VOTING ENSEMBLES IN SPAM CLASSIFICATION: AN APPROACH FOR OPTIMIZING MODEL SELECTION

Charles Lamb [1,†,*]

[1] Northwestern University School of Professional Studies

Masters of Data Science Program

633 Clark St, Evanston, IL 60208

†Address to which correspondence should be addressed:

CGLamb@outlook.com

*LinkedIn Profile: https://www.linkedin.com/in/charles-g-lamb/

Release Date: 2023-08-20

Abstract

This paper proposes a voting ensemble technique for spam classification. While previous authors have proposed voting ensemble methods for spam filtering, this research offers an alternative criteria for determining which underlying methods should be included in the ensemble. The suggested approach relies on a training, validation, and test data design, whereby the validation dataset is used to evaluate an optimal ensemble. The voting ensemble considered here is a three-method model with hard voting and equal weighting. Eleven different underlying methods are considered for inclusion in the ensemble. These underlying methods range from older naïve Bayes methods to modern Bidirectional Encoder Representations from Transformers. All possible three-method ensembles using these eleven underlying methods are evaluated against the validation dataset, and the resulting ensemble with the highest F-score is selected as optimal. For the dataset examined in this paper, the resulting ensemble is found to outperform the ensemble design proposed by previous authors.

Keywords: spam classification, voting ensemble, BERT (Bidirectional Encoder Representations from Transformers), support vector machine, random forests, extremely randomized trees, neural networks, natural language processing

1. Introduction

This paper proposes a voting ensemble method for spam classification. Ensemble methods are meta-methods, combining predictions from multiple standalone methods into a single prediction. Voting methods are a type of ensemble method that combines the predictions of the standalone methods into a single prediction by giving each member votes. The usage of voting ensembles in the problem of spam classification is not new and has been previously proposed by Singh et al. (2022). This paper extends their work proposing an alternative criteria for determining which models to include in the ensemble. Their approach based inclusion in the ensemble on best accuracy of the standalone methods. The approach presented here exhaustively tests ensembles against a validation dataset and selects the optimal ensemble based on best fit against the validation data.

Spam is defined as unsolicited electronic communications and can occur across almost any digital communication mode, including: email, short message services (SMS), and direct messaging (Kosta, Valcke, and Stevens 2009). In the modern world, spam is a significant problem both in terms of its volume and the amount of distraction it causes. Spam is estimated to account for 87% of global email traffic (equating to billions of spam messages sent daily) and $20.5 billion annually in estimated financial burden on companies and the recipients of spam (MAAWG 2011, Rao and Reiley 2012). Furthermore recent advances in generative AI threatens to expound the spam problem and could lead to even greater spam volumes and spam that is more difficult to differentiate from non-spam (Stokel-Walker and Noorden 2023). Given the volume of spam circulating within global communication infrastructure, even modest incremental improvements in spam detection techniques can yield significant results. The need for improved spam detection motivates the research of this paper.

2. Literature Review

Spam classification techniques have developed rapidly over the last twenty years. The most common spam detection approach used in early commercial spam software was naïve Bayes filtering (Process Software, n.d.). Despite its age, naïve Bayes has the advantages of needing less training data than most modern methods and being very scalable (Dada et al. 2019). While modern methods generally offer greater predictive power, naïve Bayes is often used in modern literature as a baseline against which to measure other classification methods. Hidalgo et al. (2006), Shahariar et al. (2019), Wu, Shi, and Ma (2021), and Singh et al. (2022) all use naïve Bayes models as comparison models against which to measure newer classification models.

Other authors have applied neural network designs to the problem of spam detection. Shahariar et al. (2019) used multi-layer perceptron (MLP), convolutional neural networks (CNN), and long short term memory networks (LSTM) to detect truthful and deceptive Yelp reviews and found these deep learning algorithms achieved greater accuracy than traditional machine learning approaches in situations where a large amount of training data was available. Wu, Shi, and Ma (2021) combined a CNN with a LSTM classification model. Their model used a CNN layer to extract features from the data and a LSTM layer to allow for sequence prediction. In their testing, this hybrid model outperformed standalone CNN and LSTM models, as well as six other types of common machine learning techniques.

Support vector machines (SVMs) are also commonly used in the spam detection literature and valued for their ability to handle high dimensionality over small datasets (Dada et al. 2019). Hidalgo et al. (2006) proposed a SVM model for spam detection on both English and Spanish SMS databases and preferred the SVM model over both a naïve Bayes and decision tree model. Shahariar et al. (2019) also applied a SVM model and found that it outperformed both a CNN and MLP, but ultimately underperformed their preferred LSTM model.

Several authors have also used decision tree methods in spam filtering, Omotehinwa and Oyewola (2023) applied decision tree methods to email spam data and showed promising results using both random forest and gradient-boosted trees. Similarly, Abdulrahman and Salim (2022) used decision tree techniques to successfully detect spam in an email dataset written in Malay.

Newer, state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) methods have been applied to spam classification by Oswald, Simon, and Bhattacharya (2022). They were able to use BERT models to improve contextual understanding and boost predictive power over other classification models. This increase in contextual understanding is the result of the pre-trained nature of BERT models. The model was pre-trained on a large corpus unrelated to spam classification before being trained/tuned on the spam corpus. This pre-training imparted a deeper understanding of words and phrases than a model which was only trained on the spam corpus. This deeper understanding allowed the BERT model to better extract intent from the spam corpus.

While new methods like BERT have been helpful in improving spam filtering, recent advances in large language models (LLMs) offer a powerful new tool to spammers. Licato (2023) discusses the advances LLMs have made recently and their ability to be used to generate both more spam and spam that is more difficult to differentiate from ham (non-spam). He also discusses how LLMs could be used to facilitate back-and-forth correspondence with someone being spammed, which would further confound the authenticity problem. The way in which LLMs will shape the spam landscape remains an open-ended question, but much work will have to be done to develop new techniques and technologies to help deter their potential negative potential for spam generation.

A number of the authors mentioned above focus their research on comparing and contrasting the performance of various methods, incluing the research of: Hidalgo et al. (2006), Silva, Yamakami, and Almeida (2012), Shahariar et al. (2019), Wu, Shi, and Ma (2021), Abdulrahman and Salim (2022), and

Omotehinwa and Oyewola (2023). The abundance of methods and the relative advantages and disadvantages of each method on various corpuses led Singh et al. (2022) to propose an ensemble technique, whereby a number of standalone methods were evaluated based on highest accuracy and then combined to produce a single prediction of spam/ham. Their voting ensemble is presented as a method in this paper and serves as a baseline against which to compare the alternative approach suggested by this paper.

3. Data

The models in this paper are trained on labelled data comprised of 5,169 unique SMS messages. The dataset was generated by Almeida and Hidalgo (2012) and is an amalgamation of four separate datasets. The documents in the corpus are English text, but contain a number of misspellings, shorthands, and slangs commonly found in mobile text messages. 653 of the 5,169 documents or approximately 13% of the corpus is spam. The dataset has also been reproduced on Kaggle (n.d.) and the interested reader is pointed there for a wealth of relevant coding help.

Standard preprocessing procedures were performed on the data prior to model training, including: converting text to lower case, removing stop words, removing punctuation, and stemming. As the data contained a significant number of tokens common amongst phone text messages, additional procedures were applied to remove non-alphanumeric characters (primarily emojis). Additionally, uniform resource locators (URLs) were replaced with the word “URL”. This was done as the presence of a URL (now represented by the keyword “URL”) likely provides value in spam prediction, while the specifics of the URL are deemed irrelevant.

Both a count vectorization and term frequency-inverse document frequency (tf-idf) were applied to the data. All methods were run once using each vectorization, and resulting F-scores were compared. Performance was evenly split with half of the methods producing a better F-score using the

tf-idf and the rest producing a better F-score using count vectorization. Since there was little difference in model performance given the different vectorizations, tf-idf vectorization is used throughout the remainder of this paper for simplification and to avoid presenting the same methods multiple times.

A train, validation, and test experiment design was used, with 50% of the data allocated to the training dataset, 25% to the validation dataset, and 25% to the test dataset. The inclusion of a validation layer was primarily used in order to facilitate ensemble optimization. The individual models were all trained on the training dataset, but the optimal voting ensemble was selected based on fit against the validation dataset.

Two different approaches were taken to address data imbalance. Class weight balancing was used in methods that allowed for it. For methods that do not allow class weighting, an under sampling pipeline was built that under sampled from the majority class (ham class).

4. Methods

The following classification methods were trained and hyper tuned: logistic regression, naïve Bayes, SVM, random forest, gradient boosted trees, extremely randomized trees, adaptive boosting, random forest with extreme gradient boosting (RF with extreme GB), CNN, LSTM, and BERT.

4.1. Singh Ensemble Method

Additionally, based on the work of Singh et al. (2022) a voting ensemble method was also added. This method is described as the “Singh Voting Ensemble” throughout the remainder of this paper. The Singh Voting Ensemble is a meta-method combining three methods using a voting approach where each member is given equal voting power. No probabilistic predictions were passed to the ensemble; any probabilistic predictions were converted to binary before passing to the ensemble (i.e. hard voting was used). The ensemble prediction is taken to be the class predicted by the majority of the

methods. The three methods included in the Singh Voting Ensemble were the 3 methods with the best F-score on the validation dataset. It should be noted that Singh et al. (2022) originally used accuracy for the selection criteria. However as this paper will be evaluating model fit based on F-score instead of accuracy, F-score was used instead. The change in metrics has no impact on the ensemble as the three models with the highest accuracy on the validation dataset also were the models with the three highest F-scores.

4.2. Alternative Voting Ensemble

In addition to the Singh Voting Ensemble, an alternative voting ensemble is proposed based on the criteria for model selection proposed in this paper. This alternative ensemble replicates the Singh Voting Ensemble approaches in all aspects other than criteria for model inclusion. Whereas the Singh ensemble selects models for inclusion based on highest F-score of the underlying models, this method evaluates all possible three-method ensembles and chooses the voting ensemble with the best F-score against the validation dataset as the optimal method.

5. Results

Observational error metrics (accuracy, precision, recall, and F-score) for each method are shown below in Figure A. The metrics are based on fit against the test dataset.

Figure A.

Method	Accuracy	Precision	Recall	F-score
Random Forest	97.9%	96.6%	86.4%	91.2%
Alternative Voting Ensemble	97.8%	97.2%	85.2%	90.8%
Singh Voting Ensemble	97.8%	96.5%	85.2%	90.5%
BERT	97.5%	91.7%	88.3%	89.9%
Extremely Randomized Trees	97.5%	97.1%	82.7%	89.3%
Gradient Boosted Trees	97.4%	96.4%	82.7%	89.0%
RF with Extreme GB	96.8%	86.6%	87.7%	87.1%
Naïve Bayes	96.9%	91.2%	83.3%	87.1%
SVM	96.8%	92.9%	80.2%	86.1%
CNN	96.7%	92.8%	79.6%	85.7%
LTSM	96.4%	87.2%	84.0%	85.5%
Adaptive Boosting	96.4%	90.3%	80.2%	85.0%
Logistic Regression	96.1%	85.0%	84.0%	84.5%

The three methods included in the Singh Voting Ensemble are the CNN, gradient boosted trees, and random forest. The three methods included in the alternative voting ensemble were the CNN, gradient boosted trees, and BERT.

6. Analysis and Interpretation

The alternative ensemble method proposed by this paper generated a slightly better set of predictions than the Singh Voting Ensemble, as evident by the improvement in precision and F-score (and tie based on accuracy and recall). While the outperformance of the alternative ensemble was slight, these experimental results show that ensemble hyper tuning can produce better results than simply selecting models for inclusion based on best standalone performance.

It should be noted that the random forest method did outperform the voting ensemble methods, generating a slightly better F-score. Given the design of this experiment called for a train/validation/test split to allow for a validation layer to hyper tune the voting ensemble, the random forest outperforming the voting ensembles suggests that a more practical overall approach to the spam classification problem associated with this dataset may have been to forgo a voting ensemble method entirely and simply use a train/test split that allowed for a larger training dataset.

7. Conclusions

Spam is prevalent in everyday life. The ability to identify and filter out spam is critical to ensure the usability of modern communication platforms. A significant amount of work has been done to apply various techniques to the problem of identifying spam. In addition to applying state-of-the-art techniques like BERT, voting ensemble techniques have been previously proposed that combine the predictions of multiple individual methods into a single prediction in an effort to boost overall predictive power. This research finds that previously proposed voting ensembles can be improved upon by using a

different criteria for underlying model inclusion. The technique proposed in this paper relies on an experimental design that uses a validation dataset to select the optimal ensemble.

8. Direction for Future Work

This paper maintained a relatively simple voting ensemble design that considered only three-method ensembles and imposed hard voting and equal weighting. Future research should explore alternative ensemble approaches, such as: weighted voting, stacking approaches, and bagging. This research could determine if other ensemble techniques can generate an ensemble that outperforms all standalone methods.

9. Code

The code used to conduct this research is available as a Jupyter Notebook at the following public Github repository: https://github.com/cglamb/Voting_Ensemble_Spam.

A. Appendix

This exhibit summarizes the assumptions used to generate the voting ensemble methods described in this paper.

Properties	Singh Voting Ensemble	Alternative Voting Ensemble
Number of Models in Ensemble:	3	3
Voting Weight:	Equal Weight	Equal Weight
Methods Considered for Inclusion:	All	All
Inclusion Criteria for Methods:	Best F-Score	All combinations considered, ensemble with best F-score on validation data selected as optimal

References

Abdulrahman, Saifuldeen and Mohammad Salim. 2022. "Using Decision Tree Algorithms in Detecting Spam Emails Written in Malay: A Comparison Study." *ITM Web of Conferences* 42: 1001–. https://doi.org/10.1051/itmconf/20224201001.

Almeida, Tiago and Jos Hidalgo. 2012. "SMS Spam Collection." *UC Irvine Machine Learning Repository*, Accessed July 1st, 2023. https://archive.ics.uci.edu/dataset/228/sms+spam+collection.

Dada, Emmanuel, Joseph Bassi, Haruna Chiroma, Shafi'i Abdulhamid, Adebayo Adetunmbi, and Opeyemi Ajibuwa. 2019. "Machine Learning for Email Spam Filtering: Review, Approaches and Open Research Problems." *Heliyon* 5: no. 6: e01802–e01802. https://doi.org/10.1016/j.heliyon.2019.e01802.

Hidalgo, José, Guillermo Bringas, Enrique Sánz, and Francisco García. 2006. "Content Based SMS Spam Filtering." In *Proceedings of the 2006 ACM Symposium on Document Engineering* pg 107–114. https://doi.org/10.1145/1166160.1166191.

Kaggle. n.d. "SMS Spam Collection Dataset." Accessed July 1st, 2023. https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset/code.

Kosta, Eleni, Peggy Valcke, and David Stevens. 2009. "'Spam, Spam, Spam, Spam ... Lovely Spam!' Why Is Bluespam Different?" *International Review of Law, Computers & Technology* 23, no. 1-2: 89–97. https://doi.org/10.1080/13600860902742513.

Licato, John. 2023. "AI-generated Spam May Soon Be Flooding Your Inbox – And It Will Be Personalized To Be Especially Persuasive." *The Conversation*. Accessed August 10th, 2023. https://theconversation.com/ai-generated-spam-may-soon-be-flooding-your-inbox-and-it-will-be-personalized-to-be-especially-persuasive-201535.

Messaging Anti-Abuse Working Group (MAAWG). 2011. "Email Metrics Program: The Network Operator's Perspective." Accessed July 2nd, 2023. https://www.m3aawg.org/sites/default/files/document/MAAWG_2011_Q1-4_Metrics_Report15Rev.pdf.

Omotehinwa, Oluwatosin, and David Oyewola. 2023. "Hyperparameter Optimization of Ensemble Models for Spam Email Detection." *Applied Sciences* 13, no. 3: 1971–. https://doi.org/10.3390/app13031971.

Oswald, C., Sona Simon, and Arnab Bhattacharya. 2022. "SpotSpam: Intention Analysis–driven SMS Spam Detection Using BERT Embeddings." *ACM Transactions on the Web* 16, no. 3: 1–27. https://doi.org/10.1145/3538491.

Process Software. n.d. "Common Spam Filtering Techniques." Accessed July 10th, 2023. https://www.process.com/products/pmas/whitepapers/explanation_filter_techniques.pdf.

Roa, Justin, and David Reiley. 2012. "The Economics of Spam." *The Journal of Economic Perspectives* 26, no. 3: 87-110. https://doi.org/10.1257/jep.26.3.87.

Shahariar, G. M., Swapnil Biswas, Faiza Omar, Faisal Muhammad Shah, and Samiha Binte Hassan. 2019. "Spam Review Detection Using Deep Learning." Paper presented at *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference, Vancouver, Canada. October 17-19th, 2019.* https://doi.org/10.1109/IEMCON.2019.8936148.

Silva, R. M., A. Yamakami, and T. A. Almeida. 2012. "An Analysis of Machine Learning Methods for Spam Host Detection." Paper presented at *2012 11th International Conference on Machine Learning and Applications*, *Boca Rata, FL. December 12-15th, 2012.* https://doi.org/10.1109/ICMLA.2012.161.

Singh, Aasha, Awadhesh Kumar, Ajay Kumar Bharti, and Vaishali Singh. 2022. "An E-Mail Spam Detection Using Stacking and Voting Classification Methodologies." *International Journal of Information Engineering and Electronic Business* 14, no. 6: 27-36. https://doi.org/10.5815/ijieeb.2022.06.03.

Stokel-Walker, Chris, and Richard Van Noorden. 2023. "The Promise and Peril of Generative AI." *Nature (London)* 614, no. 7947: 214–216. https://doi.org/10.1038/d41586-023-00340-6.

Wu, Di, Wei Shi, and Xiangyu Ma. 2021. "A Novel Real-Time Anti-Spam Framework." *ACM Transactions on Internet Technology* 21, no. 4: 1-27. https://doi.org/10.1145/3423153.