American Journal of Engineering Research (AJER) **2026**
American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN: 2320-0936
Volume-15, Issue-4, pp-29-35
www.ajer.org
Research Paper Open Access

Structural and Optimization-Based Comparison of Binary Classifiers for Spam Detection

Stefka Popova [1], Hristo Nenov [2]
[1] *Software and Internet Technologies department, Technical University of Varna, Bulgaria*
[2] *Software and Internet Technologies department, Technical University of Varna, Bulgaria*

ABSTRACT : *Spam detection is one of the most common examples of classification task that deals with high-variance data and large number of attributes. Machine learning algorithms has been used in vast number of research to solve this binary task, most of them focused on empirical estimations rather than on going deeper into details of structural properties or different optimization approaches. This paper proposes a comparative framework for some of the most popular classifiers based on supervised machine learning methods, including probabilistic (Naïve Bayes), discriminative (Support Vector Machine, Neural Network, Logistic Regression) and ensemble and tree-based (Decision Tree and Random Forest) models. All classifiers are considered and evaluated upon two main criteria: (a) their structural characteristics as capacity, robustness or interpretability, and (b) suitable optimization approaches with smoothing, regularization or convergence among others. Findings confirm the assumption that there is no universally best classifier and usually its effectiveness depends on balance between model complexity and optimization sensitivity. The proposed framework widens understanding of how binary classifiers work beyond quantitative metrics and could be helpful in the moment of model selection for solving spam detection task.*

KEYWORDS *spam filtering, machine learning, binary classification.*

Date of Submission: 03-04-2026 Date of acceptance: 14-04-2026

I. INTRODUCTION

Spam continues to be ongoing problem in modern electronic communications. Building and implementing efficient spam filters is challenging task because of many reasons including varying nature of emails sent, working with very large feature sets, dealing with noise inside data, and more. Over the past years, different machine learning algorithms have been for creating spam detection models used in attempts to overcome the imposed challenge. Models' effectiveness is usually evaluated in terms of classification accuracy and error, frequently combined with metrics such as precision, recall and F1-score.

Most of existing numerous studies are focused on model performance in quantitative terms rather than qualitative ones, ignoring the influence of model properties and their optimization behavior over their effectiveness and practical suitability. Since classifiers differ in their characteristics, assumptions and the way of solving classification tasks, it is important to form better understanding and thus improve model selection choice.

The present study proposes structured framework that compares different supervised machine learning methods used for spam detection. The conducted analysis is done with focus on two main criteria: models structural characteristics and the possible optimization approaches. Consideration of both criteria in addition of performance-driven analysis should give better understanding and help create a systematic perspective of model selection as part of the spam filter design.

The problem of spam detection has been extensively researched, and there are lot of published studies that deal with application of machine learning algorithms as mean to solve the binary classification problem.

The theoretical foundations for the classifiers considered in the current paper are well established in the literature. Thorough analysis of probabilistic methods such as Naïve Bayes and logistic regression, and their optimization properties, is presented in [1]. Systematic and in-depth study of statistical learning methods, including support vector machines and tree-based methods, as well as their respective characteristics, is offered in [2]. The theory behind neural networks with their optimization challenges and regularization strategies, is

discussed in [3]. All these previous works set up the mathematical framework that forms the basis for the structural properties analysis presented in this paper.

Many surveys are available for application of machine learning algorithms in spam detection. Overview of email spam filtering techniques, pointing the predominance of Naïve Bayes and SVM in early systems, is discussed in [4]. Application of different algorithms for detection of spam distributed via various channels is reviewed in [5], and the conclusion there is no universally best algorithm is drawn and validated independent on the channel type. The importance of feature selection process and methods for feature reduction are discussed in [6], emphasizing their influence over classification accuracy.

Direct experimental comparisons of classifiers for spam detection form the closest body of work to the present study. A comparison of Naïve Bayes, decision trees, and SVM in terms of spam detection was conducted in [7] and reports best performance by SVM-based model that comes with greater training time. SVM and different boosting methods are considered for solving spam detection task in [8], with SVM-based model again outperforming the rest examined methods. More recently, random forests were examined and compared against other classifiers [9] and demonstrated better noise robustness while being used for spam detection. Focus on kernel selection importance and the hyperparameter tuning for SVM-based model is offered in [10]. Different experimental comparisons of supervised machine learning-based models are reported in [11, 12, 13], as the research in the field is an ongoing process.

While all these studies bring very valuable information about different aspects of spam detection process, most of them focus mainly on performance metrics such as accuracy, precision, and recall. There are few attempts to explore the reasons why certain algorithms perform better or worse while solving spam recognition problems. Considerations about noise sensitivity or interpretability as part of the algorithm structural properties are not properly discussed, as well as the influence that some optimization approaches have over the model performance.

The present paper addresses these gaps by proposing a comparative framework that evaluates different prominent machine learning-based binary classifiers. The evaluation is focused on two criteria: structural properties and optimization sensitivity. By investigating the nature of these characteristics, we contribute to a better understanding of model behavior within spam detection contexts.

II. CLASSIFIER ANALYSIS: FORMULATION AND PROPERTIES

The experimental results cited throughout this section are drawn from authors previous work [14, 15, 16], where we evaluated six machine learning-based classifiers on a dataset of approximately 14,000 emails combining existing public SpamAssassin corpus for emails written in English language and personal correspondence in a span of 5 years for emails written in Bulgarian language. The classifier performances were evaluated using standard spam detection metrics. Here, we focus on synthesizing these findings through a structural and optimization-based comparative framework.

Spam detection is a classic task of binary classification where received email messages should be categorized in two categories – spam and legitimate messages, or ham. Solution of such task should be able to find meaningful text patterns and to handle misspellings, different obfuscation techniques and expect changing behavior.

Machine learning algorithms have become very popular in solving binary classification problems. There are many reasons for such popularity, some of which are their ability to represent complex functions, find patterns in high-dimensional data, supply of various techniques to prevent overfitting, offer possibility to automate the whole classification process, and many more. While training machine learning models with exiting data is necessary, most algorithms can successfully handle new data.

Different algorithms have different mathematical properties, but the general classification model for solving binary classification task could be formally described as:

$$(x_i, y_i), \qquad y \in \{0, 1\},$$

where (x_i, y_i) represents data point from given dataset with x_i being a feature vector for the *i*-th example, and y_i – its output.

Each algorithm solves classification tasks in its own way, but the decision function could be formally defined as:

$$\hat{y} = f(x;\, \theta),$$

where $\hat{y}$ denotes the predicted label, θ represents the model parameters, and f is the mapping function.

The purpose of decision function is to minimize the loss function, which shows the discrepancies between predicted and correct label:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L\big(y_i, f(x_i; \theta)\big) + \Omega(\theta)$$

There is different loss functions used depending on the problem. Cross-entropy loss, hinge loss, and squared hinge loss are some possible functions to be applied for model performance estimation with cross-entropy being usually the default choice loss function. While hinge loss is margin-based and thus suitable for maximum-margin classification, cross-entropy evaluates divergence between predicted and actual label.

1) Naïve Bayes

Naïve Bayes algorithm is example of supervised machine learning algorithms used mainly for solving classification problems. Its usage is appropriate in cases where there are lot of features, and is based on assumptions for feature independence for class *j*:

$$P(y|x) \propto P(y)\prod_{j} P(x_j|y)$$

While the assumption is not true in most cases, having such consideration usually significantly simplifies estimations.

To prevent zero-probability issues leading to incorrect classification, additive smoothing technique is applied. Depending on the value added there are Laplace smoothing ($\alpha = 1$) and Lidstone smoothing ($\alpha < 1$), and the probability for N features is calculated as:

$$\hat{P}(x_i|y) = \frac{count(x_i, y) + \alpha}{count(y) + \alpha N}$$

Structural properties

Naïve Bayes algorithm uses low-capacity, high-bias, and probabilistic generative models. Depending on feature distribution, its decision boundaries could be considered both linear (multinomial Bayes) and non-linear (Gaussian with different variance). Some of the advantages of model usage include computational efficiency, very good interpretability, and low overfitting tendency leading to accurate classification results.

Optimization sensitivity

Naïve Bayesian classifier is suitable for solving classification problems in case of high-dimensional data and is highly scalable. Applying some type of smoothing improves model robustness making it one of most stable, fast-converging and low-variance classifiers.

2) Logistic Regression

Another widely used supervised machine learning algorithm is used for classification problems. It calculates an event occurrence probability using sigmoid function for transformations. Model is defined as:

$$P(y = 1 \mid x) = \sigma(w^T x + b)$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function, and w and b are model parameters.

Model training is formulated as the minimization of the regularized empirical risk:

$$\min_{\omega,b} \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i(\omega^T x_i + b)}\right) + \lambda\|w\|_2^2$$

where the second term represents L2 regularization, controlling model complexity.

Structural properties

Logistic Regression defines a linear decision boundary and belongs to the class of parametric models with moderate capacity. The algorithm is frequently used as baseline for spam detection due to its computational efficiency and low implementation complexity. It is part of the linear algorithms and is characterized with high interpretability and fast training process, but at the same time shows overfitting tendency.

Optimization sensitivity

To overcome that overfitting tendency Logistic Regression uses so called regularization that has positive influence over classification accuracy. Logistic regression has also high sensitivity towards data scaling and often applies normalization to compensate for that weakness. Another challenge faced by the method is data correlation that could lead to misleading and incorrect coefficient interpretation.

3) Support Vector Machine

Support Vector Machine (SVM) is a machine learning algorithm used both for classification and regression tasks, which tries to find the optimal hyperplane to separate different classes. For linearly separable data, the optimization problem can be formulated as

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

which is subject to

$$y_i(w^T x_i + b) \geq 1$$

For non-separable data, slack variables are introduced, leading to soft-margin formulation:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i$$

where C controls the trade-off between margin maximization and classification error.

Alongside linear data, through kernel functions SVMs can represent non-linear decision boundaries without explicit feature transformation which makes it suitable for solving complex tasks.

Structural properties

SVMs control model capacity through margin maximization, implementing the principle of structural risk minimization. The optimization problem is convex, ensuring global optimum. Compared to linear models, SVMs can achieve higher representational flexibility when kernels are used, though at increased computational cost.

Optimization sensitivity

Kernel choice is one of the factors influencing model's performance since it directly affects the feature space and its transformation. SVM is also dependent on regularization that could prevent overfitting or affect its generalization capabilities. One point to be considered is the high-computational nature of the algorithm contrasted with the linear models.

4) Decision Tree

Decision Tree represents mathematical structure of connected nodes, where each node defines state of system and nodes are connected via threshold function [17]. It is intuitive algorithm that breaks down datasets into smaller datasets according to simple rules. Standard decision tree consists of root (the node represents whole dataset and is located on the top of the structure), branches (representing outcome of a given node and connecting it to other nodes), intermediate nodes (representing different tests), and leaf-nodes, where no more feature division is happening.

Decision trees use recursive partitioning using the calculation of so-called information gain:

$$IG = H(parent) - \sum_{k} \frac{|D_k|}{|D|} H(D_k)$$

Structural properties

Decision Tree is example of non-probabilistic discriminative model, which has non-linear decision boundary and could be considered to have moderate to high capacity. It is recognized for its very good interpretability and has no other parameters than the depth of the tree limiting number of dataset splits or minimally required information gain. Unfortunately, the algorithm is very prone to overfitting, which can be prevented by applying technique called pruning or by combining different trees in ensemble.

Optimization sensitivity

While it is very easy to understand and interpret, decision trees are not stable models and are very sensitive to change of data. On the other hand, such models are not feature scaling-dependent and offer iterative solutions, require minimal data preparation and handle well cases of missing values. Decision tree models could be very suitable for non-linear relationships detection unlike regression models.

5) Random Forest

Random Forest is example of ensemble learning techniques that could be used for solving problems of classification, regression or unsupervised machine learning. Such models consist of various decision trees and for classification, such as spam detection problems, the final predicted output label is defined by the majority vote between all trees. The technique is called "random" since different trees are trained with different samples from the input dataset and not taking into consideration all the input features.

Ensemble model is formally described as

$$\hat{y} = majority\ vote\ \{T_m(x)\}$$

Structural properties

Combining decision trees in one model makes it more complex in terms of computational cost and interpretability but introduces possibility of parallelized work with minimal preprocessing. The ensemble model is high-capacity non-linear and non-parametric model which is resistant to overfitting and usually achieve high classification accuracy.

Optimization sensitivity

Random Forest method handles successfully high-dimensional non-linear data better than Logistic Regression and is regularization independent. It has stable convergence, good noise resilience and is scale-invariant model. Optimization of such model usually includes tuning of hyperparameters like maximum depth of individual trees, number of features or number of trees to be included in the ensemble.

6) Neural Networks

Neural Networks is example of non-linear probabilistic methods and represent complex model which is capable of hidden data dependency detection. It consists of input and output layers with one or more hidden layers between them. The non-linearity is introduced by different types of activation functions such as ReLU or sigmoid function. For a simple two-layer network

$$f(x) = \sigma(W_2\phi(W_1x + b_1) + b_2)$$

where ϕ represents a hidden-layer activation function and σ is usually the sigmoid function for binary classification.

Training of Neural Network model is an iterative process dedicated to loss function minimization:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i; \theta))$$

and uses optimization algorithms like gradient descent.

Structural properties

Neural Networks are models with layered architectures which facilitate their ability to learn hierarchical data representation. The non-linear nature of such models makes it highly suitable for approximating complex decision boundaries and thus effective for identification of spam patterns in large datasets. While highly effective, the Neural Networks are usually characterized by low result interpretability compared to tree-based or linear methods.

Optimization sensitivity

Neural Network model is very sensitive to hyperparameter tuning, including number of layers and neurons, batch size or learning rate. Another drawback is that such models expect high-dimensional data to achieve better results, meaning the model is also sensitive to data scaling and usually requires extensive preprocessing to display stable behavior.

III. COMPARATIVE DISCUSSION

The comparative perspective of this study is useful for selecting and configuring models for binary classification in spam detection task. Proposed framework supports more systematic model evaluation beyond usual performance metrics by considering both structural properties and sensitivity to optimization.

Table I considers all previously discussed models in terms of their structural properties, while Table II presents their optimization sensitivity to different approaches.

The analysis shows that supervised machine learning algorithms for spam detection differ on both criteria and these differences affect their stability and performance.

Table I Structural comparison between machine learning algorithms

Algorithm	Computational Complexity	Implementation Complexity	Result Interpretabilty	Noise Sensitivity	Overfitting Tendency
Naïve Bayes	low	low	high	moderate	low
Logistic Regression	low	low	high	low	low
Support Vector Machine	high	moderate	low	high	moderate
Decision Tree	low	low	high	low	high
Random Forest	moderate	moderate	low	low	low
Neural Network	hidh	high	low	low	high

Table II Optimization sensitivity of machine learning algorithms

Algorithm	Input Attributes Sensitivity	Min Document Frequency Sensitivity	Threshold Sensitivity	Regularization / Smoothing Sensitivity	Pruning Sensitivity	Iterations Sensitivity
Naïve Bayes	low	moderate	low	high	not applicable	not applicable
Logistic Regression	medium	moderate	moderate	high	not applicable	moderate
Support Vector Machine	high	moderate	high	high	not applicable	moderate
Decision Tree	low	low	not applicable	not applicable	high	low
Random Forest	low	low	low	low	low	moderate
Neural Network	high	moderate	moderate	high	not applicable	high

From structural point of view, Naïve Bayes represent low-capacity generative model and has strong independence assumptions resulting in stable behavior and low overfitting risk. Logistic Regression and linear Support Vector Machines stay in the middle of the capacity spectrum and provide linear decision boundaries with controlled through regularization or margin maximization complexity. Decision Trees present higher variance and noise sensitivity, while Random Forests reduce this effect by ensemble averaging. Highest representational capacity is demonstrated by Neural Networks which allows complex non-linear decision boundaries at increased overfitting risk.

Clear differences between classifiers can be observed also from optimization sensitivity perspective. Logistic Regression and Support Vector Machines models benefit from stable optimization and convergence while Neural Networks rely on non-convex iterative training making them more sensitive to hyperparameter choices. Pruning strategies affect performance of tree-based models, while Naïve Bayes requires less tuning, usually related to feature dimensionality and smoothing.

The analysis shows that classifier suitability depends on balance between its structural capacity and optimization level control. Models with higher expressive power often require more careful tuning to achieve good generalization, while simpler models tend to be more stable and easier to configure. For this reason, model selection in spam detection should consider not only classification accuracy but also optimization effort, data characteristics and interpretability.

In high-dimensional and sparse text data, models with limited capacity often provide stable performance with minimal tuning. Linear models offer controlled complexity through regularization and benefit from stable optimization. On the other hand, more complex models provide better flexibility but expect more careful tuning and training.

Tree-based and ensemble methods could be considered as middle ground, combining non-linear decision boundaries with mechanisms for model complexity control. Appropriate parameter settings have positive effects over performance and stability of such models.

The proposed perspective is supported by authors previously conducted experimental studies in spam detection [14, 15, 16]. The results show that model performance is related to both representational capacity and optimization settings. More complex models tend to perform better when properly tuned, while simpler models remain stable even with limited optimization.

Overall, appropriate model selection requires balance between data characteristics, interpretability, computational resources, and optimization. This can lead to more consistent and practical model configurations in real-world applications for spam detection.

IV. CONCLUSION

This study presented a structured comparative framework for analyzing commonly used binary classification algorithms in spam detection. Combination of structural model characteristics with their optimization sensitivity in the proposed approach highlights the relationship between model capacity and optimization requirements and goes beyond simple performance-based comparisons.

The analysis shows that differences in model behavior are closely related to their structural assumptions and optimization properties. Instead of identifying a single best-performing model, the results emphasize that model suitability depends on the specific context, including computational constraints, configuration choices and data characteristics.

By considering all these aspects together in one unified perspective, the study contributes to a clearer understanding of how binary classifiers for spam detection can be selected and applied in practice.

REFERENCES

[1]. Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2006.
[2]. Hastie, T., Tibshirani, R., & Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer, New York, 2009.
[3]. Goodfellow, Ian, et al. *Deep Learning*. MIT Press, 2016.
[4]. Cormack, Gordon V. "Email Spam Filtering: A Systematic Review." *Foundations and Trends in Information Retrieval*, vol. 1, no. 4, 2008, pp. 335-455. *Now Publishers*
[5]. Guzella, Thiago S., and Walmir M. Caminhas. "A Review of Machine Learning Approaches to Spam Filtering." *Expert Systems with Applications*, vol. 36, no. 7, 2009, pp. 10206-22. *ScienceDirect*, https://doi.org/10.1016/j.eswa.2009.02.037
[6]. M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," J Big Data, vol. 2, no. 1, Dec. 2015, doi: 10.1186/s40537-015-0029-9
[7]. Metsis, Vangelis & Androutsopoulos, Ion & Paliouras, Georgios. (2006). Spam Filtering with Naive Bayes - Which Naive Bayes?. In CEAS
[8]. Drucker, Harris, Donghui Wu, and Vladimir N. Vapnik. "Support Vector Machines for Spam Categorization." *IEEE Transactions on Neural Networks*, vol. 10, no. 5, 1999, pp. 1048-54. *IEEE Xplore*
[9]. Jindal, Nitin & Liu, Bing. (2008). Opinion Spam and Analysis. 10.1145/1341531.1341560.
[10]. Blanzieri, E. and Bryl, A. (2008). A survey of learning-based techniques of email spam filtering. Artificial Intelligence Revolution, 29:63–92
[11]. Zhang, Chenwei. (2025). Enhancing Spam Filtering: A Comparative Study of Modern Advanced Machine Learning Techniques. ITM Web of Conferences. 70. 10.1051/itmconf/20257004013
[12]. Kumar, Rahul & Garg, Gaurav. (2025). A Comparative Analysis of Machine Learning and Deep Learning Architectures for Spam Mail Classification. Journal of Artificial Intelligence, Machine Learning and Data Science. 4. 3000-3008. 10.51219/JAIMLD/rahul-kumar/622
[13]. Borotić, Gordana & Granoša, Lara & Kovačević, Jurica & Bagic Babac, Marina. (2024). Effective Spam Detection with Machine Learning. Croatian Regional Development Journal. 4. 43-64. 10.2478/crdj-2023-0007
[14]. Popova, Stefka, Hristo Nenov, and Donika Stoyanova. "Spam Detection System Based on Hybrid Scorings." *International Conference Automatics and Informatics, ICAI 2024,* DOI: *10.1109/ICAI63388.2024.10851565*
[15]. Popova, Stefka, Hristo Nenov, and Velislav Kolesnichenko. " An Automatic Spam Detection System Based on Hybrid Scoring Models." *International Conference Automatics and Informatics, ICAI 2024,* DOI: *10.1109/ICAI63388.2024.10851541*
[16]. Popova, Stefka, Hristo Nenov, and Genoveva Dimitrova. " Evaluating Neural Network Parameter Effects on Spam Classification Accuracy." *International Conference Automatics and Informatics, ICAI 2025,* DOI: *10.1109/ICAI67591.2025.11324459*
[17]. Friedland, Gerald, and Christian Christian. *Information-Driven Machine Learning*. Cambridge University Press, 2019