Indonesian Journal of Electrical Engineering and Computer Science
Vol. 37, No. 1, January 2025, pp. 356∼364
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v37.i1.pp356-364

Enhanced SMS spam classification using machine learning with optimized hyperparameters

Nasereddine Hafidi[1], Zakaria Khoudi[1], Mourad Nachaoui[1], Soufiane Lyaqini[2]
[1]Equipe Mathématiques et Interactions, Faculté des sciences et Techniques, Sultan Moulay Slimane University, Beni Mellal, Morocco
[2]Ecole Nationale des Sciences Appliquees, LAMSAD Laboratory, Hassan First University, Settat, Morocco

Article Info

Article history:

Received Jun 2, 2024
Revised Sep 8, 2024
Accepted Sep 29, 2024

Keywords:

Classification
Genetic algorithm
Hyperparameter tuning
SMS classification
Spam detection
Supervised learning

ABSTRACT

Short message service (SMS) text messages are indispensable, but they face a significant issue with spam. Therefore, there is a need for robust models capable of classifying SMS messages as spam or non-spam. Machine learning offers a promising approach for this classification, based on existing datasets. This study explores a comparison of several techniques, including logistic regression (LR), support vector machines (SVM), gradient boosting (GB), and neural networks (NN). Hyperparameters play a crucial role in the performance of these models, and their optimization is essential for achieving high accuracy. To this end, we employ an evolutionary programming approach for hyperparameter optimization. This approach evaluates the performance of these models before and after hyperparameter optimization, aiming to identify the most effective model for SMS spam classification.

Corresponding Author:

Nasereddine Hafidi
Equipe Mathématiques et Interactions, Faculté des sciences et Techniques, Sultan Moulay Slimane University
Beni Mellal 23000, Morocco
Email: nasereddine.hafidi@usms.ma

1. INTRODUCTION

Short message service (SMS) text messages are indispensable in modern communication, yet they face a significant challenge from spam messages. In general, spammers use these messages to promote their utilities or businesses. Spam messages can be annoying and, in some cases, harmful to recipients. Sometimes, users can also suffer financial losses due to these spam messages, making it crucial to develop robust models that can effectively classify SMS messages as spam or non-spam. Machine learning provides a promising approach to tackle this problem by leveraging existing datasets to train models for accurate classification [1], especially for spam classification [2].

Numerous machine-learning methods can be used for SMS spam classification. Each method has strengths and weaknesses, and choosing the most appropriate algorithm is challenging. Additionally, the performance of these algorithms can be highly dependent on the proper tuning of hyperparameters. Despite the variety of comparative studies, adjusting hyperparameters can lead to substantial changes in algorithm performance, adding another layer of complexity to the problem.

This study aims to develop a robust machine-learning architecture for classifying SMS messages into spam or non-spam. We will compare various machine learning techniques, including logistic regression (LR) [3], support vector machines (SVM) [4], random forest (RF) [5] and gradient boosting (GB) [6], and to select the most suitable model for optimal results. The performance of each model relies heavily on a set of hyperparameters, which play a crucial role in influencing the results. Even small changes in these hyperparameters can significantly change the model's performance and the overall set of hyperparameters.

To ensure the model's robustness and optimal performance, we will employ evolutionary programming [7] for hyperparameter optimization. This technique will iteratively search for the best set of hyperparameters, ensuring high accuracy and generalizability to new data. The study will compare the performance of models before and after optimization to determine if the optimized models outperform their manually tuned counterparts. The primary contribution of this paper is to propose a comparison of supervised learning approaches, considering the hyperparameter settings for each model, to classify SMS messages as spam or non-spam. By examining various features, such as message content, length, and sender information, we aim to identify the most significant predictors of spam messages. The resulting models will accurately identify spam messages, enabling effective filtering and management. This paper also provides a comparative analysis of models with and without hyperparameter optimization, ensuring the selection of the best model for final classification.

The paper is organized as follows: we begin with a problem statement, outlining the mathematical formulation of our model and exploring the influence of hyperparameters on the results. We then review related works, present our proposed approach, and conclude with our experimental results.

2. SETTING OF THE PROBLEM

Supervised learning [8] is a branch of machine learning that uses labeled data to train algorithms to make predictions and recognize patterns. The supervised learning problem can be summarized as the following minimization problem:

$$\min_{h_{\mathcal{T}} \in H} \mathcal{J}(h) := \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i)) + \beta |h|^2 \tag{1}$$

Here, $h_{\mathcal{T}}$ denotes a function from the hypothesis space $\mathcal{H}$, $\mathcal{T}$ represents the training dataset with n samples, $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{n}$, and ℓ is a non-negative loss function that evaluates the disagreement between y and $h(x)$. The term β is the regularization parameter. The goal is to classify observations into one of two classes, -1 or $+1$. Good classification involves minimizing errors and maximizing the similarity between training and validation data. Two crucial steps are required: selecting the right model and finding the optimal set of hyperparameters for that model.

Choosing the best model involves comparing models based on similarity metrics, but finding the optimal hyperparameters remains a critical and challenging task. Manually identifying the best set of hyperparameters is often impractical. To illustrate the influence of hyperparameters on model performance, we varied key hyperparameters for each model and observed the resulting changes in evaluation metrics. Table 1 shows the effects of varying the inverse regularization parameter (C) in LR, which controls the regularization strength. A low C value (e.g., 0.5) results in heavy regularization, leading to lower scores across all metrics. Conversely, a higher C value (e.g., 2.5) weakens regularization, resulting in higher scores but increasing the risk of overfitting.

Table 1. Logistic regression-chaging of metrics value while changin the hyperparameter C

	C				
	2.5	2.0	1.5	1.0	0.5
F1-Score	0.8755	0.8664	0.8524	0.8085	0.7069
Precision score	0.9819	0.9816	0.9811	0.9793	0.9870
Recall score	0.7898	0.7754	0.7536	0.6884	0.5507

Beyond LR, we also examined the effects of hyperparameter variation on other models, including SVM, RF, and GB. The Table 2 summarizes the results obtained by varying hyperparameters for each model. According to Table 2, our analysis demonstrates the importance of hyperparameter tuning in machine learning models. Several methods exist for optimizing hyperparameters [9], [10]. Techniques such as evolutionary programming, specifically genetic algorithms, can effectively optimize hyperparameters based on various evaluation metrics. We can enhance model performance and ensure better generalization to new data by selecting the optimal set of hyperparameters.

Table 2. Hyperparameter change influence about all used models

	LR		SVM		Random forest		GB		
	C		C / Kernel		Max depth		Lear Rate / max iter / max depth		
	2.5	1.0	1 / 'rbf'	0.1 / 'rbf'	100	80	0.5/100/3	0.2/200/3	0.1/80/2
F1-score	0.8755	0.8085	0.9027	0.6132	0.9105	0.9019	0.8923	0.8923	0.7826
Precision score	0.9819	0.9793	0.9747	0.9897	0.9832	0.9829	0.9508	0.9508	0.9782
Recall score	0.0798	0.6884	0.8406	0.4423	0.8478	0.8333	0.8406	0.8406	0.6522

3. RELATED WORKS

Machine learning, a branch of artificial intelligence, focuses on identifying complex patterns in data for predictions [11], classifications [12], [13], or advanced exploratory data analysis [14], [15]. Its applications have become widespread in various domains, including text classification [16] and spam detection [17]–[19]. The success of machine learning in these areas leverages historical data, enabling the use of supervised learning techniques for regression and classification tasks. Effectively training a classification model can be crucial for various applications, such as filtering spam messages to improve communication efficiency and security [20], [21].

Several machine-learning classification techniques have been applied to SMS spam detection. For instance, studies like [22] have explored logistic regression for spam classification, demonstrating its effectiveness in handling large text datasets. Similarly, [23], [24] applied SVM to classify spam messages, highlighting its robustness in distinguishing between spam and non-spam texts. GB, which has variants, has also been employed in this domain, as shown in [25], [26], which enhanced classification accuracy through iterative learning processes. Neural networks, known for capturing complex patterns, have been utilized in [27], [28] for SMS spam detection, showcasing their potential to improve classification performance.

In addition to individual machine-learning techniques, several studies have investigated the comparative performance of different models for spam detection. For instance, Sumathi *et al.* [29] compared various classification methods, including logistic regression, SVM, decision tree, and RF, to determine the most effective model for SMS spam detection. The study emphasized the importance of hyperparameter tuning in optimizing model performance. Hyperparameters significantly influence the results, and their optimal settings can lead to substantial improvements in classification accuracy [10].

Evolutionary programming has been identified as an effective method for hyperparameter optimization. Studies like Fogel [7] have demonstrated the utility of evolutionary programming in iteratively searching for the best set of hyperparameters, ensuring models achieve high accuracy and generalize well to new data. The need for a study comparing selected methods with manually tuned and optimized hyperparameters is evident. Such comparisons help in understanding the impact of hyperparameter settings on model performance and determining whether the effectiveness of the best method changes with different hyperparameter configurations. In summary, this study aims to fill the gap by comparing machine learning models for SMS spam detection, both with and without hyperparameter optimization. This approach will help identify the most effective model and understand the role of hyperparameters in enhancing classification performance.

4. PROPOSED METHOD

The key idea of our approach is to find all the hyperparameters used for model training simultaneously. The optimal set of hyperparameters we search for ensures the robustness of the constructed architecture. This optimal set should provide good results on the validation part of the dataset. In this section, we detail our proposed approach for optimizing the hyperparameters of our model. First, we introduce the used algorithm, outlining its fundamental principles. We then describe how this algorithm is employed in our method to find the optimal hyperparameter set and the convergence conditions that determine when the optimization process is complete.

4.1. Genetic Algorithm

Genetic algorithm (GA) [30] is a search heuristic that mimics natural selection. It generates high-quality optimization and search problem solutions using bio-inspired operators such as mutation, crossover, and selection. Here is a simplified pseudo-code of the genetic algorithm.

The GA we employ, as outlined in Algorithm 1, is used to optimize the hyperparameters of our model. This algorithm starts by initializing a population of hyperparameter configurations, which are then evaluated using a fitness function, which, in our case, is a combined score between F1-score, precision, and recall. The best configurations are selected for reproduction through crossover and mutation operations, generating a new population. This process continues until convergence criteria are met.

Algorithm 1. Genetic algorithm

Input: Population size, mutation rate, crossover rate, termination criteria
Output: Best solution
Initialization: Create initial random population
While *Termination criteria not met* **do**
  Evaluation: Assess fitness of each individual
  Selection: Choose the fittest individuals for reproduction
  Crossover: Combine pairs of individuals to produce offspring
  Mutation: Apply random changes to offspring
  Replacement: Form new population from parents and offspring
End

4.2. Method

Figure 1 summarizes our proposed architecture, which follows a structured approach to optimize hyperparameters for our model. Initially, we perform dataset pretreatment. Following this, we use the GA to search for the optimal hyperparameters based on predefined metrics. This involves training the model with various hyperparameter sets and using the algorithm to generate and assess new sets iteratively. Once the algorithm converges, we select the best hyperparameter set identified. This optimal set is then utilized for final model training and prediction.

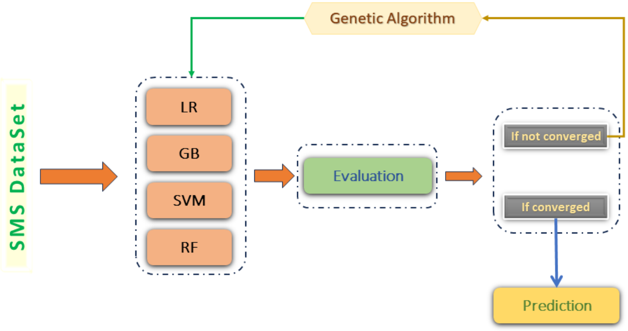

Figure 1. Proposed approach architecture

As shown in the architecture in Figure 1, our model starts with a pretreatment of the collected dataset. Then, the hyperparameter optimization process begins with a training/evaluation step. In this step, the genetic algorithm trains the chosen model with the current set of hyperparameters. Next, we evaluate the model by combining the selected metrics (F1-score, precision, and recall) with a weighted approach as follows:

$$\text{Combined score} = 0.4 \times (1 - \text{F1-score}) + 0.3 \times (1 - \text{Precision}) + 0.3 \times (1 - \text{Recall}) \tag{2}$$

Using these metrics and their specific weights is essential to balance different aspects of model performance.

- F1-Score: the F1-score is the harmonic mean of precision and recall, providing a single metric that balances false positives and false negatives.
- Precision: measures the accuracy of the positive predictions, calculated as the ratio of true positive predictions to the total positive predictions (true positives+false positives).
- Recall: measures the model's ability to identify all relevant instances, calculated as the ratio of true positive predictions to the total actual positives (true positives+false negatives).

The weights assigned to each metric (0.4 for F1-score, 0.3 for precision, and 0.3 for recall) reflect their relative importance in our evaluation. The higher weight for F1-score ensures that the balance between precision and recall is prioritized. Precision and recall are equally weighted to ensure that both metrics are considered without overwhelming the F1-score's influence. This combined score allows us to minimize the overall error of the model by addressing different types of misclassifications through a balanced and comprehensive evaluation strategy.

After the evaluation, if the algorithm does not converge, the GA updates the current set of hyperparameters and trains the model again. We repeat this until convergence is achieved. Once the algorithm converges, we proceed to the classification phase with the best set of hyperparameters found.

5. RESULTS AND DISCUSSIONS

This section presents the primary findings of our study and interprets them about the set objectives. Standardized benchmark datasets are often useful for assessing this approach. We opted to evaluate the proposed method using a real-life dataset from Kaggle, known as the SMS spam collection: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset. Figure 2 shows that the percentage of non-spam is too high (87.37%) compared to the percentage of spam messages. So, the data needs to be more balanced. Data distribution analysis reveals a significant imbalance, emphasizing the need for precise hyperparameter optimization to achieve robust models.

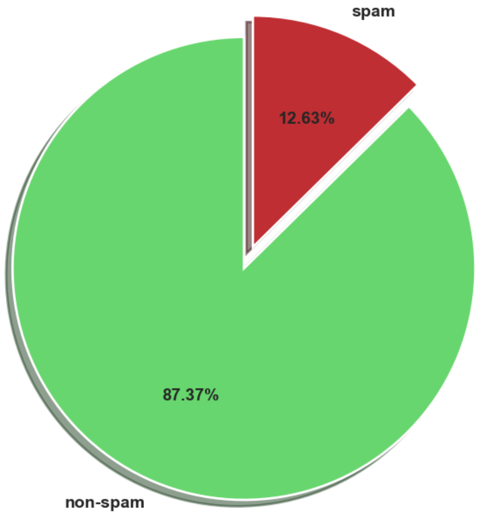

Figure 2. Dataset that contains spam and non-spam SMS

We compared the obtained results from the four algorithms used: LR, GB, SVM, and RF. The goal is to show that while some algorithms may initially outperform others, this performance can be altered by adjusting the hyperparameters. We compare the models with a manual selection of those hyperparameters and those optimized by a genetic algorithm. The comparison of models with manually selected hyperparameters versus those optimized by the genetic algorithm is presented in Table 3.

In Figure 3, we can see that some models converge earlier than others. This indicates that different models benefit from hyperparameter optimization at varying rates, reflecting the complexity and nature of each algorithm. This indicates that the best set of hyperparameters is found more quickly. This is due to several factors. For instance, the number of hyperparameters being searched in LR in Figure 3(a) is different from in GB in Figure 3(b), SVM in Figure 3(c), or RF in Figure 3(d).

Additionally, the parameters of the genetic algorithm, such as the population size in each iteration, vary depending on the size of the hyperparameter search space. After finding the optimal set of hyperparameters for each model, we use this set to train the model. Table 3 compares all models before and after using the GA (with a manual selection of hyperparameters) and after using the GA (using the optimal hyperparameters).

As observed in Table 3, the GA effectively optimized the hyperparameters for each model, resulting in improved performance. These improvements demonstrate the effectiveness of GA in fine-tuning machine-learning models for enhanced performance in SMS spam classification. Specifically:

– LR: the combined score decreased from 0.1762 to 0.1093, representing an improvement of 37.97%. The F1-score and recall significantly improved, indicating a better balance between precision and recall.

– GB: the combined score decreased from 0.1978 to 0.1013, representing an improvement of 48.79%. The precision score remained high, with notable improvements in the F1 score and recall.
– SVM: the combined score decreased from 0.0942 to 0.0864, representing an improvement of 8.28%. Both the F1 score and recall score showed enhancements, demonstrating overall improved performance.
– RF: the combined score decreased from 0.0983 to 0.0864, representing an improvement of 12.11%. This model showed consistent improvements across all metrics.

Table 3. Comparison of models before and after hyperparameter optimization using a genetic algorithm

Metric	LR		GB		SVM		RF	
	Before	After	Before	After	Before	After	Before	After
F1-score	0.8085	0.8889	0.7826	0.8973	0.9027	0.9112	0.8976	0.9105
Precision score	0.9793	0.9431	0.9782	0.9440	0.9747	0.9752	0.9827	0.9832
Recall score	0.6884	0.8406	0.6521	0.8551	0.8405	0.8551	0.8261	0.8478
Combined	0.1762	0.1093	0.1978	0.1013	0.0942	0.0864	0.0983	0.0864
Improvement(%)	-	37.97	-	48.79	-	8.28	-	12.11

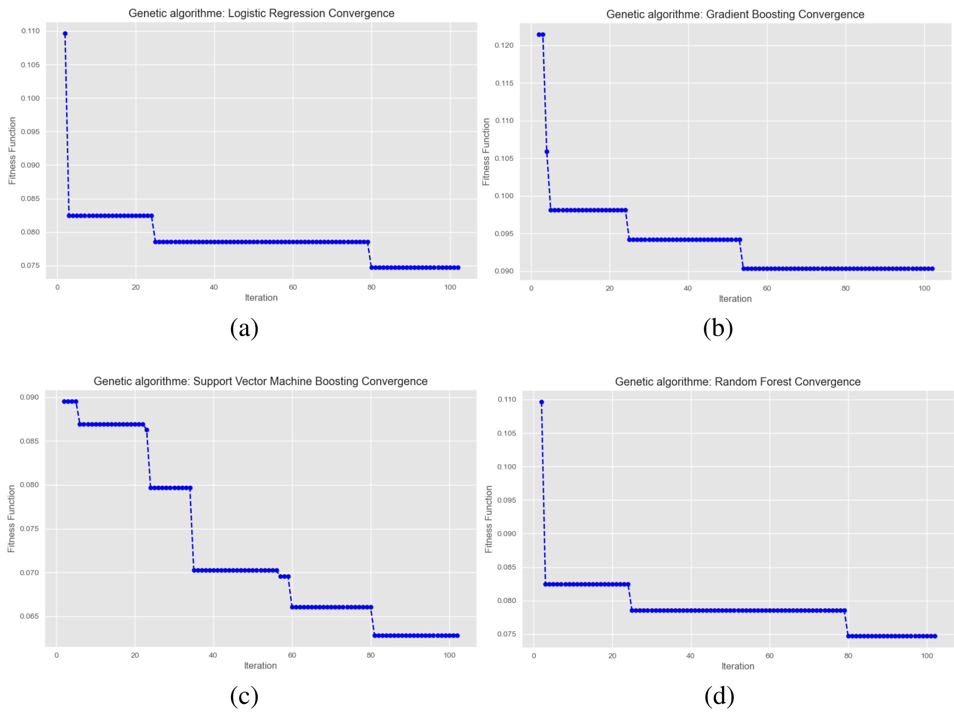

Figure 3. Genetic algorithm convergence: (a) logistic regression, (b) gradient boosting, (c) support vector machine, and (d) random forest

Our findings align with previous studies, showing that hyperparameter optimization can significantly enhance model performance. However, the extent of improvement varies depending on the initial setup and complexity of the models. To further illustrate the impact of our approach, Table 4 compares the performance metrics of our optimized models against the state-of-the-art results. Our models consistently outperform the state-of-the-art across most metrics, demonstrating the effectiveness of genetic algorithm optimization. For example, the precision score for our optimized SVM model improved to 0.9752 from the state-of-the-art 0.9211, highlighting the impact of hyperparameter tuning.

We employed a GA to optimize the hyperparameters of these models, which significantly improved their performance. The comparison before and after hyperparameter optimization demonstrated the substantial impact of this tuning process, highlighting that even models with initially lower performance can achieve competitive results with the right set of hyperparameters. The final models showed promising accuracy in classifying SMS as spam or non-spam, with improvements across multiple metrics such as F1-score, precision, recall, and a combined score. This underscores the potential of machine learning in effectively addressing the issue of spam detection. Our findings suggest that machine learning, combined with rigorous hyperparameter optimization, can be an effective tool for accurately classifying SMS messages and mitigating the impact of spam. Despite the promising results, the study is limited by the dataset's imbalance and the computational resources required for extensive hyperparameter optimization.

Table 4. Comparison of another method with manually selected hyperparameters and our method

Study	Model	Other methods			Our		
		F1	Precision	Recall	F1	Precision	Recall
[31]	SVM	0.9204	0.9211	-	0.9112	**0.9752**	-
[32]	LR	-	0.8333	-	**-**	**0.9431**	**-**
[31]	GB	0.8921	0.8923	0.8923	**0.8973**	**0.9440**	0.8551
[32]	RF	0.9091	0.8333	-	**0.9105**	**0.9832**	-

6. CONCLUSION

Our research demonstrated that hyperparameter optimization using a genetic algorithm significantly improves the performance of machine-learning models for SMS spam classification. These findings highlight the importance of hyperparameter tuning, showing that even models with initially lower performance can achieve competitive results. This work underscores the potential of machine learning in enhancing spam detection systems. The implications of our findings extend beyond SMS spam classification; they suggest that rigorous hyperparameter optimization can substantially enhance model performance in various machine-learning applications. Future research could explore the real-time deployment of these optimized models and their scalability across different languages and regions.

REFERENCES

[1] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: a review of classification techniques," *Informatica (Ljubljana)*, vol. 160, no. 1, pp. 3–24, 2007.

[2] M. Raza, N. D. Jayasinghe, and M. M. A. Muslam, "A comprehensive review on email spam classification using machine learning algorithms," *International Conference on Information Networking*, vol. 2021-January, pp. 327–332, 2021, doi: 10.1109/ICOIN50884.2021.9334020.

[3] D. G. Kleinbaum, K. Dietz, M. Gai, and M. Klein, *Logistic regression*. Springer, 2002.

[4] W. Noble, "What is a support vector machine?," *Nature biotechnology*, vol. 24, pp. 1565–1567, 2006.

[5] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *Stata Journal*, vol. 20, no. 1, pp. 3–29, 2020, doi: 10.1177/1536867X20909688.

[6] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in Neurorobotics*, vol. 7, no. DEC, p. 21, 2013, doi: 10.3389/fnbot.2013.00021.

[7] D. B. Fogel, "An overview of evolutionary programming," in *Evolutionary algorithms*, Springer, 1999, pp. 89–109.

[8] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia: case studies on organization and retrieval*, Springer, 2008, pp. 21–49.

[9] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.

[10] A. Morales-Hernández, I. Van Nieuwenhuyse, and S. Rojas Gonzalez, "A survey on multi-objective hyperparameter optimization algorithms for machine learning," *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8043–8093, 2023, doi: 10.1007/s10462-022-10359-2.

[11] L. Zhang *et al.*, "A review of machine learning in building load prediction," *Applied Energy*, vol. 285, p. 116452, 2021, doi: 10.1016/j.apenergy.2021.116452.

[12] F.Y. Osisanwo, J.E.T. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology*, vol. 48, no. 3, pp. 128–129, 2017, [Online]. Available: http://www.ijcttjournal.org.

[13] Z. Khoudi, M. Nachaoui, and S. Lyaqini, "Finding the contextual impacts on students' mathematical performance using a machine learning-based approach," *Infocommunications journal*, vol. 16, no. Special Issue, pp. 12–21, 2024, doi: 10.36244/ICJ.2024.5.2.

[14] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, 2018, doi: 10.1016/j.dcan.2017.10.002.

[15] Z. Khoudi, M. Nachaoui, and S. Lyaqini, "Identifying the contextual factors related to the reading performance of Moroccan fourth-grade students from a machine learning-based approach.," *Education and Information Technologies*, vol. 29, no. 3, pp. 3047–3073, 2024, doi: 10.1007/s10639-023-11881-8.

[16] M. Dhingra, D. Dhabliya, M. K. Dubey, A. Gupta, and D. H. Reddy, "A review on comparison of machine learning algorithms for text classification," in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, Dec. 2022, pp. 1818–1823, doi: 10.1109/IC3I56241.2022.10072502.

[17] N. Kumar, S. Sonowal, and Nishant, "Email spam detection using machine learning algorithms," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Jul. 2020, pp. 108–113, doi: 10.1109/ICIRCA48905.2020.9183098.

[18] S. Gadde, A. Lakshmanarao, and S. Satyanarayana, "SMS spam detection using machine learning and deep learning techniques," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2021, pp. 358–362, doi: 10.1109/ICACCS51430.2021.9441783.

[19] N. Ahmed, R. Amin, H. Aldabbas, D. Koundal, B. Alouffi, and T. Shah, "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: 10.1155/2022/1862888.

[20] L. GuangJun, S. Nazir, H. U. Khan, and A. U. Haq, "Spam detection approach for secure mobile message communication using machine learning algorithms," *Security and Communication Networks*, vol. 2020, pp. 1–6, Jul. 2020, doi: 10.1155/2020/8873639.

[21] S. Nandhini and J. Marseline K.S., "Performance evaluation of machine learning algorithms for email spam detection," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Feb. 2020, pp. 1–4, doi: 10.1109/ic-ETITE47903.2020.312.

[22] B. K. Dedeturk and B. Akay, "Spam filtering using a logistic regression model trained by an artificial bee colony algorithm," *Applied Soft Computing*, vol. 91, p. 106229, Jun. 2020, doi: 10.1016/j.asoc.2020.106229.

[23] S. A. Khamis, C. F. M. Foozy, M. F. A. Aziz, and N. Rahim, "Header based email spam detection framework using support vector machine (SVM) technique," in *Advances in Intelligent Systems and Computing*, vol. 978 AISC, Springer, 2020, pp. 57–65.

[24] S. B. S. Ahmad, M. Rafie, and S. M. Ghorabie, "Spam detection on Twitter using a support vector machine and users' features by identifying their interactions," *Multimedia Tools and Applications*, vol. 80, no. 8, pp. 11583–11605, Mar. 2021, doi: 10.1007/s11042-020-10405-7.

[25] I. B. Mustapha, S. Hasan, S. O. Olatunji, S. M. Shamsuddin, and A. Kazeem, "Effective email spam detection system using extreme gradient boosting," *arXiv preprint arXiv:2012.14430*, 2020, [Online]. Available: http://arxiv.org/abs/2012.14430.

[26] K. K. Devi and G. A. S. Kumar, "Stochastic gradient boosting model for twitter spam detection," *Computer Systems Science and Engineering*, vol. 41, no. 2, pp. 849–859, 2022, doi: 10.32604/csse.2022.020836.

[27] U. Srinivasarao and A. Sharaff, "SMS sentiment classification using an evolutionary optimization based fuzzy recurrent neural network," *Multimedia Tools and Applications*, vol. 82, no. 27, pp. 42207–42238, 2023, doi: 10.1007/s11042-023-15206-2.

[28] G. Waja, G. Patil, C. Mehta, and S. Patil, "How AI can be used for governance of messaging services: a study on spam classification leveraging multi-channel convolutional neural network," *International Journal of Information Management Data Insights*, vol. 3, no. 1, p. 100147, 2023, doi: 10.1016/j.jjimei.2022.100147.

[29] V. P. Sumathi, V. Vanitha, R. Kalaiselvi, and T. T. Tania, "Performance comparison of machine learning algorithms in short message service spam classification," in *2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation, ICAECA 2023*, Jun. 2023, pp. 1–4, doi: 10.1109/ICAECA56562.2023.10199265.

[30] K. Sastry, D. Goldberg, and G. Kendall, "Genetic algorithms," *Search methodologies: Introductory tutorials in optimization and decision support techniques*, pp. 97–125, 2005, doi: 10.1007/0-387-28356-0_4.

[31] S. Rao, A. K. Verma, and T. Bhatia, "Hybrid ensemble framework with self-attention mechanism for social spam detection on imbalanced data," *Expert Systems with Applications*, vol. 217, p. 119594, 2023, doi: 10.1016/j.eswa.2023.119594.

[32] N. Jalal, A. Mehmood, G. S. Choi, and I. Ashraf, "A novel improved random forest for text classification using feature ranking and optimal number of trees," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2733–2742, 2022, doi: 10.1016/j.jksuci.2022.03.012.

BIOGRAPHIES OF AUTHORS

Nasereddine Hafidi received a Master's degree in Business Intelligence from Sultan Moulay Slimane University. Currently, he is pursuing a Ph.D. in Computer Science at the same institution. His research interests encompass a wide range of topics within the realms of inverse problems, machine learning, and data sciences. He is particularly focused on exploring innovative methodologies and applications in these fields, aiming to contribute significantly to both academic research and practical implementations. He can be contacted at email: nasereddine.hafidi@usms.ma.

Zakaria Khoudi received a Master's degree in Information and Communication Technology from Abdelmalek Essaadi University. Currently, he is pursuing a Ph.D. in Computer Science at Sultan Moulay Slimane University. His research interests are diverse and focus on data science, educational data mining, learning analytics, and machine learning applications. He aims to advance the field through innovative research and practical solutions that enhance the understanding and application of these technologies in educational settings and beyond. He can be contacted at email: zakaria.khoudi@usms.ma.

Mourad Nachaoui received his Ph.D. from Nantes University in 2011. He is currently a professor at Sultan Moulay Slimane University. From 2013 to 2019, he served as a guest professor at the Laboratory of Mathematics Jean Leray at Nantes University. His primary research interests lie in the fields of applied mathematics, inverse problems, data sciences, and computer sciences. Throughout his career, he has made significant contributions to these areas, advancing both theoretical understanding and practical applications. He can be contacted at email: nachaoui@gmail.com.

Soufiane Lyaqini received his Ph.D. from Sultan Moulay Slimane University. He is currently an associate professor at Hassan 1 University. His research interests are focused on machine learning, data mining, and optimization. Throughout his academic career, he has dedicated himself to advancing these fields, contributing to both the theoretical foundations and practical applications. His work aims to solve complex problems and improve methodologies within these domains, impacting a wide range of industries and academic disciplines. He can be contacted at email: soufiane.lyaqini@uhp.ac.ma.