L stacked transformer encoder layers. Each layer applies multi-head self-attention. For head h, queries, keys and values are computed as $Q\_h = EW^H\_h$, $K\_h = EW^K\_h$, $V\_h = EW^v\_h$, and scaled dot-product attention is defined as:

$$Attention(Q\_h, K\_h, V\_h) = softmax(Q\_h K\_h^T / \sqrt{d\_k}) V\_h$$

where d_k is the per-head key dimension. The outputs of the H attention heads are concatenated and linearly projected, $MultiHead(E) = Concat(head_1, \ldots, head\_H)W^O$, followed by a position-wise feed-forward network, residual connections and layer normalization, repeated across the L layers.

For classification, the final hidden state of the [CLS] token, $h\_[CLS] \in \mathbb{R}^d$, is passed through a linear classification head with a sigmoid activation:

$$\hat{y} = \sigma(W\_c\, h\_[CLS] + b\_c)$$

producing the predicted probability that an email is spam. The model is fine-tuned by minimizing the binary cross-entropy loss:

$$L = -(1/N) \Sigma_i [y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i)]$$

computed over the N training examples of each split.

2.2 SOM

Self-Organizing Maps (SOMs) [13] are neural network models used to transform high-dimensional data into a two-dimensional layout, making it easier to explore patterns and relationships, and they have been used in different classification exercises [13,14]. In the present study, the SOM consists of a single layer of neurons connected by synapses and arranged on a two-dimensional grid, which is trained using unlabelled input data. The resulting map provides a visual representation of the nonlinear relationships present within the multidimensional input space. The algorithm iteratively updates synapses' weights by identifying the neuron closest to each input (based on Euclidean distance) and adjusting it and its neighbors using a neighborhood function: at the beginning of the training process, each neuron is assigned a randomly initialised weight vector whose dimensionality corresponds to that of the input data. The network is then trained iteratively by presenting the input vectors to the map. For each input, the best matching unit (BMU) is identified as the neuron whose weight vector has the smallest Euclidean distance from the input vector.

Once trained, SOM forms a map that reveals the nonlinear structure of the data. In this case, the resulting map is analyzed post-training to detect clusters and examine how the two classes of email (Spam and Ham) are distributed across the map, although these scenarios were not used during the learning phase.

In our approach we use an incremental approach that introduces an initial feature vector model from the first email batch, and trains online a SOM with randomized on this batch. Later email batches are then incorporated, the feature vector model is updated, and the SOM is tested using the existing weights, but only on the new batch data. This cycle repeats iteratively until all batches are processed. Hyperparameters (learning rate, neighborhood size) have been tuned by REVAC; to mitigate initial weight bias, each training phase was replicated across multiple runs, with results averaged (see Section 4). For completeness and reproducibility, we formalize below the SOM algorithm and the rule used to derive spam/ham predictions from the trained map. Let each preprocessed email be represented as a feature vector $v \in \mathbb{R}^m$. The SOM consists of a two-dimensional grid of neurons, each neuron j associated with a weight vector $w_j \in \mathbb{R}^m$, initialized at random. For an input v, the best matching unit (BMU) is the neuron minimizing Euclidean distance:

$$c(v) = \text{argmin}_j \|v - w_j\|$$

During training, the BMU and its neighboring neurons are updated as:

$$w_j(t+1) = w_j(t) + \alpha(t) \cdot h_{j,c(v)}(t) \cdot (v - w_j(t))$$

where $\alpha(t)$ is the learning rate, decaying as $\alpha(t) = \alpha_0 \exp(-t/\tau)$, and h is a Gaussian neighborhood function $h_{j,c(v)}(t) = \exp(-\|r_j - r_{c(v)}\|^2 / 2\sigma(t)^2)$, with r_j, $r_{c(v)}$ the grid coordinates of neuron j and the BMU, and $\sigma(t)$ the neighborhood radius, also decaying over training.

Once training has been completed, the resulting neuron map is examined to identify groups of neighbouring neurons characterised by relatively small pairwise distances, which may indicate the presence of clusters within the data. In the present application, information concerning the potential mail type (i.e., ham or spam) is represented separately by a vector defining two categories: a majority-vote procedure is applied as a post-training labelling procedure, in which each each training observation is mapped to its BMU: for every training observation x_i, identify the best matching unit (BMU), i.e. the neuron whose weight vector w_j has the smallest Euclidean distance, is identified; then, observations assigned to each neuron are collected, and the neuron is assigned to the majority class (ties are broken via the nearest-distance rule).

It should be emphasised that the scenario labels were not supplied to the SOM during the training phase. Instead, they were associated with the observations only after the final map had been obtained. Consequently, the SOM performed the organisation of the input data independently of the predefined mail scenarios, which were subsequently used to interpret and characterise the structure emerging from the trained map.

2.3 REVAC

REVAC (Relevance Estimation and Value Calibration [26]) is a *parameter tuning* algorithm based on the Estimation of Distribution (EDA), that exploits information-theoretic measures to assess the relevance of individual parameters during the optimization process. More specifically, REVAC maintains a probability distribution over the parameter space, defined as the Cartesian

product of the domains of the parameters to be optimized. The distribution is progressively adapted so as to assign greater probability to parameter values associated with favorable trade-offs between algorithmic performance and complexity. Within REVAC, complexity is quantified in terms of Shannon entropy, thereby providing an information-theoretic measure of the uncertainty associated with the parameter distributions.

REVAC operates iteratively, beginning with a uniform probability distribution over the parameter space, which is subsequently refined through evolutionary computation techniques through a process referred to as *smoothing,* maintaining a population of parameter vectors and generates new candidate configurations by preferentially selecting high-performing individuals [27]. Specifically, a subset of the current population is selected according to the expected performance of the corresponding parameter configurations, and the resulting configurations are used to replace older individuals in the population.

The smoothing mechanism is implemented through an operator that defines a mutation interval for each parameter. At each iteration, the parameter values represented in the current population are sorted, and a predefined number of extreme values are removed to obtain a refined empirical distribution. This distribution is subsequently used to generate the parameter values of the next population through random sampling. As the optimization progresses, the Shannon entropy of the parameter distributions is expected to decrease, reflecting a reduction in uncertainty regarding promising regions of the parameter space. The resulting changes in entropy can therefore also be used to assess parameter relevance. In particular, parameters exhibiting a substantial reduction in entropy are interpreted as being more sensitive to their assigned values and, consequently, as potentially more influential in determining the performance of the underlying algorithm.

This contribution uses REVAC to tune the hyperparameters of both BERT and SOM. Formally, let $\theta = (\theta_1,\ldots,\theta\_p) \in \Theta$ denote a vector of hyperparameters (e.g. learning rate, batch size, neighborhood size) with domain $\Theta = \Pi_k \Theta\_k$, and let $f(\theta)$ denote the validation cost function to be optimized (in our case, a function of precision, recall, and accuracy on a held-out validation partition). REVAC maintains a population of candidate configurations $\{\theta_1,\ldots,\theta\_M\}$ together with an estimated relevance-weighted density $p(\theta)$ over Θ, and iteratively: (i) samples new candidate configurations from $p(\theta)$; (ii) evaluates $f(\theta)$ for each candidate on the validation partition; (iii) updates $p(\theta)$ by giving higher density to regions of Θ associated with lower cost, using a kernel-density-style update over the current population; and (iv) repeats until a stopping criterion is met (e.g. a maximum number of generations or negligible improvement in f).

3. Material

This section details data at hand used for our experiments: Section 3.1 describes the five sets of data used; Section 3.2 outlines the pre-processing operations necessary for dealing with these sets of data. The full pipeline of our experimental analysis is reported in Section 3.3

3.1 Sets of data

For our experiments we have used five publicly available sets of data, that are described as follows:

- **Spambase** set of data (referred to as A in what follows), introduced by [8] and publicly available at https://archive.ics.uci.edu/dataset/94/spambase. This set of data contains 1813 *spam* messages and 2788 *ham* messages: spam represents the 39% of the corpus;
- **Spamassassin** set of data (referred to as B), introduced by [16] and publicly available at https://www.kaggle.com/datasets/ganiyuolalekan/spam-assassin-email-classification-dataset This set of data contains 1897 *spam* messages and 4150 *ham* messages: spam represents the 31% of the corpus;
- **GenSpam** set of data (referred to as C in what follows), introduced by [17] and publicly available at http://www.benmedlock.co.uk/genspam.html. This set of data contains 31196 *spam* messages and 9212 *ham* messages: spam represents the 78% of the corpus;
- **CSDMC2010** set of data (referred to as D in what follows), introduced by [18] and publicly available at https://github.com/zrz1996/Spam-Email-Classifier-DataSet. This set of data contains 1378 *spam* messages and 2929 *ham* messages: spam represents the 32% of the corpus;
- **PU1** set of data (referred to as E in what follows), introduced by [19] and publicly available at http://www.csmining.org/index.php/pu1-and-pu123a-s.html This set of data contains 481 *spam* messages and 618 *ham* messages: spam represents the 44% of the corpus.

3.2 Data Pre-Processing

Spam email data typically includes noise, misspellings, and non-standard language. Thus, pre-processing is vital to clean and normalise the text before inputting it into the BERT model.

First, the data pre-processing stage involves cleaning and standardising the raw email text: this is done through a comprehensive linguistic pre-processing pipeline composed of tokenisation and lemmatisation (performed using the Python library spaCy, as it is referred to in Section 3.3), used to convert the unstructured email content into a standardized sequence of linguistic units [21]. The resulting textual representations are subsequently used to construct custom word embeddings that capture semantic and contextual relationships among words and enable the model to better represent domain-specific language , to ensure that the raw email data can be transformed efficiently and consistently into structured representations suitable for subsequent spam-detection modelling.

Then, feature engineering [22] is crucial in spam detection. In traditional models, this involves techniques like Term Frequency-Inverse Document Frequency (TF-IDF) to weigh words based on their importance within the corpus or word embeddings that represent words as vectors in a continuous space [5]. With BERT, however, feature engineering is inherently part of the model's process through its use of embeddings and attention mechanisms: BERT generates embeddings that capture both the semantic meaning and contextual information of each word in the email [5]. These embeddings allow the model to detect nuanced language patterns that are often characteristic of spam emails, such as ambiguous phrasing or unconventional language designed to bypass keyword filters. By generating context-aware embeddings, BERT can identify even subtle indicators of spam, such as the misuse of punctuation, excessive use of certain keywords, or the presence of hyperlinks masked with deceptive text.

Last, we had to consider that spam sets of data often show class imbalances, with spam messages typically outweighing *ham* ones. We have resorted to SMOTE (Synthetic Minority Over-sampling Technique) to create a balanced model [23]: for a minority-class (spam or ham,

whichever is underrepresented in a given split) sample x_i with k nearest minority-class neighbors, SMOTE generates a synthetic sample as:

$x\_new = x_i + \lambda \cdot (x\_zi - x_i),\ \lambda \sim U(0,1)$

where x_zi is one of the k nearest minority-class neighbors of x_i (chosen at random) and λ is drawn uniformly from [0,1], so that x_new lies on the line segment joining x_i and x_zi in feature space. Critically, SMOTE is applied only within the training partition of each of the thirty splits: for every split, the training and test sets are first separated, and oversampling is performed exclusively on the resulting training set, after which the (unmodified, real) test set is used for evaluation.

The proposed pipeline incorporates deep-learning frameworks to support the extraction of linguistic information and the learning of semantic representations from the email corpus. A machine-learning framework is employed to implement and train the named-entity recognition component, which identifies and categorises entities such as persons, organisations, and locations within email content. This information provides additional contextual and semantic cues that can contribute to distinguishing legitimate messages from spam. A separate deep-learning framework is used to train the word-embedding model, transforming individual words into dense numerical representations that encode their semantic relationships based on their patterns of occurrence within the corpus. These embeddings enable semantically and contextually related words to be represented in similar regions of the learned vector space, thereby providing the subsequent classification model with a richer representation of email-specific language and contextual information [7].

All data-dependent pre-processing, model-selection, and model-training procedures are performed strictly within the training partition associated with each experimental split. Specifically, oversampling, feature normalization, vocabulary construction, embedding generation, hyperparameter optimization through REVAC, and subsequent model fitting are independently carried out using only the data available in the corresponding training partition. Any statistics, representations, or parameter values learned during these procedures are therefore derived exclusively from the training data and are subsequently applied to the corresponding held-out test partition without further adaptation. In particular, neither the feature distributions nor the class composition, vocabulary, embeddings, or validation performance observed during hyperparameter optimization are allowed to incorporate information from the test data. The test partition remains completely unseen throughout pre-processing, hyperparameter selection, and model fitting and is used exclusively for the final evaluation of the resulting model. This nested procedure prevents information leakage between the training and test sets and ensures that the reported performance provides an unbiased estimate of the model's ability to generalize to previously unseen data.

3.3 Experimental Pipeline Overview

The full experimental pipeline, applied independently within each of the thirty train-test partitions, proceeds as summarized in the pseudocode below.

```
for split s = 1 … 30:
    (train_s, test_s) ← split(data, seed=s)
```

```
    train_s ← preprocess(train_s)   # tokenization / spaCy pipeline /
cleaning
    train_s ← SMOTE(train_s)        # oversampling, TRAIN PARTITION
ONLY
    θ* ← REVAC(train_s, model ∈ {BERT, SOM})   # hyperparameter
tuning on a validation fold of train_s
    model_s ← fit(train_s, θ*)
    test_s ← preprocess(test_s)      # using parameters/vocabulary
learned on train_s only
    y_hat_s ← predict(model_s, test_s)
    metrics_s ← evaluate(y_hat_s, y_test_s)   # precision, recall,
accuracy
end for
report summary statistics (min, max, median, mean, std) of metrics_s
over s = 1…30
```

4. Results

This section presents the empirical results obtained from the BERT and Self-Organizing Map (SOM) models across the five email corpora. *Spam* is defined as the positive class and *ham* as the negative class. Please notice that, according to the guidelines introduced by [20], we have performed our experiments over thirty different partitions of data at hand, and in what follows we assess the fitness of our approach by showing main statistics of precision, recall, and accuracy over the thirty defined test sets. In addition, the F1-score was calculated as the harmonic mean of precision and recall (F1 = 2·Precision·Recall/(Precision+Recall)) and reported over a sample test set. The results obtained with BERT and SOM are reported in Tables 1 and 2, respectively. Table 3 summarizes the F1-score obtained by the two approaches for each set of data.

The results show that BERT outperforms SOM over all sets of data: BERT achieves a higher F1-score on all five sets of data, with values ranging from 0.940 for A to 0.997 for C. SOM also provides generally strong classification performance, with F1-scores ranging from 0.842 to 0.967, but remains below BERT on every instance. The difference between the two approaches is relatively small for instances A and B, where the F1-score gap is 0.006 and 0.003, respectively. The difference is more pronounced for instances D and C, where the corresponding gaps are 0.078 and 0.130. These results suggest that the advantage of BERT is not necessarily uniform across sets of data and may become more relevant when the linguistic and structural characteristics of the corpus create a greater need for contextual representations.

The particularly high F1-score achieved by BERT on instance C is noteworthy. With an F1-score of 0.997, the model provides an almost complete separation between the two classes under the experimental conditions adopted in this study. This result should nevertheless be interpreted in the context of the characteristics of the individual corpus and should not be generalized as an expected performance level for arbitrary future *spam* sets of data. On the other side, the substantially lower SOM performance on instance E indicates that the effectiveness of a distance-based representation can be more sensitive to the characteristics of the underlying feature space. Overall, the results are consistent with the broader literature reporting strong performance from machine-learning and deep-learning approaches for spam detection [24] that shows performances over the three aggregates (i.e., accuracy, precision, and recall) around 0.9. However, direct comparisons with previously reported results should be made cautiously

because studies differ considerably in sets of data, preprocessing, experimental protocols, class distributions, and evaluation procedures. Please remark anyhow that, unlike many works in the literature that evaluate performance on the entire set of data [24] —potentially inflating predictive metrics—, the performances of our models are assessed strictly on the test partition, ensuring that no data leakage occurs. This reflects a more realistic assessment of generalization and robustness and represents a good feature of our approaches, that are apt to generalise their behaviour without incurring in over-fitting.

The observed difference between the two models can be interpreted in terms of their fundamentally different representations of textual information.

SOM operates on a fixed-dimensional representation of each email and organizes observations according to their distances within the resulting feature space. Its learning mechanism is useful for identifying topological structures and groups of similar observations, but it does not explicitly model token order, long-range dependencies, or the context-dependent meaning of individual words. Consequently, emails that appear similar according to their aggregate feature representation may be positioned close to one another even when their linguistic meaning or classification category differs.

BERT, in contrast, constructs contextual representations using self-attention. The representation of each token is influenced by the other tokens in the sequence, allowing the model to account for relationships that depend on the surrounding linguistic context. This is particularly relevant to spam detection because malicious or unwanted messages may deliberately employ ambiguous wording, indirect requests, unusual combinations of terms, or linguistic structures intended to evade conventional filters.

The comparative results therefore provide empirical support for the hypothesis that contextual representations can offer an advantage over static distance-based representations in spam classification. Importantly, this advantage is observed across all five sets of data rather than being confined to a single corpus.

At the same time, the results do not imply that SOM is ineffective for text classification. Its F1-score exceeds 0.90 on four of the five sets of data and reaches 0.967 on instance B. SOM consequently provides a competitive alternative under several experimental conditions. Its principal limitation in the present comparison appears to concern the representation of complex linguistic information rather than an inability to separate spam and ham altogether.

This performance gap can be linked more precisely to how each model represents textual data. SOM operates on a fixed-dimensional feature vector per email and organizes emails purely by distance in that static feature space; it has no mechanism for modeling token order, long-range dependencies, or context-dependent word meaning, so two emails with similar surface-level features but different word order or semantic role can be placed close together on the map even when their spam/ham status differs. BERT, by contrast, is designed specifically to encode sequential and contextual structure: its self-attention mechanism lets each token's representation depend on the other tokens it co-occurs with, allowing the model to capture semantic ambiguity, negation, and context-dependent phrasing that are precisely the kinds of sophisticated language patterns spammers use to evade detection. This distinction — static, distance-based representation versus context-aware, sequence-based representation — is consistent with SOM's comparatively higher variance and lower ceiling in Table 2 relative to BERT's results in Table 1.

To assess whether the differences in predictive performance between BERT and SOM are statistically significant, paired comparisons were performed across the thirty train-test partitions. For each set of data and each evaluation metric, the performance obtained by BERT on partition s was paired with the performance obtained by SOM on the same partition, thereby controlling for variability induced by the particular train-test split. Let $m_{B,s}$ and $m_{S,s}$ denote the performance of BERT and SOM, respectively, on partition s, with $s = 1, \ldots, 30$. The paired difference was defined as $d_s = m_{B,s} - m_{S,s}$. Because the number of partitions is limited and the distribution of performance differences cannot be assumed a priori to be Gaussian, the Wilcoxon signed-rank test was employed to assess the null hypothesis that the median paired difference is zero. The analysis was conducted for the principal performance measures reported in the study, including accuracy, precision, recall, F1-score, specificity, and balanced accuracy. To account for the number of statistical comparisons, the resulting p-values were adjusted using the Holm correction for multiple testing. In addition to statistical significance, an appropriate effect-size measure was reported to quantify the magnitude and direction of the observed differences. All statistical comparisons were conducted using the predictions obtained exclusively on the test sets: this lead us to reject the null hypothesis H_0, which states that there is no statistically significant difference between the performance of BERT and SOM across the thirty paired train-test partitions.

5. Discussion and Conclusions

This study investigated the effectiveness of two substantially different artificial-intelligence approaches for spam email classification: Bidirectional Encoder Representations from Transformers (BERT) and Self-Organizing Maps (SOM). The central research question was whether the contextual, sequence-aware representations generated by BERT provide a systematic advantage over the distance-based topological representation produced by SOM. The empirical evidence obtained from five publicly available email corpora and thirty train-test partitions provides a consistent answer: both approaches are capable of effectively distinguishing spam from legitimate email, but BERT achieves superior predictive performance across all five sets of data.

The results are particularly clear when considering the F1-score, which jointly accounts for precision and recall. BERT achieves F1-scores between 0.940 and 0.997, while SOM ranges between 0.842 and 0.967. Although the difference is relatively modest for two instances, BERT shows substantially larger advantages on the remaining ones. This variation is relevant because it indicates that the benefit of contextual language modeling is not independent of the characteristics of the underlying corpus. In sets of data where spam messages exhibit more complex, variable, or context-dependent linguistic patterns, the ability of BERT to construct representations conditioned on surrounding text may become particularly valuable.

The comparison also provides insight into the different strengths of the two approaches. BERT is explicitly designed to capture contextual relationships in language through transformer-based self-attention. Its representations can therefore incorporate relationships between tokens that are difficult to express through a fixed-dimensional, distance-based representation. This characteristic is well suited to spam detection, where seemingly innocuous words can become indicative of spam depending on their combination and context.

SOM follows a fundamentally different learning principle. Rather than learning a contextual representation of language, it organizes observations according to their similarity within a feature

space and provides a topological representation of that space. This property makes SOM attractive for exploratory analysis and visualization and may provide an additional interpretability advantage. Nevertheless, the present results indicate that its classification performance is more limited when the distinction between spam and ham depends strongly on contextual or sequential information.

The results should therefore not be interpreted as suggesting that more complex deep-learning architectures are universally preferable. Rather, they provide evidence that the representation of textual information is a critical determinant of spam-classification performance. BERT's advantage in the present experiments is plausibly related to its ability to represent contextual semantics, whereas SOM's performance is constrained by the information encoded in its input feature representation and its distance-based organization mechanism.

A second contribution of the study concerns the experimental methodology. Rather than relying on a single train-test partition, the analysis was repeated over thirty partitions for each set of data. This provides a more informative assessment of model robustness and reduces the possibility that the conclusions are driven by a particular division of the observations. Moreover, all data-dependent operations were restricted to the corresponding training partitions. In particular, SMOTE was applied only to training observations, while preprocessing, representation learning, hyperparameter optimization, and model fitting were performed without access to the final test observations. This leakage-aware design is important when evaluating text-classification systems because inadvertent use of test information can substantially inflate reported predictive performance.

The use of REVAC represents an additional methodological contribution. Hyperparameter selection can have a substantial influence on the performance of machine-learning models, particularly when comparing architectures with fundamentally different learning mechanisms. Applying the same systematic parameter-optimization framework to BERT and SOM reduces reliance on manually chosen parameter configurations and provides a more structured basis for comparison. More generally, the results suggest that REVAC can be usefully employed as a model-independent optimization mechanism across different classes of learning algorithms.

The study also has several limitations. First, although five publicly available sets of data were considered, all corpora are established benchmark sets of data and may not fully represent the current evolution of spam, phishing, and adversarial email campaigns. Spam evolves continuously, and contemporary attacks may exhibit linguistic, multilingual, and structural characteristics that differ from those represented in historical corpora. Consequently, the reported results should not be interpreted as evidence that the models will maintain the same performance on future operational email streams.

Second, the present comparison focuses primarily on predictive performance. Although SOM offers useful possibilities for visualization and structural interpretation, and BERT provides highly expressive contextual representations, the explainability of the final classification decisions remains an important issue. In operational cybersecurity applications, identifying why a particular message has been classified as spam can be nearly as important as achieving a high classification rate. Future research should therefore investigate explainability techniques such as SHAP and LIME and assess whether explanations can be generated without substantially compromising predictive performance.

Third, computational efficiency deserves further consideration. BERT provides strong predictive performance but generally requires considerably more computational resources than simpler neural or statistical models. This consideration becomes particularly important in real-time or resource-constrained email-filtering environments. Future studies should therefore

compare the predictive performance and computational requirements of compact transformer architectures, including DistilBERT and TinyBERT, with those of the full BERT model.

Several directions for future research follow directly from these findings. One important extension is the development of multilingual spam-detection models. Email communication is inherently multilingual, whereas many existing benchmark sets of data are dominated by English-language messages. Evaluating BERT and related transformer architectures on multilingual corpora could therefore provide a more realistic assessment of their applicability in international communication environments.

A second direction concerns adversarial robustness. Modern spam campaigns may deliberately manipulate spelling, punctuation, formatting, hyperlinks, and semantic content to evade automated detection. Future experiments should therefore evaluate the resilience of BERT and SOM against controlled obfuscation and adversarial perturbations. Such experiments could provide a more demanding assessment of their effectiveness under realistic attack conditions.

A third avenue involves hybrid architectures. The present results reveal complementary characteristics: BERT provides strong contextual representation and predictive performance, whereas SOM provides an explicit topological organization of the feature space. Combining these properties could potentially produce models that retain much of BERT's representational power while providing improved visualization or interpretability. Attention-guided self-organization, manifold-learning approaches, or hybrid metaheuristics architectures represent promising directions for investigation.

Finally, future research could extend the analysis beyond the binary distinction between *spam* and *ham*. Real-world email filtering involves several potentially overlapping categories, including advertising, phishing, malware distribution, fraud, and legitimate commercial communication. A multiclass framework could therefore provide a more detailed representation of email threats and allow the models to distinguish not only whether an email is unwanted but also the nature of the associated threat.

Metrics	Min	Max	Median	Avg.	Std. dev.
Set of data A					
Precision	0.933	0.962	0.947	0.950	0.006
Recall	0.909	0.948	0.929	0.930	0.008
Accuracy	0.919	0.949	0.940	0.941	0.002
Set of data B					
Precision	0.958	0.986	0.972	0.971	0.003
Recall	0.957	0.979	0.964	0.968	0.004
Accuracy	0.958	0.977	0.968	0.969	0.002
Set of data C					
Precision	0.996	1.000	0.997	0.998	0.001
Recall	0.997	0.998	0.997	0.997	<0.001
Accuracy	0.995	1.000	0.997	0.996	0.002
Set of data D					
Precision	0.955	0.965	0.961	0.960	0.004
Recall	0.965	0.972	0.970	0.969	0.005
Accuracy	0.949	0.957	0.953	0.953	0.002
Set of data E					
Precision	0.952	0.981	0.967	0.968	0.003
Recall	0.969	0.986	0.977	0.976	0.002
Accuracy	0.965	0.984	0.975	0.974	0.002

Table *1*: Summary statistics of the classification exercise obtained by BERT. Experiments have been performed over 30 different training/test partitions, and accuracy, precision, and recall have been computed over the 30 test sets.

Metrics	Min	Max	Median	Avg.	Std. dev.
Set of data A					
Precision	0.863	1.000	0.884	0.946	0.042
Recall	0.901	1.000	0.932	0.922	0.071
Accuracy	0.821	1.000	0.935	0.952	0.051
Set of data B					
Precision	0.952	0.973	0.963	0.962	0.008
Recall	0.953	0.982	0.967	0.972	0.009
Accuracy	0.959	0.970	0.964	0.962	0.012
Set of data C					
Precision	0.827	1.000	0.921	0.938	0.052
Recall	0.815	1.000	0.894	0.901	0.061
Accuracy	0.822	0.941	0.872	0.863	0.084
Set of data D					
Precision	0.932	0.972	0.963	0.968	0.014
Recall	0.903	0.952	0.932	0.942	0.017
Accuracy	0.904	0.947	0.925	0.927	0.015
Set of data E					
Precision	0.752	0.921	0.835	0.872	0.093
Recall	0.732	0.918	0.792	0.814	0.097
Accuracy	0.729	1.000	0.826	0.814	0.162

Table *2*: Summary statistics of the classification exercise obtained by SOM. Experiments have been performed over 30 different training/test partitions, and accuracy, precision, and recall have been computed over the 30 test sets.

Set of data	F1 (BERT)	F1 (SOM)
A	0.940	0.934
B	0.970	0.967
C	0.997	0.919
D	0.964	0.955
E	0.972	0.842

Table *3*: Sample F1-scores obtained over the different sets of data by both BERT and SOM and model over one partition of train/test.

Funding Information

Authors have received no funding for this contribution.

Author Contributions

Authors have equally contributed to this contribution.

Conflict of Interest

Authors have no conflict of interest to disclose about this contribution.

Ethics Statements

This research did not require IRB approval.

References

[1] Y. Kontsewaya, E. Antonov, and A. Artamonov (2021). "Evaluating the Effectiveness of Machine Learning Methods for Spam Detection," Procedia Computer Science, vol. 190, no. 2021, pp. 479–486, 2021, doi: https://doi.org/10.1016/j.procs.2021.06.056.

[2] I. AbdulNabi and Q. Yaseen (2021). "Spam Email Detection Using Deep Learning Techniques," Procedia Computer Science, vol. 184, no. 2021, pp. 853–858, doi: https://doi.org/10.1016/j.procs.2021.03.107.

[3] J. Rajesh Kumar, G. Mahalakshmi, and P. Sudarshan (2020). "Email Spam Detection Using Machine Learning Techniques," IARJSET, vol. 8, no. 6, pp. 189–193, Jun. 2020, doi: https://doi.org/10.17148/iarjset.2021.8632.

[4] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, and T. Shah (2022). "Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges," Security and Communication Networks, vol. 2022, pp. 1–19, Feb. 2022, doi: https://doi.org/10.1155/2022/1862888.

[5] G. Nasreen, M.M. Khan, M. Younus, B. Zafar, and M.K. Hanif (2024). "Email Spam Detection by Deep Learning Models Using Novel Feature Selection Technique and BERT," Egyptian Informatics Journal, vol. 26, no. 2024, pp. 100473–100473, Jun. 2024, doi: https://doi.org/10.1016/j.eij.2024.100473.

[6] Y. Guo, Z. Mustafaoglu, and D. Koundal (2023). "Spam Detection Using Bidirectional Transformers and Machine Learning Classifier Algorithms," Journal of Computational and Cognitive Engineering, vol. 2, no. 1, pp. 5–9, doi: https://doi.org/10.47852/bonviewJCCE2202192.

[7] A. Najam (2024). "Unlocking Spam with Transformers: Sentiment-Driven Detection via Fine-tuned BERT," Medium, Feb. 13, 2024.

https://medium.com/@areeshanajam275/from-word-embedding-to-transformer-based-architecture-fine-tuning-bert-for-text-classification-of-80f87906d63c (accessed Nov. 17, 2024).

[8] V. S. Tida and S. H. Hsu (2022). "Universal Spam Detection Using Transfer Learning of BERT Model," in Proceedings of the Annual Hawaii International Conference on System Sciences, Hawaii: University of Hawaii at Mānoa Hamilton Library, pp. 7669–7677. doi: https://doi.org/10.24251/hicss.2022.921.

[9] N.M. Gardazi, A. Daud, M.K Malik (2025). "BERT applications in natural language processing: a review". Artif Intell Rev 58, 166. https://doi.org/10.1007/s10462-025-11162-5.

[10] S.S.R. Subramanya Hemant Konduri, K. Netti (2024). "An Improved Email Spam Classification System Using Random Forest Classifier". In: Choudrie, J., Mahalle, P.N.,

Perumal, T., Joshi, A. (eds) ICT for Intelligent Systems. ICTIS 2024. Lecture Notes in Networks and Systems, vol 1110. Springer, Singapore. https://doi.org/10.1007/978-981-97-6678-9_23.

[11] N. Camatti, G. di Tollo, F., Gastaldi, F. Camerin (2025). "Cultural heritage reuse applying fuzzy expert knowledge and machine learning: Venice's fortresses case study". Regional Studies, Regional Science, 12(1), 225–251. https://doi.org/10.1080/21681376.2025.2472058.

[12] M. Corazza, G. di Tollo, G. Fasano, R. Pesenti (2021). "A novel hybrid PSO-based metaheuristic for costly portfolio selection problems". Ann Oper Res 304, 109–137. https://doi.org/10.1007/s10479-021-04075-3.

[13] M. Licen, M. Astel, S. Tsakovski (2023). "Self-organizing map algorithm for assessing spatial and temporal patterns of pollutants in environmental compartments: A review". Science of The Total Environment, Volume 878. https://doi.org/10.1016/j.scitotenv.2023.163084.

[14] G. di Tollo, S. Tanev, K.M. Slim, D. De March (2014). "Determining the Relationship Between Co-creation and Innovation by Neural Networks". In: Faggini, M., Parziale, A. (eds) Complexity in Economics: Cutting Edge Research. New Economic Windows. Springer, Cham. https://doi.org/10.1007/978-3-319-05185-7_3.

[15] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, P. Stamatopoulos (2001). "Stacking classifiers for anti-spam filtering of e-mail". CoRR. cs.CL/0106040. 10.48550/arXiv.cs/0106040.

[16] J.R. Méndez, F. Fdez-Riverola, F., Díaz, E.L. Iglesias, J.M. Corchado (2006). "A Comparative Performance Study of Feature Selection Methods for the Anti-spam Filtering Domain". In: Perner, P. (eds) Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining. ICDM 2006. Lecture Notes in Computer Science(), vol 4065. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11790853_9.

[17] G.V. Cormack, T. R. Lynam. (2007). "Online supervised spam filter evaluation". ACM Trans. Inf. Syst. 25, 3 (July 2007), 11–es. https://doi.org/10.1145/1247715.1247717.

[18] K.J. Gazal (2022). "Two-phase fuzzy feature-filter based hybrid model for spam classification". Journal of King Saud University - Computer and Information Sciences, Volume 34, Issue 10, Part B, 2022, Pages 10339-10355, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2022.10.025.

[19] A. Attar, R.M. Rad, R.E. Atani (2013). "A survey of image spamming and filtering techniques". Artif Intell Rev 40, 71–105. https://doi.org/10.1007/s10462-011-9280-4.

[20] N. Camatti, G. di Tollo, G., Filograsso, S. Ghilardi (2024). "Predicting Airbnb pricing: a comparative analysis of artificial intelligence and traditional approaches". Comput Manag Sci 21, 30. https://doi.org/10.1007/s10287-024-00511-4.

[21] M. Honnibal, I. Montani (2017). "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks, and incremental parsing".

[22] T. Verdonck, B. Baesens, M. Óskarsdóttir et al. (2024). Special issue on feature engineering editorial. Mach Learn 113, 3917–3928. https://doi.org/10.1007/s10994-021-06042-2.

[23] N.V. Chawla, K. W. Bowyer, L. O. Hall, W.P. Kegelmeyer (2002). "SMOTE: synthetic minority over-sampling technique". J. Artif. Int. Res. 16, 1 (January 2002), 321–357.

[24] E.H. Tusher, M. A. Ismail, M. A. Rahman, A. H. Alenezi and M. Uddin (2024), "Email Spam: A Comprehensive Review of Optimize Detection Methods, Challenges, and Open

Research Problems," in *IEEE Access*, vol. 12, pp. 143627-143657, doi: 10.1109/ACCESS.2024.3467996.

[25] A. Bhowmick, S. Hazarika (2016). "Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends". 10.48550/arXiv.1606.01042.

[26] V. Nannen, A.E. Eiben (2007). "Relevance estimation and value calibration of evolutionary algorithm parameters." In M. M. Veloso (Ed.), IJCAI 2007, proceedings of the 20th international joint conference on artificial intelligence (pp. 1034–1039).

[27] A. E. Eiben, J.E. Smith (2003). "Introduction to Evolutionary Computing". Berlin: Springer.