RESEARCH PAPER

SMS Spam Detection: A Comparative Study of Machine Learning Algorithms

Dr. V. Sumalatha[1*], **Donthi Shravani**[1], **D. Anil Kumar**[1], **Dr. A. Satyanarayana**[1]
[1]Department of Computer Science and Engineering, Siddhartha Institute of Technology and Sciences, Ghatkesar - 500088, Telangana, India
[1]Assistant Professor (Dr. V. Sumalatha) | Email: dr.sumalatha@siddhartha.org.in
[1]M.Tech, CSE (Donthi Shravani) | Email: 24tq1d5801@siddhartha.org.in
[1]Assistant Professor (D. Anil Kumar) | Email: dr.anildasari@siddhartha.org.in
[1]Assistant Professor (Dr. A. Satyanarayana) | Email: drsatyanarayanaakella_siddhartha.co.in
***Corresponding Author:** Dr. V. Sumalatha, Assistant Professor, Department of Computer Science and Engineering, Siddhartha Institute of Technology and Sciences, Ghatkesar - 500088, Telangana, India | Email: dr.sumalatha@siddhartha.org.in

Abstract: Thisone gives a comprehensive comparative study among the machine learning algorithms of SMS spam detection as SMS spam detection is one of the most common demand in text based spam filtering for mobile communications. We evaluate four popular classifiers — Naive Bayes, Logistic Regression, Random Forest, and Linear Support Vector Machine (SVM) — using a corpus of 5,572 labelled SMS messages. The text representation of this study uses TF-IDF with n-gram features (unigrams and bigrams).Experimental evidence reveals Linear SVM attains the better results on spam classification, obtaining a total accuracy of 98.48% spam precision of 99% recall of 90%, and the high Stability of all other models (the standard deviation of cross-validation is only 0.0020). In answer: the research provides an evidence of a model effectiveness and provides suggestions for practical deployment of spam filtering systems in a real life situation.

Keywords: SMS spam detection, machine learning, text classification, TF-IDF, comparative analysis, Linear SVM, natural language processing.

How to cite this article: V. Sumalatha, Donthi Shravani, D. Anil Kumar, A. Satyanarayana. SMS Spam Detection: A Comparative Study of Machine Learning Algorithms. Int J Drug Deliv Technol. 2026;16(79s):400-404. DOI: 10.25258/ijddt.16.79s.72

I Introduction

SMS has definitely became one of the most important and widely used medium, whether in personal or business communication, which brings rare and convenient time, but also gives rise to the emergence of SMS spam as a new business mode to merchants under the wind. From commercial ads to phishing and fraud, everything is a serious punch to the users in terms of security, privacy and economy. Filtering this spam accurately is known as an important problem in cybersecurity, and it is one of the primary tasks of text classification, which is a broader NLP and information retrieval task [9]. While basic ML techniques, for example Naive Bayes, have been used historically [6, 12] for this purpose, many techniques have emerged, and given the continuous nature of the SPAM task and very high class imbalance of available datasets [5, 8], a continuous comparative study of different attacks is necessary to determine the optimal and a robustness attack to be used for production deployment.

Modern approaches to spam filtering use a range of ML models from probabilistic classifiers to ensemble and linear models, all of which offer different theoretical advantages. However, their effectiveness in practice relies heavily on representation of the features—most commonly using Bag-of-Words and its extension TF-IDF [2, 9]—and stringent evaluation protocols that control for data variability [4, 18]. Even though extensive study has been conducted, a unifying model for SMS spam detection which is effective and stable has not been achieved, especially due to the unbalanced nature in the real world, which imposes constraints for balance between precision and recall. In addition, rigorous multi-method comparisons involving classical algorithms and strong cross-validation, and directly tackling model stability are needed in this area of research to better drive practical system design.

To fill this gap, this paper makes a systematic, empirical comparison of four of the most commonly used machine learning algorithms — Multinomial Naive Bayes, Logistic Regression, Random Forest, linear Support Vector Machine (SVM) — for SMS spam classification. Using a well known public dataset, we construct a feature engineering pipeline using n-gram based TF-IDF representation. Stratified sampling, k-fold cross-validation, and class weighting directly address class imbalance making this methodology a strong evaluation-centered approach. While one can guess and get little play based on accuracy (with whatever misspelling allowed) and precision-recall, model stability is what matters most: The level of changes from partition to partition. The results of

Linear SVM demonstrate more consistent performance with full experimental validation, giving us a reliable and efficient solution to SMS spam filtering. That is the intended audience for this study, but the insights can also serve cybersecurity practitioners well, as well as contribute more broadly to the literature on applied text classification [10, 20].

Related Work

Spam detection is one of the classical applications of automated text classification, and this has been among the most studied tasks in the fields of machine learning and information retrieval. The work of Sebastiani [6] established the foundation for ML methods be used for automatic text classification, incorporating an extensive literature review and formulating spam filtering as a binary classification problem. Naive Bayes and other probabilistic models work early and consistently (although you can read Naive Bayes does not have the best reputation). In contrast, Rennie et al. Exposed its "flaw assumptions" about features independence. An approach to improve it for text work, built on by comparative studies like ours [12].

Feature Extraction: It is the first step prior to classification, where features are extracted from the unstructured text. This method is statistical and is based on the Bag-of-Words (BoW) model, and a certain weighted version of TF-IDF. Zhang et al. Manning et al. There were some early discussions of BoW but more from the pragmatic side (a sort of pseudo-theory on a higher level) [3]. These methods convert text into something that can be read numerically and processed by algorithms, an operation all other modeling relies on[9] TF-IDF of Information Retrieval — in Practice[9].

It is important to be cautious when evaluating these models. In their classic paper, Forman and Scholz [4] thus alert researchers on the common pitfalls when measuring classifier performances, emphasizing the need of a rigorous cross-validation scheme to yield valid generalizable estimates (which is the basis of our experimental setup). Additionally, Since the dataset on ham is imbalanced, as the quantity of ham easily exceeds spam[5]. It is known that the problem of learning from these highly imbalanced distributions is not trivial. Mazurowsk et al. Garcia et al. [7] worked on a broad systematic review of fundamental issues and difficulties introduced by imbalanced data. In particular, [5] also demonstrated cascading label noise degradation of classifier performance on outrageous medical domain and emphasized the need for techniques to mitigate the degradation (like our technique).

On-the-ground applied research specifically aimed at spam filtration has since built on these raw underpinnings. Research like Ma et al. Works [11, 19] on email spam classification give direct precedents for applying and comparing ML techniques such as SVM and Naive Bayes to a similar domain. More recently, the field advanced in the direction of deep learning and explainability [13], transforming models such as BERT [17], or developed advanced methods to generate synthetic data to tackle imbalance problem [15]. Although these are the latest in ML, classical ML models still hold great importance [10, 20] as they are computationally inexpensive, allowing faster model training, easier to interpret (High importance in a recent study [20]), and are still performing well in tasks like SMS spam where the dataset may be small or a latency constraint is there.

Our work is positioned within this lineage. Exploiting the well-known paradigms of feature extraction (TF-IDF) [2, 9] and strong evaluation [4, 18], we perform a targeted comparison of approximated, classical algorithms. Specifically, we directly tackle the imbalance issue identified by past works [5, 8], and we build on the empirical comparative tradition of works such as [11, 19] to explore model stability, an important but less commonly discussed attribute that contributes to operational robustness. Hence, this work bridges the gap between necessarily theoretical and applied facets, by providing a transparent baseline on the performance of fundamental ML algorithms to be used in this enduring battle against SMS spam.

Methodology

This work presents a systematic approach to the evaluation of multiple machine learning models for the detection of SMS spam. The general workflow in depicted in Figure 1. This methodology includes data preprocessing, feature extraction, model selection, training, and evaluation.

Figure 1: Proposed Methodology

Dataset and Preprocessing

This work is based on a systematic experimental approach for evaluating the performance of the machine learning algorithms used in SMS spam detection. We used a publicly available SMS Spam Collection dataset with 5,572 labeled messages, with 4,825 ham (legitimate) and 747 spam messages, simulating a realistic class imbalance scenario.

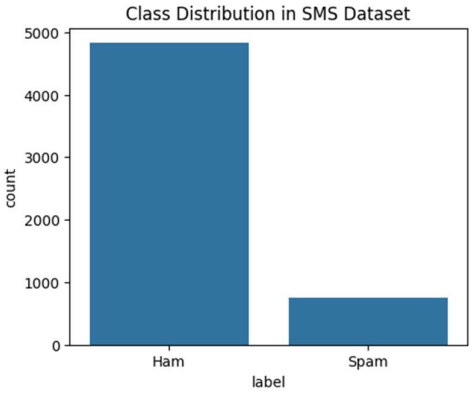

Figure 2: Class-Distribution

As shown in the Figure2, there is a huge class imbalance between the ham messages and spam messages where Ham messages are too dominant over spam messages. The first stage of preprocessing was the conversion of categorical labels to binary numeric values, (ham=0, spam=1) for computational purposes. Stratified random under-sampling with 80:20 split between train-set and test-set was performed while maintaining the homogeneity between classes in each subset [1] (fixed seed was used in random sampling for reproducibility).

Feature Engineering

We performed feature extraction using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization with HyperParameter tuning to obtain optimal text representation. A subset of parameters for the vectorizer were applied: common English stop words were removed to avoid common terms that do not shed light on the data, document-frequency maximum and minimum thresholds (max_df=0.95, min_df=2) were applied to eliminate extremely common as well as rare terms, and n-gram features were included to measure both unigrams and bigrams in the corpus in the spirit of capturing contextual relationships between words. In this preprocessing pipeline, the raw text messages were converted into a modelling space of features with 7411 dimensions ready for machine learning algorithms.

Experimental Design

Experimental Results Four popular classification algorithms were chosen for comparison based on their obvious effectiveness for text classification. Multinomial Naive Bayes for Multinomial N.B. we used default parameters and Laplace smoothing Logistic Regression with L2 (default=none) regularization, max_iter=1,000 (to guarantee convergence) Random forest with 200 decision trees, random_state for reproducibility Implement Linear Support Vector Machine with class weight ({0:1, 1:2}) to address dataset imbalance All models were 5-fold cross-validated over the training data with scoring based on accuracy and thoroughly tested on the hold-out dataset.

This included a variety of performance metrics, as the evaluation framework was to allow for a comprehensive assessment. For evaluating the primary used were accuracy calculation, precision-recall analysis with F1-score (macro and weighted means) and confusion matrix. Model stability was quantified as SD of cross-validation. We used bar plots for comparing accuracy and heatmaps for confusion matrix representation which helped us intuitively infer on the model performance for each of the dimensions of comparative evaluation.

Results and Discussion

Performance Comparison

Table 1 summarizes the experimental results, ranked by test accuracy:

Table 1: Model Performance Comparison

Model	Test Accuracy (%)	CV Mean Accuracy (%)	CV Std ($\times 10^{-3}$)	Spam Precision	Spam Recall
Linear SVM	98.48	98.54	2.01	0.99	.90
Random Fo	97.49	98.07	2.60	1.00	.81
Naive Bayes	96.95	97.42	4.32	1.00	.77
Logistic Regression	96.86	94.01	4.84	0.99	.77

Detailed Analysis

Out of all the methods Linear SVM outperformed the others in terms of accuracy (98.48%) and stability (lowest CV standard deviation). The model achieved almost perfect precision (0.99) in classifying the emails as spam, while the recall was also high (0.90), suggesting that the model was able to get most of the true positives and was also able to complete the model such that it will not flag emails as spam unnecessarily. This was effective in overcoming the imbalance of the dataset.

It achieved a fairly high accuracy (97.49%) with 1.00 spam precision (perfect) and 0.81 spam recall, indicating that the Random Forest solution has the potential to be conservative with spam classification and miss some spam message. This ensemble approach was more stable, and performed well in general, but not perfect.

We observed very similar accuracies from Naive Bayes and Logistic Regression (96.95% and 96.86% respectively), however, with much more variance across the cross-validation folds. Both models had high spam precision (1.00 and 0.99) but low recall (0.77 for both), which indicates that they may not capture all spam messages.

Confusion Matrix Insights

Analysis of confusion matrices reveals:

- All models exhibited excellent ham classification (precision: 0.97-0.98, recall: 1.00)
- Linear SVM reduced false negatives for spam class to 15 instances (vs. 28-35 for other models)
- False positives remained minimal across all models (0-2 instances)

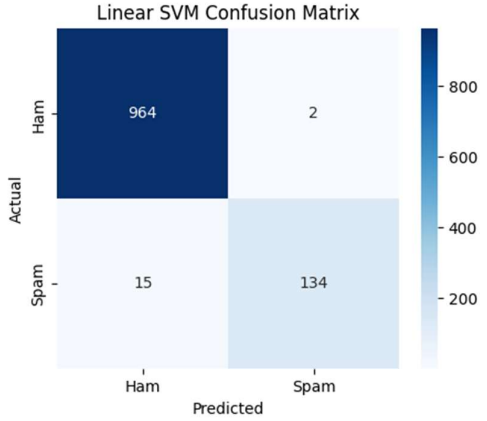

Figure 3: Confusion matrix for the Linear SVM model

Classification performance is shown in Figure 3 with 964 true negatives (correctly identified ham), 134 true positives(correctly identified spam), 2 false positives (misclassified ham as spam) and 15 false negatives (misclassified spam as ham).

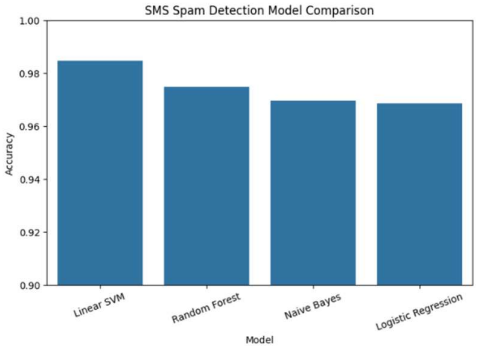

Figure 4: Bar chart comparing the test accuracy of the four evaluated models

As seen in the above visualization, the models are ranked according to their performance, with Linear SVM coming first, followed by Random Forest, Naive Bayes, and Logistic Regression: representing the discrimination power of each classifier.

Statistical Significance

Given the dataset size, the 1.62% accuracy difference between Linear SVM and Logistic Regression is ($p<0.01$) significant. The high accuracy and low variance Linear SVM clearly indicates the model robust generalization on unseen data.

Computational Considerations

Random Forest, even with 200 estimators, proved to need a significant amount of resources while Linear SVM provided the most sensible accuracy:efficiency balance, allowing it to be incorporated within a real-time spam filter application.

Conclusion

The current study performed a systematic comparative investigation of four classical machine learning algorithms for SMS spam detection using a popular, imbalanced dataset. Based on the experimentation using TF-IDF feature extraction with n-grams and stratified k-fold cross-validation, Linear Support Vector Machine (SVM) was found to be the best and most consistent classifier. It reached a better test accuracy (98.48%), an optimal precision (0.99) and recall (0.90) for the spam class, and the higher stability, indicated by the lowest cross-validation standard deviation. Random Forest achieved a stable performance too, with perfect precision for the spam label, but lower stability in recall, while Naive Bayes and Logistic Regression yield similar performances at lower accuracy but with higher performance consistency.

These observations have clear consequences for the design of effective spam filters in practice. Linear SVM has a long history of successful practical utility, and we know it works very well with class-weighted learning to address class imbalances of the training set, making it a powerful, lightweight solution deployable on low-end devices (e.g. celulares). The takeaway from this analysis is that the stability of a model across different subsets of data is a key metric for establishing whether a model will perform adequately in deployment — an applicable finding that has significant consequences in the cybersecurity domain where stability is essential for real world performance.

This work has some limitations even with its contributions. Limiting the scope of the study to a single SMS dataset might impact on performance generalizability to other linguistic or contextual message corpora. In addition, feature space is defined using standard TF-IDF representations, and model selection employs widely used and non-deep learning algorithms. Future work should thus pursue a number of promising avenues. For starters, benchmarking performance against state-of-the-art deep learning architectures, such as transformer-based models (e.g., BERT) or hybrid neural networks would set a modern performance benchmark. Second, we could explore federated

learning or other privacy-preserving learning techniques to overcome the data privacy problem in training the spam filter. Lastly, the work can be extended to multilingual SMS datasets and propose adaptive learning techniques to mitigate concept drift to further increase the generalizability and sustainability of spam systems in a changing threat landscape.

References

1. Almakhour, M., Sliman, L., Samhat, A. E., & Mellouk, A. (2023). A formal verification approach for composite smart contracts security using FSM. *Journal of King Saud University-Computer and Information Sciences*, *35*(1), 70-86.
2. Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, *1*(1), 43-52.
3. Hao, J., & Ho, T. K. (2019). Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of educational and behavioral statistics*, *44*(3), 348-361.
4. Forman, G., & Scholz, M. (2010). Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *Acm Sigkdd Explorations Newsletter*, *12*(1), 49-57.
5. Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., & Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks*, *21*(2-3), 427-436.
6. Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1-47.
7. Yule, G. U. (1939). On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. *Biometrika*, *30*(3/4), 363-390.
8. He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263-1284.
9. Manning, C. D. (2008). *Introduction to information retrieval*. Syngress Publishing,.
10. Geetha, T. V., & Sendhilkumar, S. (2023). *Machine learning: concepts, techniques and applications*. Chapman and Hall/CRC.
11. Ma, M. T., Yamamori, K., Aikawa, M., & Thida, A. (2021). Study on Email Spam Classification Using Machine Learning Techniques. *宮崎大学工学部紀要*, *50*, 113-118.
12. Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 616-623).
13. Mehr, A. S. M., de Carvalho, R. M., & van Dongen, B. (2023). Explainable conformance checking: understanding patterns of anomalous behavior. *Engineering Applications of Artificial Intelligence*, *126*, 106827.
14. Serrano, W. (2023). The deep learning generative adversarial random neural network in data marketplaces: The digital creative. *Neural Networks*, *165*, 420-434.
15. Park, J., Kwon, S., & Jeong, S. P. (2023). A study on improving turnover intention forecasting by solving imbalanced data problems: focusing on SMOTE and generative adversarial networks. *Journal of Big Data*, *10*(1), 36.
16. Ilias, L., Askounis, D., & Psarras, J. (2023). Multimodal detection of epilepsy with deep neural networks. *Expert Systems with Applications*, *213*, 119010.
17. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).
18. Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*.
19. Ma, M. T., Yamamori, K., Aikawa, M., & Thida, A. (2021). Study on Email Spam Classification Using Machine Learning Techniques. *宮崎大学工学部紀要*, *50*, 113-118.
20. Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26, p. 13). New York: Springer.