ZKG INTERNATIONAL

ISSN: 2366-1313

An approach for Spam Email classification with Machine Learning

Mr. Naresh Alvala, Yadlapati VaaniSri, Gowravaram Ruthvik Reddy, Thota Himaja Sree, Idamakanti YuvaKiran, Naga Praneeth Madasu

[1]Associate Professor, CSE Department, GITAM University, Rudraram, Hyderabad 502329,India.

[23456]UG GITAM University, Rudraram, Hyderabad 502329,India.

Abstract:

Many individuals and organizations have found electronic mail to be a more convenient way of communicating. Spammers use this method to send unsolicited email and make fraudulent gains. This article will present a machine learning algorithm that detects spam emails using bio-inspired algorithms. To find the most efficient methods, we reviewed literature on various datasets. On seven different email datasets, extensive research was conducted to develop machine learning models using Naive Bayes and Support Vector Machines, Random Forest, Decision Tree and Multi-Layer Perceptron. Also, feature extraction and pre-processing were performed. To optimize classifier performance, bio-inspired algorithms such as Particle Swarm Optimization or Genetic algorithms were used. The Multinomial Naive Bayes and Genetic Algorithm performed best overall.

Keywords: Machine learning, bio-inspired algorithms, cross-validation, particle swarm optimization, genetic algorithm.

1 Introduction:

The use of machine learning models in computer science has been used for many purposes, from solving network traffic issues to detecting malware. Many people use email regularly for communication and socializing. Spammers can use security breaches to compromise customer data to send spam emails to an illegitimate email address. This can also be used to gain unauthorized access to the device. The spam link in the spam email is tricked into clicking it, which constitutes a phishing attempt [1]. Companies offer many tools and techniques to detect spam email in a network. Companies have implemented filtering mechanisms in order to identify unsolicited email. By setting up rules and configuring firewall settings. Google is a top company that has 99.9% success in detecting spam emails [2]. The spam filters can be deployed in different places, such as the router, cloud-hosted applications, or the user's own computer. Methods such as Bayesian filtering, rule-based filtering and content-based filtering are used to eliminate spam email detection. Contrary to 'knowledge engineering' which has spam detection rules that are constantly updated and need to be set up, machine learning is easier. It learns to recognize legitimate and unsolicited email (ham) and applies the learned instructions to unknown incoming mails [2].

To further investigate the proposed spam detection, feature selection and automated parameter selection can be made for the models. Five machine learning models are used in this research, including Particle Swarm Optimization and Genetic Algorithm. These will be compared to the base models in order to determine if the proposed models have improved performance with parameter tuning.

II Related Work

Researchers are leading the implementation of machine learning models to detect spam email. The paper [3] describes six machine learning algorithms that were used to detect spam emails. Neighbour (KNN), Artificial Neural Networks (ANN), Support Vector Machines (SVM), Artificial Immune System (AIS) and Rough Sets. The experiment's purpose was to mimic the human ability of recognising and detecting objects. The concept of tokenisation was explored. It provided two stages: Filtering and Training. The algorithm was comprised of four steps: Email pre-processing, description of the feature and spam classification, as well as performance evaluation. The highest precision, recall and accuracy was achieved by the Naive Bayes, according to the study. Feng et al. [1] describes a hybrid of two machine-learning algorithms, i.e. SVM-NB. The proposed method involves applying the SVM algorithm to generate the hyperplane between given dimensions. Datapoints are then eliminated from the training set. The NB algorithm will be used to predict the outcome. This experiment was done on a Chinese text corpus. The proposed algorithm was implemented successfully and accuracy increased compared to NB or SVM alone.

Mohammed et al. [4] The goal was to detect unsolicited email by using different classifiers like NB, SVM KNN Tree, Rule-based algorithms, and KNN. They created a vocabulary of Spam/Ham emails that was used to filter through the testing and training data. The experiment was done using Python programming language and the Email-1431 dataset. They found that NB was the most effective working classifier, followed closely by Support Vector Machine. Wijaya & Bisri [5] propose a hybrid-based algorithm that combines Logistic Regression with Decision Tree along with False Negative threshold. They were able to increase the performance of DT. These results were compared to the previous research. The SpamBase dataset was used for the experiment. The proposed method achieved a 91.67% accuracy.

To optimize the performance of the ML algorithm, many researchers have also looked into human evolutionary processes. Taloba [8] and Ismail (9) explored Genetic Algorithm optimization by integrating Decision Tree (DT) with it. The authors recognize the problem of overfitting due to dimension of feature space. They attempt to solve this issue using feature extraction with Principle Component Analysis. This paper provides a detailed background on algorithms and then proceeds to the proposed algorithm. Further research was done on optimization algorithms such as Firefly or Cuckoo search. SVM was used to implement the Firefly algorithm described in [10].

Researchers experimented with the Arabic text and feature selection.

The paper concluded that the proposed method is superior to the SVM. The paper [11] proposes Enhanced Cuckoo Search for bloom filter optimization. Here is where the spam word's weight is taken into account. It was concluded that

Their proposed optimization technique, ECS, outperforms the standard Cuckoo search. This research provided insight into both hybrid systems and optimization techniques. Bio-inspired techniques are more effective in detecting spam emails.

III Methodology:

This research will test Bio-inspired algorithms and Machine learning models. This research will be done on publicly available spam email corpora. This paper is designed to accomplish the following goals:

1) To investigate machine learning algorithms to solve the spam detection problem.

2) To examine the algorithm's operation with the data.

3. To implement bio-inspired algorithms.

4) To compare and test the accuracy of base models with bioinspired implementation.

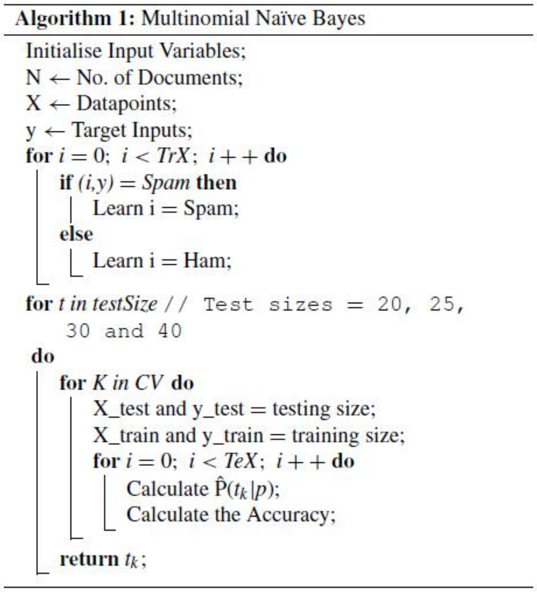
5) To implement the framework with Python.

Scikit-Learn library is used to run the experiments using Python. This will allow you to edit the models and perform pre-processing. You can also calculate the results. The optimization techniques will be used to further optimize the program scripts and they will be compared to the base results, i.e. with default parameters. The spam detection engine must be able take in email data and, with text mining and optimized supervision algorithms, should be capable of classifying the email as spam or ham..

IV Proposed Method

Kumar and Agarwal [6] tried NB in combination with the Particle Swarm Optimisation technique (PSO). This paper was based on emails from LingSpam corpus. It aimed to improve F1-score accuracy, precision, recall, and precision. Correlation Feature Selection was used (CFS), to choose the appropriate features from the data. The dataset was divided into a 60:40 ratio. Particle Swarm Optimization was combined with Naive Bayes. Their proposed integrated method improved the accuracy of detection by 6 times more than NB alone. Belkebir [7] and Guessoum [7] looked at the SVM algorithm in conjunction with Bee Swarm Optimization and Chi-Squared for Arabic Text. The authors reviewed the algorithm work on Arabic text, as there has been a lot of research on text mining in English and other European languages. They tried three approaches to categorize automatic text: Neural, Semantic, and Multimodal.

Networks, Support Vector Machine (SVM), and SVM optimizing using Bee Swarm Algorithms (BSO) together with Chi-Squared. To achieve a global solution, Bee Swarming Optimization algorithm was inspired by the behavior of swarms of bees. Each bee is given a section of the search area to explore. Each solution is shared among the bees, and the best one is accepted. The process continues until the solution meets all criteria.



Algorithm 1: Multinomial Naïve Bayes

```
Initialise Input Variables;
N ← No. of Documents;
X ← Datapoints;
y ← Target Inputs;
for $i = 0;\ i < TrX;\ i{+}{+}$ do
    if $(i,y) = Spam$ then
        Learn i = Spam;
    else
        Learn i = Ham;
for $t$ in $testSize$ // Test sizes = 20, 25,
  30 and 40
do
    for $K$ in $CV$ do
        X_test and y_test = testing size;
        X_train and y_train = training size;
        for $i = 0;\ i < TeX;\ i{+}{+}$ do
            Calculate $\hat{P}(t_k|p)$;
            Calculate the Accuracy;
    return $t_k$;
```

The predictive method is the basis of the Decision Tree model. The model creates a category, which is then further divided into sub-categories. The algorithm continues until the user terminates or the program reaches its end.

The data is used to train the model, which predicts the data's value. A tree that is longer or deeper will have more complex rules. The Random Forest (RF), algorithm can be used both for classification and regression. The algorithm uses multiple decision trees to predict the classes. Each tree predicts classification class. The RF model is used to determine which class will be assigned as a prediction.

Two key applications are used to introduce the proposed framework. One is the anaconda prompt, which is very similar to the standard command prompt, and the other is Jupyter. Jupyter is an integrated python-development environment. Anaconda prompt can be used to run conda and anaconda commands without changing directories. It also allows you to connect to the local host to download and extract packages.

After all packages are checked and processed, you can access the local host with jupyter. This includes many code cells.

A. Dataset Extraction

The dataset is used to collect the first data, which in this case is Twitter messages. After the data is collected, it is cleaned by removing extra spaces and removing duplicates.

B. Collecting Metadata

The cleaned dataset is used to implement the RB features. The first step is to identify the message's time frame. Once the time frame has been identified, it is compared to the threshold rating deviation. This is where the diversity and variance of spammers are checked. The metadata about the spam message is then collected.

C. Generalize messages

All messages on Twitter are collected and analyzed, regardless of spam. It is possible to save a lot time by generalizing messages.

D. Implementing ML algorithms

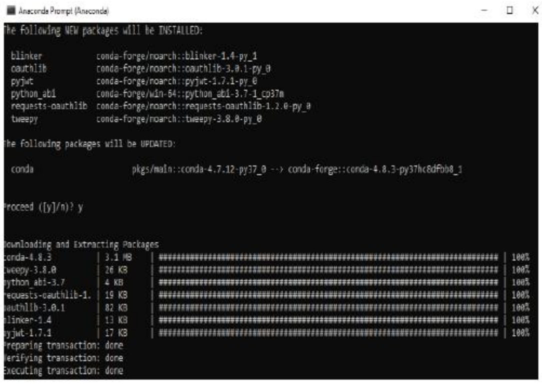
This stage is where the ML algorithms are implemented by segregating messages into spam content or original content. There are many ML algorithms that can be used, including Random forest, Bayes Network and Naive Bayes, as well as K-nearest neighbour and support vector machines.

E. Generating Spam Text Data, and information about Spammer

Once the ML algorithms are implemented, spam messages will be identified and obtained. The information about the spammer responsible for the message will also be collected. This information

ISSN: 2366-1313

allows you to access the entire spammer's history and allow you to analyze all of his messages.



```
The following NEW packages will be INSTALLED:

  blinker            conda-forge/noarch::blinker-1.4-py_1
  oauthlib           conda-forge/noarch::oauthlib-3.0.1-py_0
  pyjwt              conda-forge/noarch::pyjwt-1.7.1-py_0
  python_abi         conda-forge/win-64::python_abi-3.7-1_cp37m
  requests-oauthlib  conda-forge/noarch::requests-oauthlib-1.2.0-py_0
  tweepy             conda-forge/noarch::tweepy-3.8.0-py_0

The following packages will be UPDATED:

  conda                      pkgs/main::conda-4.7.12-py37_0 --> conda-forge::conda-4.8.3-py37hc8dfbb8_1


Proceed ([y]/n)? y


Downloading and Extracting Packages
conda-4.8.3          | 3.1 MB    | ############################################################################ | 100%
tweepy-3.8.0         | 26 KB     | ############################################################################ | 100%
python_abi-3.7       | 4 KB      | ############################################################################ | 100%
requests-oauthlib-1. | 19 KB     | ############################################################################ | 100%
oauthlib-3.0.1       | 82 KB     | ############################################################################ | 100%
blinker-1.4          | 13 KB     | ############################################################################ | 100%
pyjwt-1.7.1          | 17 KB     | ############################################################################ | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Figure 1: Command prompt calling the model

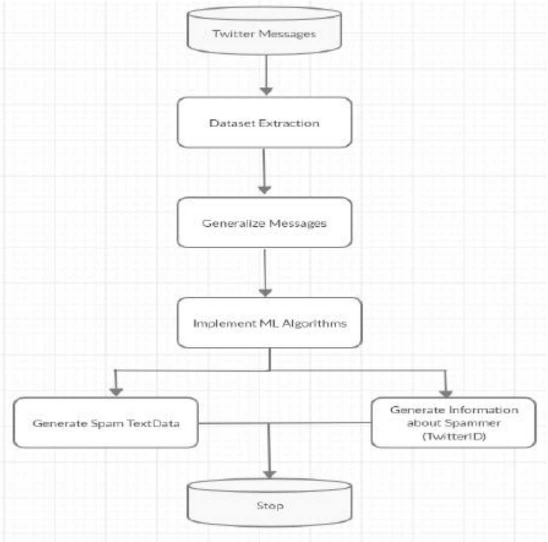


Figure 2 :Flow Diagram

The flow diagram shows the entire flow and steps of the framework. Feature Engineering is available. Therefore, features of rawdata can be easily extracted with the help of data mining. It is used improve the performance of Machine learning algorithms.

Each and every data obtained is accurate.

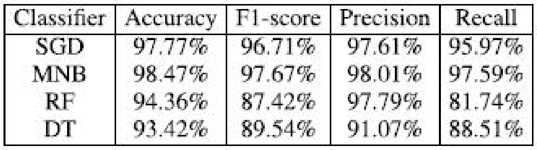
Spam Features are as a built in function.

Less human interaction.

It is statistics based approach.

Supports review centric spam detection.

Supports reviewer centric spam detection.



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%

Figure 3 : Accuracy calculations

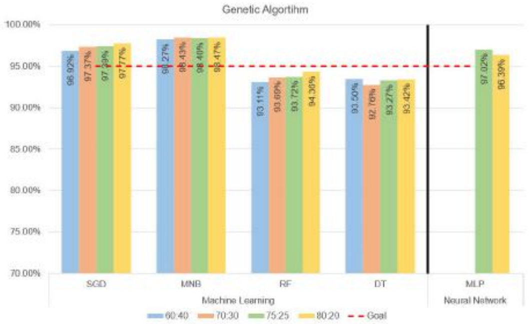


Figure 4: Graph representing the models accuracy

V Conclusion

This paper uses machine learning algorithms and NLP concepts to identify spammers and spammers in a Twitter dataset. The entire spammer's details can be accessed by reviewing the spam. This helps in identifying other spammers, spammers, and their style of writing messages. Two attribute sets were considered, which include content and user behavior. The content was determined using the average content similitude and maximum content similitude, as well as the ratio of exclamation sentences, and the ratios of first personal pronouns. Properties such as reviews written, average negative ratio and average number of complaints are used to determine the user's behavior. This makes it an effective and precise spam detection tool.

VI References

1. Nurul Fitriah Rusland, Norfaradilla Wahid, Shahreen Kasim, Hanayanti Hafit, “Analysis of Naive Bayes Algorithm for Email Spam Filtering across Multiple Datasets”.

2. J. Rout, S. Singh, S. Jena, and S. Bakshi, “Deceptive Review Detection Using Labeled and Unlabeled Data”.

3. Feng Qian, Abhinav Pathak, Y. Charlie Hu, Z. Morley Mao, and Yinglian Xie, “A Case for Unsupervised-Learning-based Spam Filtering”.

4. Shrawan Kumar Trivedi, “A Study of Machine Learning Classifiers for Spam Detection”.

5. W.A. Awad, S.M. ELseuofi, “Machine Learning Methods for Spam E-mail Classification”

6. S. Gharge, and M. Chavan``An integrated approach for malicious tweets detection using NLP," in Proc. Int. Conf. Inventive Commun. Comput.Technol. (ICICCT), Mar. 2017, pp. 435_438.

7. T. Wu, S. Wen, Y. Xiang, and W. Zhou, ``Twitter spam detection: Survey of new approaches and comparative study," Comput. Secur., vol. 76, pp. 265_284, Jul. 2018.

8. M. Mateen, M. A. Iqbal, M. Aleem, and M. A. Islam, ``A hybrid approach for spam detection for Twitter," in Proc. 14th Int. Bhurban Conf. Appl. Sci.Technol. (IBCAST), Jan. 2017, pp. 466_471.

9. F. Fathaliani and M. Bouguessa, ``A model-based approach for identifying spammers in social networks," in Proc. IEEE Int. Conf. Data Sci. Adv.Anal. (DSAA), Oct. 2015, pp. 1_9.

10. Saeedreza Shehnepoor, Mostafa Salehi*, Reza Farahbakhsh, Noel Crespi, “NetSpam: a Network-based Spam Detection Framework for Reviews in Online Social Media “

11. G. Jain, M. Sharma, and B. Agarwal, ``Spam detection in social media using convolutional and long short term memory neural network," Ann. Math. Artif. Intell., vol. 85, no. 1, pp. 21_44, Jan. 2019.

12. C. Meda, F. Bisio, P. Gastaldo, and R. Zunino, ``A machine learning approach for Twitter spammers detection," in Proc. Int. Carnahan Conf.Secur. Technol. (ICCST), Oct. 2014, pp. 1_6