International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 3, ISSN No. 2455-2143, Pages 157-161
Published Online July 2022 in IJEAST (http://www.ijeast.com)

DETECTING SPAM EMAIL WITH MACHINE LEARNING OPTIMIZED WITH BIO-INSPIRED METAHEURISTIC ALGORITHMS

Srinath N R
PG Student, Department of CSE,
Akshaya Institute of Technology, Tumakuru, Karnataka
Visvesvaraya Technological University,
Belagavi, Karnataka, India

Roopa T
Assistant Professor, Dept of CSE,
Akshaya Institute of Technology, Tumakuru, Karnataka
Visvesvaraya Technological University,
Belagavi, Karnataka, India

***Abstract*— Groups and individuals used email regularly, for sending and receiving messages. When spammers get to know any valid email address, then they can exploit it easily by sending unsolicited emails, which will directly reach our mail inbox. Detecting spam using the machine learning technique is a known way, but we want to improve the accuracy, so the ML algorithms were optimized with bio-inspired methods to achieve better accuracy. The research was done to implement ML models using many algorithms, such as Naive Bayes, Random Forest, Support Vector Machine, and Decision Tree on a particular email dataset, along with feature extraction and pre-processing. The techniques occurring in nature can be used to achieve better accuracy, viz. Genetic Algorithm and Particle Swarm Optimization.**

Keywords: **Spam Email Detection, Machine Learning.**

I. INTRODUCTION

Machine learning models can be used to solve a variety of problems in various fields. Emails are used daily for communication and for socializing. Security breaches that compromise customer data allows 'spammers' to spoof a compromised email address to send illegitimate (spam) emails. In a phishing attack, user is tricked to open the spam link inside the spam email, providing unauthorized access to their device. Several companies offer tools to detect in a network any spam emails. Firewalls are often configured and setup with complex rules within an organization to filter unsolicited emails. Google catches almost all the spam, more than 99%. One can deploy spam filter on the gate way (router), on the cloud-hosted applications, or on the user's computer, there are many choices. To overcome the detection problem of spam emails, methods such as content-based filtering, rule-based filtering, or Bayesian filtering have been applied. In 'knowledge engineering' spam detection rules are set up and have be regularly updated manually thus it's a laborious activity, on the contrary, during the creation of ML model in the training phase, the algorithm learns how to recognize which is spam and which is ham automatically and then applies that learned knowledge to unknown incoming emails. The proposed spam detection to resolve the issue of the spam classification problem can be experimented further by automated parameter selection for the models or feature selection. In this research, experiment was conducted with five different ML models with Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). This will be compared with the base models to conclude whether the proposed models have improved the performance with parameter tuning.

II. LITERATURE SURVEY

Researchers have taken a lead to implement machine learning models to detect spam emails. In the paper [3], the authors have conducted experiments with six different machine learning algorithms: Naïve Bayes (NB) classification, K-Nearest Neighbour (K-NN), Artificial Neural Network (ANN), Support Vector Machine (SVM), Artificial Immune System and Rough Sets. Their aim of the experiment was to imitate the detecting and recognizing ability of humans. Tokenization was explored and the concept provided two stages: Training and Filtering. Their algorithm consisted of four steps: Email Pre-Processing, Description of the feature, Spam Classification and Performance Evaluation. It concluded that the Naïve Bayes provided the highest accuracy, precision, and recall. Feng et al. [1] describes a hybrid system between

two machine learning algorithms i.e., SVM-NB. Their proposed method is to apply the SVM algorithm and generate the hyperplane between the given dimensions and reduce the training set by eliminating datapoints. This set will then be implemented with NB algorithm to predict the probability of the outcome. This experiment was conducted on Chinese text corpus. They successfully implemented their proposed algorithm and there was an increase in accuracy when compared to NB and SVM on their own. Mohammed et al. [4] tried to detect the unsolicited emails by experimenting with different classifiers such as: SVM, KNN, NB, Tree and Rule based algorithms. They generated a vocabulary of Spam and Ham emails which is then used to filter through the training and testing data. Their experiment was conducted with Python programming language on Email-1431 dataset. They concluded that NB was the best working classifier followed by Support Vector Machine. Wijaya and Bisri [5] proposes a hybrid-based algorithm, which is integrating Decision Tree with Logistic Regression along with False Negative threshold. They were successful in increasing the performance of DT. The results were compared with the prior research. The SpamBase dataset was used to conduct the experiment. The proposed method presented a 91.67% accuracy.

Agarwal and Kumar [6] experimented with NB along with Particle Swarm Optimization (PSO) technique. The paper used the emails from Ling-Spam corpus and aimed to acquire an improvement in F1-score, Precision, Recall and Accuracy. The paper used Correlation Feature Selection (CFS) to select appropriate features from the dataset. The dataset was split into 60:40 ratio. Particle Swarm Optimization was integrated along with Naïve Bayes. They concluded a success when their proposed integrated method increased the accuracy of the detection compared to NB alone. Belkebir and Guessoum [7] used SVM along with Bee Swarm Optimization (BSO) and Chi-Squared on Arabic Text. Since there have been plenty of research conducted for text mining on English and some European languages, the authors considered to review the algorithms work on Arabic language. They experimented with three different approaches to categorize automatic text – Neural networks, Support Vector Machine (SVM) and SVM optimizing with Bee Swarm Algorithm (BSO) along with Chi-Squared. Bee Swarming Optimization algorithm is inspired by the behavior of swarm of bees to achieve global solution. A search area is divided and each area within the divided section is assigned to other bees to explore. Every solution is distributed amongst the bees and the best solution is accepted and the process is repeated until the solution meets the criteria of the problem. The main problem advertised is: ‘‘The problem of selecting the set of attributes is NP-hard’’. The research explains the problem dealing with the feature selection due to the computation time. A vocabulary is generated and fed into the Chi2-BSO algorithm to acquire the features and finally the achieved result is loaded within the SVM algorithm. The experiment was carried on OSAC dataset which included 22,429 text records. The study randomly selected 100 texts from each category distributed by 70:30 ratio. The program performed removal of digits, Latin alphabets, isolated letters, punctuation marks and stop words. The document representation step was conducted with different modes for all approaches – SVM, BSO-CHI-SVM and artificial neural network (ANN). The SVM outperformed the ANN execution time. The algorithm BSO-CHI-SVM exceeds the learning time, but it is still identified as effective. The paper concluded that the proposed algorithm provides an accuracy rate of 95.67%. They have also stated that SVM approach outperformed ANN. A further development is to evaluate the approach of this article on other datasets and use modes such as n-gram or concept representation.

III. ISSUES IN THE EXISTING SYSTEM

Many tools and techniques are offered by companies in order to detect spam emails in a network. Organizations have set up filtering mechanisms to detect unsolicited emails by setting up rules and configuring the firewall settings. Google is one of the top companies that offers 99.9% success in detecting such emails. There are different areas for deploying the spam filters such as on the gateway (router), on the cloud hosted applications or on the user’s computer. In order to overcome the detection problem of spam emails, methods such as content-based filtering, rule-based filtering or Bayesian filtering have been applied.

Unlike the ‘knowledge engineering’ where spam detection rules are set up and are in constant need of manual updating thus consuming time and resources, Machine learning makes it easier because it learns to recognize the unsolicited emails (spam) and legitimate emails (ham) automatically and then applies those learned instructions to unknown incoming emails.

IV. PROPOSED SYSTEM

We will use various ML algorithms such as Support Vector, Random Forest, Naïve Bayes and Decision Tree, since spam email detection falls into classification category, supervised learning method will be used. Supervised learning is a concept where the dataset is split into two parts: 1) Training data and 2) Testing data. The main aim of this learning method is to train a classifier with a given data and parameters and then predict the outcome with the testing dataset which will not be known to the program or classifier.

Once we test the base models, then we add bio-inspired implementation to these base models and see if the accuracy improves.

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 3, ISSN No. 2455-2143, Pages 157-161
Published Online July 2022 in IJEAST (http://www.ijeast.com)

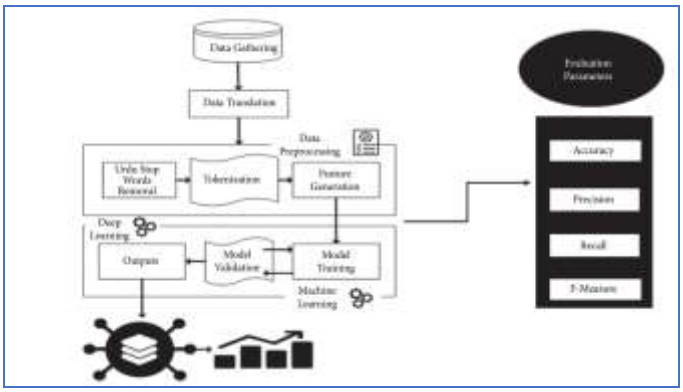

Figure 1: The Architecture of the spam detection using machine learning.

Figure 2. Stages in ML process.

Figure 3. Workflow of the blockchain-based MCS.
Step 1. Pick an email randomly from the collection.
Step 2. Email is in unprocessed state. Email must be pre-processed before the feature extraction and classification procedure can begin. Tokenization,
Step3. To use the feature extraction technique, select suitable attribute words from the validation set. Just the set of features that is most nearly connected to the category is selected.
Step4. Use extracted features and created tokens to train ML models. Later, model can easily distinguish between spam and ham emails.
Step5. Tokens are classified as spam or ham based on their feature similarity as ML models determines.

In this experiment we used Django, a python-based web framework, which allows the development of web applications rapidly. Though in machine learning projects generally Jupyter notebooks is preferred for training and testing the model, but since this is PG level project, we have tried to develop a mini web-app so that user can register, and login and then load the dataset into the system, the Django web app then uses python libraries such as numpy and pandas which provides easy to use high performance structures and data analysis tools. The web-app uses sklearn library as the library provides implementation of most of the algorithms that we plan experiment in this project. The sklearn library provides implementation for Support Vector, Random Forest, Multinomial Naïve Bayes and Decision Tree. Since spam email detection falls into classification category, supervised learning method will be used. Supervised learning is a concept where the dataset is split into two parts: 1) Training data and 2) Testing data. The main aim of this learning method is to train a classifier with a given data and parameters and then predict the outcome with the testing dataset which will not be known to the program or classifier. The models will be trained with a training dataset of 60%, 70%, 75% and 80%. Once the model is trained, model will be provided with the testing dataset which is distributed as 40%, 30%, 25% and 20% respectively with training dataset. This will provide a better knowledge of what percentage split is best suited and thus be more efficient to work with majority of the datasets. This will provide results on classifiers working best with more or less training data.

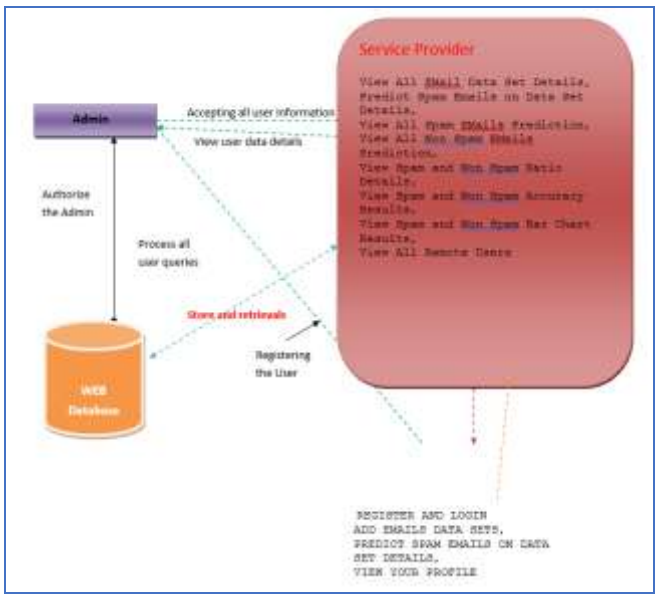

Figure 2. Architecture diagram of Django App.

V. RESULT

The project successfully implemented models combined with bio-inspired algorithms. The spam email corpus used within the project were mostly. Several emails were tested with the proposed models. The numerical corpuses (PU) had restrictions in terms of feature extraction as the words were replaced by numbers. But the alphabetical corpuses performed better in terms of extraction of the features and predicting the outcome. We ran the dataset on different classification algorithms and provided the top 4 algorithms: Multinomial Naïve Bayes, Support Vector Machine, Random Forest, and Decision Tree. These algorithms were then tested and experimented with Scikit-learns library and its modules. This resulted in upgrading the SVM module with SGD classifier, which acts the same as SVM but performs better on the large datasets. SGD was implemented using Python and experimented with feature extraction and stop words removal along with converting the tokens for the algorithms to process. Genetic Algorithm worked better overall for both text-based datasets and numerical-based datasets than PSO. The PSO

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 3, ISSN No. 2455-2143, Pages 157-161
Published Online July 2022 in IJEAST (http://www.ijeast.com)

worked well for Multinomial Naïve Bayes and Stochastic Gradient Descent, whereas GA worked well for Random Forest and Decision Tree. Naïve Bayes algorithm was proved to have been the best algorithm for spam detection. This was concluded by evaluating the results for alphabetical based dataset. The highest accuracy provided was 100% with GA optimization on randomized data distribution for 80:20 train and test split set on Spam Assassin dataset. In terms of F1-Score, precision and recall, Genetic Algorithm had more impact than PSO on MNB, SGD, RF and DT.

Stratified K-fold cross-validation - accuracy.

Classifier	Split Set			
	60:40	70:30	75:25	80:20
SGD	96.79%	96.80%	96.98%	96.92%
MNB	90.26%	90.69%	90.71%	90.62%
RF	84.82%	85.04%	85.54%	85.92%
DT	91.79%	92.03%	91.93%	92.38%
MLP	95.98%	95.36%	96.18%	96.25%

Table 1.Accuracy for base models with different split combinations.

Figure 3. Accuracy charts for various algorithms in one of the runs on a small dataset.

PSO 80:20 split set.

Classifier	Accuracy	F1-score	Precision	Recall
SGD	97.64%	95.78%	96.80%	95.59%
MNB	98.47%	97.54%	97.23%	97.86%
RF	90.81%	74.79%	96.11%	66.49%
DT	92.28%	86.71%	88.07%	86.45%

Table 2. Performance parameters for base models + PSO

GA 80:20 split set.

Classifier	Accuracy	F1-score	Precision	Recall
SGD	97.77%	96.71%	97.61%	95.97%
MNB	98.47%	97.67%	98.01%	97.59%
RF	94.36%	87.42%	97.79%	81.74%
DT	93.42%	89.54%	91.07%	88.51%

Table 2. Performance parameters for base models + GA

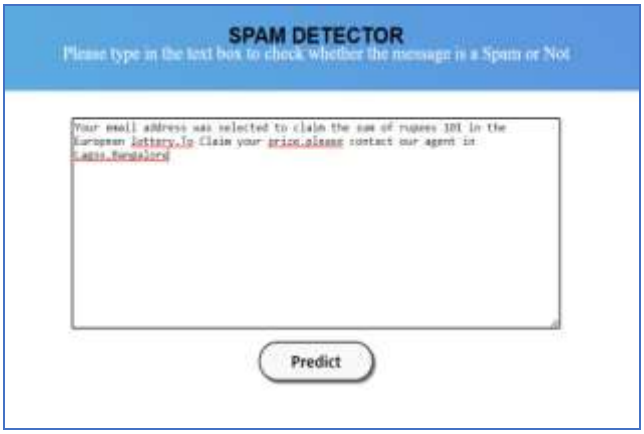

Figure 4. UI to predict if mail is spam or not

VI. REFERENCE

[1] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang (2016), ‘‘A support vector machine based Naive Bayes algorithm for spam filtering,’’ in Proc. IEEE 35th Int. Perform. Comput. Commun. Conf. (IPCCC), Dec. 2016, pp. 1–8, doi: 10.1109/pccc.2016.7820655.

[2] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa (2019), ‘‘Machine learning for email spam filtering: Review, approaches and open research problems,’’ Heliyon, vol. 5, no. 6, Jun. 2019, Art. no. e01802, doi: 10.1016/j.heliyon.2019.e01802.

[3] W. Awad and S. ELseuofi (2011), ‘‘Machine learning methods for spam E-Mail classification,’’ Int. J. Comput. Sci. Inf. Technol., vol. 3, no. 1, pp. 173–184, Feb. 2011, doi: 10.5121/ijcsit.2011.3112.

[4] S. Mohammed, O. Mohammed, and J. Fiaidhi (2013), ‘‘Classifying unsolicited bulk email (UBE) using Python machine learning techniques,’’ Int. J. Hybrid Inf. Technol., vol. 6, no. 1, pp. 43–55, 2013. [Online]. Available: https://www.researchgate.net/publication/236970412_Classifying_ Unsolicited_Bulk_Email_UBE_using_Python_Machine_Learning_ Techniques

[5] A. Wijaya and A. Bisri, ''Hybrid decision tree and logistic regression classifier for email spam detection,'' in Proc. 8th Int. Conf. Inf. Technol. Electr. Eng.

[6] K. Agarwal and T. Kumar (2018), ''Email spam detection using integrated approach of Naïve Bayes and particle swarm optimization,'' in Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS), Jun. 2018, pp. 685–690, doi: 10.1109/ICCONS.2018.8662957.

[7] R. Belkebir and A. Guessoum (2013), ''A hybrid BSO-Chi2-SVM approach to arabic text categorization,'' in Proc. ACS Int. Conf. Comput. Syst. Appl. (AICCSA), Ifran, Morocco, May 2013, pp. 1–7, doi: 10.1109/ AICCSA.2013.6616437.

[8] A. I. Taloba and S. S. I. Ismail (2019), ''An intelligent hybrid technique of decision tree and genetic algorithm for E-Mail spam detection,'' in Proc. 9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS), Cairo, Egypt, Dec. 2019, pp. 99–104, doi: 10.1109/ICICIS46948.2019.9014756.

[9] R. Karthika and P. Visalakshi (2015), ''A hybrid ACO based feature selec- tion method for email spam classification,'' WSEAS Trans. Comput, vol. 14, pp. 171–177, 2015. [Online]. Available: https://www.wseas.org/ multimedia/journals/computers/2015/a365705-553.pdf.

[10] S. L. Marie-Sainte and N. Alalyani (2020), ''Firefly algorithm based feature selection for arabic text classification,'' J. King Saud Univ.-Comput. Inf. Sci., vol. 32, no. 3, pp. 320–328, Mar. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S131 915781830106X

[11] E. A. Natarajan, S. Subramanian, and K. Premalatha (2012), ''An enhanced cuckoo search for optimization of bloom filter in spam filtering,'' Global J. Comput. Sci. Technol., vol. 12, no. 1, pp. 75–81, 2012. Accessed: Jan. 18, 2020. [Online]. Available: https://globaljournals.org/ GJCST_Volume12/12-An-Enhanced-Cuckoo-Search-for-Optimization.pdf

[12] A. Géron (2019), Hands-On Machine Learning With Scikit-Learn, Keras, and TensorFlow, 2nd ed. Newton, MA, USA: O'Reilly Media, 2019, Ch. 1.

[13] (2019). 1. Supervised Learning—Scikit-Learn 0.22.2 Documentation. Accessed: Oct. 9, 2019. [Online]. Available:https://scikit-learn.org/stable/supervised_learning.html

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas (2011), ''Scikit-learn: Machine learning in Python,'' J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Oct. 2011. [Online]. Available: http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

[15] S.ZhuandF.Chollet.(2019).WorkingWithRNNs.Accessed:Nov.2,2019. [Online]. Available: https://keras.io/guides/working_with_rnns/