COMPARTIVE STUDY OF MACHINE LEARNING APPROACHES IN DETECTING SPAM EMAILS

Reagan Phung
Computer science department
of Missouri State University
E-mail:pmh1407@live.missouristate.edu

Abstract*—In recent years, going along with the development of internet and social networks, the number of emails used to exchange information or market new products to customers witnessed a giant leap forward in both size and number. Then comes spammers who take advantage of email popularity to send indiscriminately unsolicited emails. According to statistics of Global Spam Volume [1], spam messages and emails make up for approximately 60% of the global email traffic. Despite the fact that technology has advanced in the field of Spam detection since the first unsolicited bulk email was sent in 1978 spamming remains a time consuming and expensive problem. Therefore, in our paper we focus on researching how to detect spam email by machine learning approach, including deep learning. What set this study apart from other comparative studies is the use of n-gram models for language modeling. To make it clear, I create an experiment to go through the most popular supervised learning algorithm that are currently used (Bayesian classifier, K-NN, J48 decision tree, Random Forest and Support Vector Machine) with n-gram feature extraction to find the best method should be selected for detecting spam email. After that, the best algorithms will be compared with a ubiquitous Deep Learning algorithm for language modeling, which is LSTM, to give conclusion on the strength and drawbacks of each approach.

Keywords— Machine Learning, Deep learning, n-gram, LSTM, Natural language processing, email filter, spam email

I. INTRODUCTION

According to Nucleus Research [2], spam costs US businesses an average of $712 per employee every year due to diminished productivity, lost customers, spent bandwidth and increasing the cost of maintenance. Estimates are shown by Statista [3] that slightly less than 60 percent of the incoming business email traffic is unsolicited bulk email (known as spam) which was the lowest level since 2003. However, even though the global percentage of spam/ non-spam ratio is decreasing, the competition between spammers and spam filtering techniques continues. It is fair to say that the problem is not going away, and the need for reliable anti-spam filters remains high.

The idea of automatically classifying spam and non-spam emails by applying machine learning methods has been pretty popular in academia and has been a topic of interest for many researchers.

There are currently two main approaches proved to be most efficient in this field: knowledge engineering and machine learning. The first solution focuses on creating a knowledge-based system in which pre-defined rules dictate if an incoming message is legitimate or not. The primary disadvantage of this method is that those rules need to be maintained and updated continuously by the user or a 3rd party like for example a software vendor.

The machine learning approach, in contrast, does not require pre-defined rules, but instead messages which have been successfully pre-classified. Those messages make the training dataset which is being used to fit the learning algorithm to the model. One could say the algorithm defers the classification rules from the test data.

In my paper, I choose to apply machine learning to solve this issue and our goal is to find out the best approach by comparing their F1 score and accuracy based on the given data set. We also included a Deep Learning model to see if there's a significant difference between traditional Machine Learning approach and the modern Deep Learning approach. Five machine learning algorithms are employed (K-NN, Naïve Bayes, SVMs, Decision Tree and Random Forest). Feature extraction is based on n-grams and TF-IDF (term frequency – inverse document frequent) for all algorithms to remain fair in the final result. 3-fold cross-validation is used to separate the data set. The selected model for Deep Learning is LSTM (Long short-term memory).

II. MOTIVATION

We know email classification can be approached by both Deep Learning and Machine Learning. Traditional Machine Learning algorithms are non-sequential classifiers since they do not handle the words in the emails in a sequence. A sequential classifier, such as LSTM, handle each word in the email sequential, which allows it to capture relations between words better and possibly utilize the content of the email better than a non-sequential classifier. Therefore, LSTM obviously have an advantage over ML algorithms. But if traditional ML algorithms have sequence modelling characteristics, will it win in the race against LTSM? But then, we will need to investigate the best algorithms for the comparative study. The problem requires three condition to be satisfied: First is to find the best traditional machine learning algorithm, second is to build the LSTM model, third is to find a sequential-like algorithm/model to compare.

Fortunately, n-gram is widely known as a ubiquitous model for language modelling which interact with sequences. Therefore, I plan to integrate it into traditional Machine Learning algorithm for email classification task.

III. RESEARCH QUESTIONS

1.Why do we need to use machine learning to classify spam email?

2.What is the best machine learning method used for detecting spam email?

3. How well can LSTM, compared to traditional machine learning models, classify emails?

IV. LITERATURE REVIEW

As mentioned above, knowledge engineering and machine learning are two general approaches used in email filtering. For knowledge engineering, a wide range of rules are standardized and play an important role on detecting spam emails. In the research of a group scientists from the University of Jordan [4], they listed out a wide number of feature extraction such as header features, attachment features and payload features, which can be used to classify email. It worked as a set of rules to filter an email and put it in the right category. However, A set of such rules might require a long time to prepare and need to be updated constantly. It inevitably leads to the big waste of efforts. In comparison, machine learning is more likely to deal with the huge amount of data with an efficient cost. particularly, no specific rules are required, only a set of pre-labeled emails need to be presented for training purpose. Machine learning and Deep Learning approach has been widely researched with official papers.

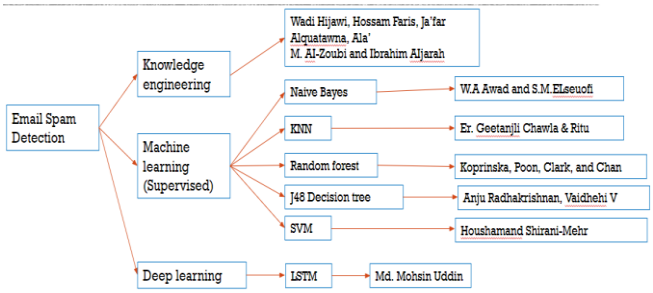
A. Related Works

In the paper of W.A. Awad and S.M.Elseuofi [5], they compared six different machine learning algorithm in the same data set and find out that Naïve Bayes perform the highest precision among the six methods, while the K NN has the worst. Their experiment is quite convincing; however, it is clear that they did not attempt to optimize each method to find out their best performance. For instance, claimed by the group of scientists in India [6], K-NN can be further extended by widening parameters like accuracy, F1-Score, precision and recall which were later collectively used to provide better result. And eventually, they pointed out that their new K-NN model performed much better than Naïve Bayes. In the different approach, Houshmand Shirani-Mehr [7] presented his research on detecting spam for SMS by using Naïve Bayes and Support Vector Machine. He went on with more details by dividing data set on multiple scales, which is done via 10-fold cross-validation and figure out that SVMs is the best model with 97.64% accuracy. However, as we all know, SMS is always far shorter and simpler than emails, so we still need to reperform our own experiment to confirm the result. In the work of Anju Radhakrishnan and Vaidhehi [8], they did experiment about J48 Decision Tree and Naïve Bayes and came to conclusion as J48 Decision Tree outperformed Naïve Bayes. Their work is fascinating, but there's still remained some room to improve. To be clear, their training set used to do experiment is small and not rescaled to improve accuracy rate. Therefore, their result might be not really correct because Naïve Bayes is well-known for dealing with the big load of data.

Regarding Deep Learning approach, Mohsin Uddin[9] proved in his paper that Deep Learning algorithms LSTM and GRU performed better than traditional ML models in terms of F1-Score and accuracy. But their dataset is SMS as well which I will have to reconfirm in my experiment as well.

Finally, after considering some other researches have been done in the field of email filter, I decide to evaluate the accuracy rate of five machine learning algorithms (K-NN, SVMs, Naïve Bayes, Decision Tree, Random Forest) in the same data set. I also optimized the experiment such as improving K-NN, applying 3-fold cross validation when training. Deep Learning model will also be implemented based on the work of traditional Machine Learning model with slight modification. The result of each method will be compared in term of spam accuracy, recall, precision and F1-score.

B. Literature Map



The literature map shows the topic that I focus on, going along with their respective research done by other people.

V. HYPOTHESIS

1.The new approach of K-NN has a very satisfying performance among other methods, while Decision tree is poor and has the worst precision percentage.

2. Deep Learning LSTM will outperform traditional Machine Learning models

VI. REVIEW OF DETECTION METHODS

A. Traditional Machine Learning methods:

Scikit-learn toolkit is used to implement the traditional machine learning algorithms:

Naïve Bayes [9]: It is a probabilistic classifier, which assumes that the features are independent to each other. It is used as a baseline for many ML related research. The multinomial Naïve Bayes (alpha=2.0, class_prior=None, fir_prior=True) is used in my research. NB is highly scalable. It scales linearly with the number of predictors and data points. NB is not sensitive to irrelevant features. NB can be used for both binary and multi-class classification problems and handles continuous and discrete data.

Support Vector Machine (SVM) [9]: SVM is used for a broader classification margin. It uses the kernel trick technique to find the optimal boundary between possible outputs. Support Vectors are simply the coordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line). If the data requires non-linear classification, SVM can employ Kernels, which are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. they convert non-separable problem to separable problem.

Decision trees (DT) [9]: DT are used for classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. Nonlinear relationships between parameters do not affect tree performance. Also, trees can explain the non-linearity in an intuitive manner.

Random Forest [9]: An ensemble learning algorithm which aggregates multiple Machine Learning algorithms together to form a stronger model. Random forest gives much more accurate predictions when compared to simple CART/CHAID or regression models in many scenarios. These cases generally have high number of predictive variables and huge sample size. This is because it captures the variance of several input variables at the same time and enables high number of observations to participate in the prediction.

K-nearest neighbor [9]: Can quickly respond to changes in input. k-NN employs lazy learning, which generalizes during testing--this allows it to change during real-time use. The downside of this algorithm is that it's sensitive to localized data and real time computation slows down learning speed.

B. Deep Learning methods

Long Short-Term Memory (LSTM) [9]: RNN's most often encounter is solved by the improved version of this deep learning algorithm, called Long Short-Term Memory. RNN has a limitation called vanishing gradient problem [10] [11] whereas LSTM resolves the issue. It can trace back to several states and see what happened which ultimately results in taking efficient results.

VII. ALGORITHM IMPLEMENTATION

1.Download "SpamAssasin" dataset through Kaggle.

2.Import and initialize plotting and feature extraction libraries

3.Setting the input folder and reading the filenames

4.Convert different email types to normal text then convert the lists to dataframes, joined the spam and ham dataframes and shuffled the resultant dataframe.

5.Clean the data by removing stop word using Snow Ball Stemmer and split the mail text using regex

For ML models:

6.Vectorization using Tfidf for ngram (1,2) range

7.Split the dataframe into train and test dataframes using k-fold cross validation. The test data was 33% of the original dataset.

8.Trained five models using the training data when ngram(1,2): K-NN, Naive Bayes, Decision Tree, SVM, Random Forest.

For DL models:

9.Reuse same steps of Preprocessing until step five

10.Use Tokenization and Padding from Keras library to replace Vectorization using n-gram

11.Run LSTM model

Finally, in both models, predict the email label for test dataset. Calculated 4 metrics to gauge performance of the models:

- Accuracy
- Precision
- Recall
- F1-score

VIII.PROPOSED METHODOLOGY

A. Data Description

The data set is taken from "SpamAssassin" which contain 3000 emails with the spam rate 16.4%.

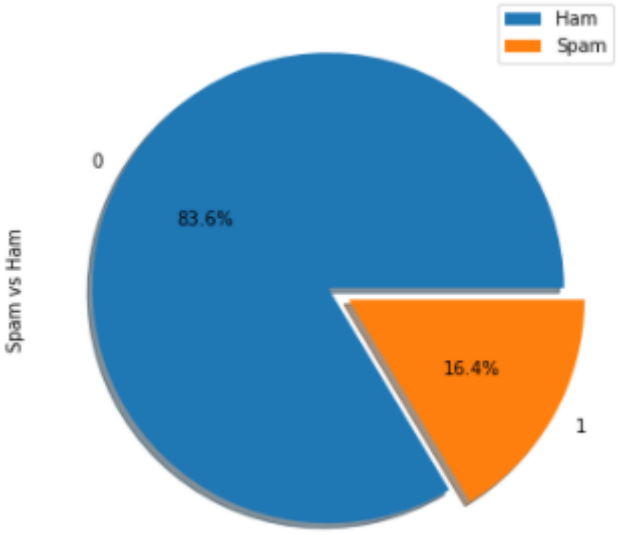


Figure 1: Dataset

B. Feature Extraction

Three feature extraction techniques are used in the proposed model such as n-grams, TF-IDF and word embeddings. Feature extraction for traditional machine learning classifiers is performed using TF-IDF values and n-grams techniques. Keras library word embeddings layer is used for feature selection in case of Long Short-Term Memory (LSTM).

C. Data Modelling using traditional Machine Learning algorithm:

Dataset is divided via 3-fold cross validation. Random sampling is used in data partitioning. Random seed is fixed to have reproducible results.

D. Data Modelling using Deep Learning model:

Keras Functional API was used to build the model. Details of model architectures are described below:

```
Model: "functional_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         [(None, 250)]             0
_________________________________________________________________
embedding_4 (Embedding)      (None, 250, 64)           3200000
_________________________________________________________________
lstm_4 (LSTM)                (None, 64)                33024
_________________________________________________________________
dense_8 (Dense)              (None, 10)                650
_________________________________________________________________
dense_9 (Dense)              (None, 2)                 22
=================================================================
Total params: 3,233,696
Trainable params: 3,233,696
Non-trainable params: 0
```

Figure 2: LSTM model for email classification

This architecture consists of an input layer that has 250 nodes, then an embedding layer with a vocabulary of 50000, a vector space of 64 dimensions in which words will be embedded, and input documents that have 250 words each. The third layer will be LSTM with 64 nodes, then two dense layers with 10 and 2 nodes respectively.

Tanh is chosen as the activation function for the first dense layer, while relu is chosen as the activation function for the second dense layer. Softmax function is chosen for the output layer. The optimizer and loss function are "adam" and "categorical_crossentropy", respectively. To train the model(s), each dataset is divided into 3 subsets: training (64%), validation (16%), test (20%). Random sampling is used in data partitioning. Random seed is fixed to have reproducible results.

E. Mechanism to assure the quality

1. Accuracy [12]:
Accuracy is a measure for how many correct predictions your model made for the complete test dataset. It is measured by the following formula: Accuracy = (TP+TN) / (TP+TN+FP+NP) (Where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative).

2. Precision [12]:
Precision refers to the closeness of two or more measurements to each other. In Machine Learning, precision is the fraction of relevant instances among the retrieved instances. Precision = TP / (TP + FP)

3. Recall [12]:
Recall is how many of the true positives were recalled (found), i.e. how many of the correct hits were also found. Recall = TP / (TP + FN)

4. F1-score [12]:
F1-score is a statistical method for determining accuracy accounting for both precision and recall. It is essentially the harmonic mean of precision and recall.

IX. PRELIMINARY RESULTS

The paper provides a result of the Precision, Recall, and F1-score because accuracy can be misinterpreted sometimes. Precision and Recall are calculated from confusion matrix and F1-score is the harmonic mean of precision and recall. The different applied methods are compared in terms of F1-score and accuracy:

Table 1: Analysis of algorithms

Evaluation Measures	K-NN	Decision Tree	Random Forest	Naïve Bayes	SVMs	LSTM
Accuracy (%)	98.85	95.25	96.89	91.33	98.69	99.2
Precision (%)	94.29	88.54	98.82	100	100	99.52
Recall (%)	99	85	84	47	92	97.5
F1-score (%)	96.59	86.73	90.81	63.95	95.83	98.47

The comparative analysis of the results as presented in table 1 clearly indicates that LSTM obviously outperformed all other traditional Machine Learning algorithms in terms of F1-score and accuracy. On another perspective, better results are achieved in terms of precision, recall and accuracy with SVMs and K-NN with 3-fold cross validation when compared with the rest. Naïve Bayes accuracy is very high (91.33%), but F1-score is very low, indicating a bad approach.

X. RESEARCH PLAN

Week 3	Find paper related to this topic in google scholar
Week 4	Doing the draft of paper and the slide for the presentation
Week 5 – Week 8	Write introduction and literature review
Week 8 – Week 14	Finish the experiment to get the result
Week 14 until now	Finish the proposal and final slide

XI. CONCLUSION AND FUTURE DIRECTION

In my paper, some of the most popular traditional machine learning methods in email filter as well as a deep learning method have been performed and compared to find out the best approach. Deep Learning model yield better performance as a result. For future work, further enhancements can be done to improve the efficiency of the classifiers, for example, trying to apply different method in feature extraction for ML models or implementing new techniques such as such as paragraph vector and Glove for DL model. In another perspective, Naïve Bayes is a very unpredictable algorithm, hence should be investigated more. In summary, the practical use of this research is that it concluded the optimal performance of Deep Learning model, pave the way for comparative studies in Deep Learning models in the long run.

XII. REFERRENCES

[1] J.Clement, Spam: share of global email traffic 2014-2019, Dec 4, 2019

[2] Nucleus Research. (2007). Spam costing US Businesses $712 Per Employee.

[3] Statista. (2017). Global spam email traffic share 2014-2017.

[4] Wadi Hijawi, Hossam Faris, Ja'far Alquatawna, Ala' M. AI-Zoubi and Ibrahim AIjarah. "Improving Email Spam Detection Using Content Based Feature Engineering Approach", 22 October 2017

[5] W.A Awad and S.M.ELseuofi. "Machine learning methods for spam E-mail Classification", International Journal of Computer Science and Information Technology (IJCSIT), VOL 3, No 1, Feb 2011

[6] Er. Geetanjli Chawla and Ritu Saini. "Implementation of Improved KNN algorithm for Email Spam Detection", International Journal of Trend in Research and Development, Volum3(5), ISSN: 2394-9333, Sep- Oct 2016

[7] Houshamand Shirani-Mehr. "SMS Spam Detection using Machine Learning Approach", Published 2013

[8] Anju Radhakrishnan, Vaidhehi V. "Email Classification Using Machine Learning Algorithms", International Journal of Engineering and Technology (IJET), Vol 9 No 2 Apr-May 2017

[9] Uddin, Md Mohsin, et al. "Detecting Bengali Spam SMS Using Recurrent Neural Network." JCM 15.4 (2020): 325-331.

[10] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in Proc. International Conference on Machine Learning, February 2013, pp. 1310-1318.

[11] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[12]https://blog.nillsf.com/index.php/2020/05/23/confusion-matrix-accuracy-recall-precision-false-positive-rate-and-f-scores-explained