Detecting spam e-mails using stop word TF-IDF and stemming algorithm with Naïve Bayes classifier on the multicore GPU

Manjit Jaiswal, Sukriti Das, Khushboo
Department of Computer Science and Engineering, Guru Ghasidas Vishwavidyalaya, Bilaspur, India

Article Info

Article history:

Received May 23, 2020
Revised Jan 11, 2021
Accepted Jan 29, 2021

Keywords:

Google Colab
GPU
Naïve Bayes
NVIDIA
Porter's algorithm
Stemming
TF-IDF

ABSTRACT

A spam filter is a program which is used to identify unwanted emails and prevents those messages from getting into a user's mail. The study was focused on how the algorithms can be applied on a number of e-mails consisting of both ham and spam e-mails. First, the working principle and steps which are followed for implementation of stop words, TF-IDF and stemming algorithm on NVIDIA's Tesla P100 GPU are discussed and to verify the findings by executing of Naïve Bayes algorithm. After complete training and testing of the spam e-mails dataset taken from Kaggle by using the proposed method, we got a high training accuracy of 99.67% and got a testing accuracy of about 99.03% on the multicore GPU that boosted the speed of execution of training time period and testing time period which is improved of training and testing accuracy around 0.22% and 0.18% respectively when compared to that after applying only Naïve Bayes i.e. conventional method to the same dataset where we found training and testing accuracy to be 99.45% and 98.85% respectively. Also, we found that training time taken on GPU is 1.361 seconds which was about 1.49X faster than that taken on CPU which is 2.029 seconds. And the testing time taken on GPU is 1.978 seconds which was about 1.15X faster than that taken on CPU which is 2.280 seconds.

Corresponding Author:

Manjit Jaiswal
Department of Computer Science and Engineering
Guru Ghasidas Vishwavidyalaya
Bilaspur-495009, Chhattisgarh, India
Email: manjit.jaiswal222@gmail.com

1. INTRODUCTION

Spam emails are slowly growing since the 1990's. They may consist of links that are untrusted which initially gives impression to users that they are familiar whereas in reality they lead to the phishing web sites which may have malware [1]. The processing of structured or semi-structured data in all organizations is becoming very difficult these days as the data has been increased tremendously [2]. Based on the observations, we developed a word stemming [3] technique that can match words which both look alike and sound alike. Like other kinds of filtering programs, a spam filter searches for criteria on which its decisions are based on. This method is not very effective, it may remove legitimate e-mails (called false positives) and pass actual spam messages to increase the accuracy of e-mails spam detection. Better programs like the Bayesian filters or other heuristic filters try to identify spams through suspicious word, phrases, patterns or word frequencies [4].

Naive Bayes is more popular in commercial and open-source spam filters and is the conventional method [5]. This is because of its simplicity that makes them easy to implement and just need short training period or fast evaluation to filter spam e-mail. The use of computers for solving problems has been done for all areas of work. This is because computing is considered to be faster in solving problems than manual computation [6].

The main goal of this paper is to be enhanced the accuracy of e-mails spam detection. To boost up the processing speed we need to be increased the speedup, therefore GPU is generally used instead of CPU for processing large sets of data used generally for machine learning or deep learning [7, 8].

This paper is consisted by as follows: Section 1 is explaining the important of spam detection and some technique which have been used for spam detection. Section 2 is describing the literature review of Naïve Bayes i.e. conventional method and various technique used with Naïve Bayes as a preprocessing phase and their limitation. Section 3 is describing a new state of art e-mail spam filter to enhance the accuracy as far as possible. Section 4 is describing the process of proposed method with flow chart. Section 5 and 6 are showing the better accuracy result of proposed method over Naïve Bayes and overview of result as conclusion respectively.

2. RELATED WORK

Rusland *et al.* [5] worked on enhancing the conventional spam detection technique which uses Naïve Bayes algorithm for classification of spam mails. Renuka and Hamsapriya [9] analysed spam filtration. Bayesian filter works by testing the probability of various words appearing in legitimate, valid and spam mails and then classifying them based on those probabilities as spam or not [9]. Shabbir and Mithun [3] showed that if some sort of word stemming or word hashing technique is used that can extract the base or stem of a misspelled or modified word, then the efficiency of any content based spam filter can be significantly improved [10]. Atsumoto *et al.* in their paper describe the result of an empirical study based on two spam detection methods, namely support vector machines (SVMs) and naive Bayes classifier (NBC). The evaluation criteria include accuracy rate, recall, precision, miss rate, and false alarm rate [11]. Issac and Jap [12] in their paper describe Porter Stemmer algorithm for stripping the word to detect spam using Naive Bayes. It improves the filter's efficiency in terms of reducing the keyword searches and also generally improves the accuracy marginally. Saidani *et al.* [13] in their paper used a text semantic analysis to improve the accuracy of spam detection. This method shows better spam detection technique. Etaiwi and Naymat [14] in their paper analyzed the impact of applying different preprocessing steps to detect spam.

3. RESEARCH METHOD

To enhance the accuracy of detecting spam emails first we started data pre-processing phase as removing stop words, Porter's Stemming algorithm and TF-IDF before the Naive Bayes machine learning classifier.

3.1. Removing stop word

In order to improve the performance to detect the spam emails, stop word play crucial role to boost up searching operation by excluding some words from mails which would appear to be little worth. There is no single global stop word list used by natural language processing tools. The most common words such as 'is', 'at', 'on', 'the', and 'am'. are to be determining stop list by arrange the terms for their frequencies into decreasing order and then take the most frequent terms to be filter out as a stop words.

3.2. Porter's stemming algorithm

The preprocessing in the text mining consists of tokenization, stopword removal, stemming. The Porter Stemmer was developed by Martin Porter in the University of Cambridge in 1980 [15]. It is a method for removing the morphological and in flexional endings from English words. This stemmer is a linear step stemmer and it has five steps applying rules within each step [16]. Porter's Stemmer has advantages to its speed and accuracy due to the reduced the size of document. It is refine the data set to enhance the process of spam e-mails detection by stripping the suffix and produces a single stem. For example, teacher, teaches and teaching derived from the stem "teach" which needs to be considered as Teach for reducing the dimension of the word as in Table 1.

Table 1. Example of Porter's stemming

S.No.	Text	After Stemming
1	Teacher	Teach
2	Teaches	Teach
3	Teaching	Teach

3.3. TF-IDF

After stemming, it is very important to understand how a word is important in a document. TF-IDF is the technique to represent the textual information into a unit vector. TF-IDF is the term weighting method in preprocessing phase of structured dataset [17, 18]. The term "term frequency" is called TF to measure that how many times a term is present in a document as in (1).

$$TF_{i,j} = \frac{f_{i,j}}{\sum_k f_{i,j}} \tag{1}$$

Where $TF_{i,j}$ is the frequency of term i^{th} in document j, while $\sum_k f_{i,j}$ is the total words in document j [19].

The term frequency (*TF*) does not able to measure the importance of those terms that appear rarely across in a few documents of a repository. So for those terms which have rarely occur in a few corpuses of documents intended to measure how important document. To measure this one as in (2), the term *IDF* is called inverse document frequency [20].

$$IDF(i,D) = \log\left(\frac{|D|}{|d \in D : i \in d|}\right) \tag{2}$$

If the word does not exist in any document, denominator could be zero. So, to avoid ∞ which could not upper bound, we use $|d \in D : i \in d| + 1$ as a denominator in (2).

$$IDF(i,D) = \log\left(\frac{|D|}{|d \in D : i \in d| + 1}\right) \tag{3}$$

Where $|D|$ is the number of training document used and $|d \in D : i \in d|$ is the number of training document *d* that contains the term i.

To established the bound of IDF (lower and upper), we need to be normalized of (3). Therefore, we use $|D| + 1$ as numerator and add 1 in (3) to be set the lower bound of 1 as in (4) called the smooth IDF.

$$IDF(i,D) = \log\left(\frac{|D|+1}{|d \in D : i \in d| + 1}\right) + 1 \tag{4}$$

To calculate the *TF-IDF*, multiply (1) and (4) together as in (5). The resulting equation filter the common words and retain the important document.

$$TF - IDF(i,j,d,D) = \frac{f_{i,j}}{\sum_k f_{i,j}} * \log\left(\frac{|D|+1}{|d \in D : i \in d| + 1}\right) + 1 \tag{5}$$

3.4. Naïve Bayes

The Naive Bayes algorithm is a simple conventional probabilistic classifier which calculates a set of probabilities. It counts the frequency and combination of values in a given dataset [21]. In this classification, we have to find out the probability of a term in the spam mails or in the no spam mails then we decide the spam probability for that particular term [22, 23]. Based upon those two probabilities we can find the spam probability for that mail. Bayes theorem provides a way of calculating posterior probability.

$$P(c/x) = \frac{P(x/c)P(c)}{P(x)} \tag{6}$$

$$P(c/X) = P(x_1/c) \times P(x_2/c) \times \ldots \times P(x_n/c) \times P(c)$$

Here, P(c|x) is the posterior probability of class(target) given predictor(attribute), P(c) is the prior probability of class, P(x|c) is the likelihood which is the probability of predictor given class, P(x) is the prior probability of predictor.

4. PROPOSED WORK

4.1. Integration of TF-IDF and stemming algorithm with Naïve Bayes

The main idea of incorporating TF-IDF and Porter's Stemming algorithm in the implementation of Naïve Bayes algorithm for spam mail classification is to enhance the accuracy and processing speed of the algorithm as this method finds the valid terms via the stemming algorithm. The stop words are eliminated and the valid words are stemmed to their root form using the Porter's Stemming algorithm. Then the term frequencies and the relevance of those valid terms are used to form a vector table using the TF-IDF method in the pre-processing phase itself thereby further reducing the preprocessing time. The TF-IDF along with Naïve Bayes classifier is implemented for optimal results.

As we know, Naïve Bayes algorithm is a simple and efficient algorithm for categorization. In order to improve the performance of categorization of Naive Bayes algorithm in text categorization, based on TF-IDF attribute weighting is used. In Figure 1 we depict how our model works by incorporating four methods, namely removing stop words, Stemming, TF-IDF and Naïve Bayes. The process starts by receiving the mail in the user's inbox after which the mail form is checked for repetitive words and stop words which are mostly irrelevant for prediction. The words which can be changes into their root form are stemmed using the stemming algorithm following which the TF-IDF [24] preprocessing is applied on the stemmed words. After the vector table is created, it is checked for being spam using the Naïve Bayes classifier. If it is found to be spam, it is rejected; other the mail is delivered to the user's inbox.

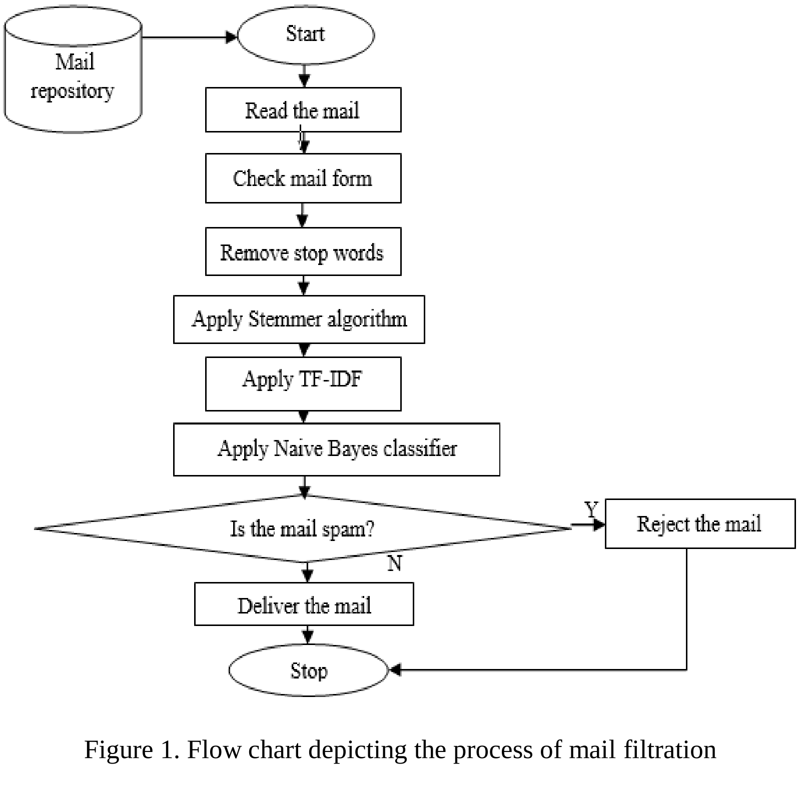
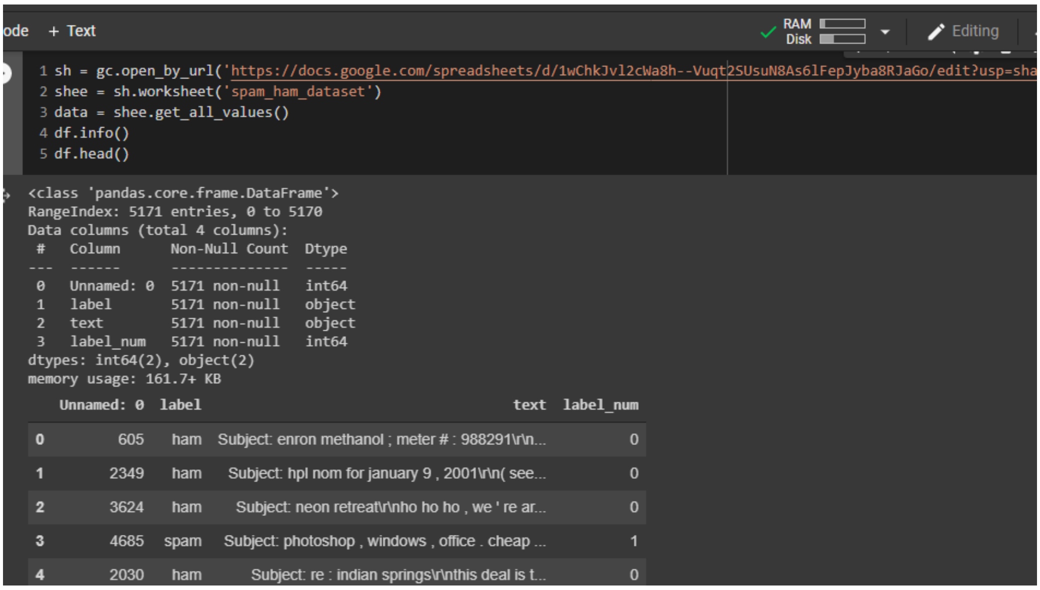


Figure 1. Flow chart depicting the process of mail filtration

5. IMPLEMENTATION AND EXPERIMENTAL RESULT

5.1. Dataset analysis

The model is trained and tested on the Kaggle [25] spam e-mails dataset. To achieve this, user email data has been collected in the *.csv* format and marked them as either spam or ham. It has 5171 emails which are labelled and are collected for email spam research. Label Encoder labels with values 0 (ham) and 1 (spam) as in Figure 2. Out of the entire dataset, 80% is used for the purpose of training while the rest 20% is used for testing.



```
sh = gc.open_by_url('https://docs.google.com/spreadsheets/d/1wChkJvl2cWa8h--Vuqt2SUsuN8As6lFepJyba8RJaGo/edit?usp=sha
shee = sh.worksheet('spam_ham_dataset')
data = shee.get_all_values()
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5171 entries, 0 to 5170
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  5171 non-null   int64
 1   label       5171 non-null   object
 2   text        5171 non-null   object
 3   label_num   5171 non-null   int64
dtypes: int64(2), object(2)
memory usage: 161.7+ KB
```

	Unnamed: 0	label	text	label_num
0	605	ham	Subject: enron methanol ; meter # : 988291\r\n...	0
1	2349	ham	Subject: hpl nom for january 9 , 2001\r\n(see...	0
2	3624	ham	Subject: neon retreat\r\nho ho ho , we ' re ar...	0
3	4685	spam	Subject: photoshop , windows , office . cheap ...	1
4	2030	ham	Subject: re : indian springs\r\nthis deal is t...	0

Figure 2. Dataset ‘spam_ham dataset’

5.2. Experimental setup

Colaboratory [26], or “Colab” for short, is a product of Google Research. Colab is a hosted Jupyter notebook service that requires no external setup for use, while providing us with free access to various computing resources including GPUs as in Table 2.

Table 2. Hardware and software specification of Colab

S. No.	Specification type	Description
1.	GPU	Tesla P100-PCIE-16 GB
2.	CPU	Intel(R) Xeon(R) CPU @2.30 GHz
3.	RAM	~12.72 GB
4.	Disk	~68.4 GB
5.	IDE	Colab Notebook
6.	Programing language	Python 3.6

5.3. Simulation and result

Various experiments are applied on the dataset which were based on natural language processing (NLP) concepts like label encoding, tokenization, stemming, stop word removal, generating features. We taught our program what a spam email looks like and what non-spam emails looks like. The formula show the evaluation measures as in (5), (6), (7) and (8).

$$Acc = \frac{TN + TP}{TP + FN + FP + TN} \tag{7}$$

$$F = \frac{2PR}{P + R} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$P = \frac{TP}{TP + FP} \tag{10}$$

Accuracy (Acc): Percentage of correctly identified spam and not spam message
- F-measure (F): Weighted average of precision and recall
- Recall (R): Percentage spam mails managed to block
- Precision (P): Percentage of correct message for spam mail

5.3.1. Training and testing evaluation

The 80% of the total labelled dataset is first used for training the system based on which it learns how to classify the mails that will be fetched to it for the purpose of testing the remaining 20% of the dataset is then fetched to the trained system and on the basis of the trained data, the system is tested for accuracy and other factors and the results of Naïve Bayes (conventional method) and proposed method on training and testing dataset are in Tables 3 and 4. In Figures 3, 4, 5 and 6 the graph depicts the results of Tables 3 and 4.

Table 3. Table for Naïve Bayes (conventional) filtering technique

S.No.	Evaluation Parameters	Percentage (%)(Training)	Percentage (%)(Testing)
1	Accuracy	99.45	98.85
2	F1 Score	97.19	95.60
3	Recall	95.24	94.05
4	Precision	98.67	97.21

Table 4. Table for proposed method

S.No.	Evaluation Parameters	Percentage (%)(Training)	Percentage (%)(Testing)
1	Accuracy	99.67	99.03
2	F1 Score	100.00	99.00
3	Recall	100.00	99.00
4	Precision	100.00	100.00

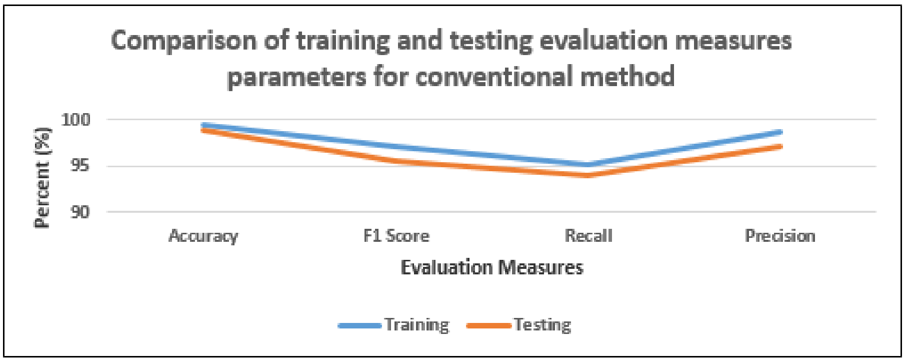


Figure 3. Accuracy curve of Naïve Bayes (conventional method) for training and testing dataset

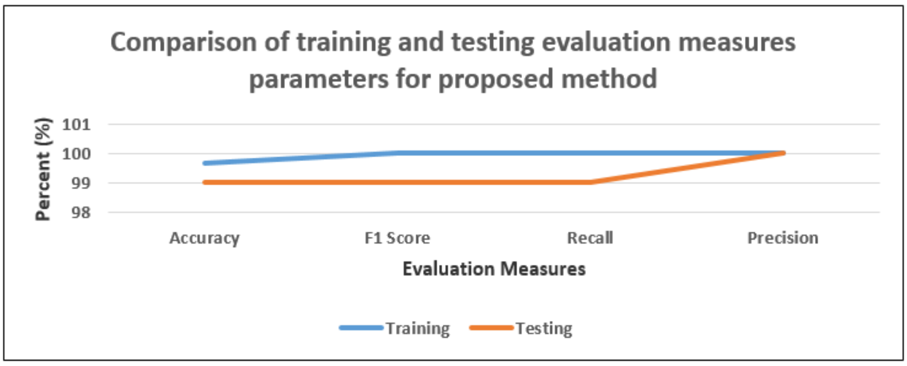


Figure 4. Accuracy curve of proposed method for training and testing dataset

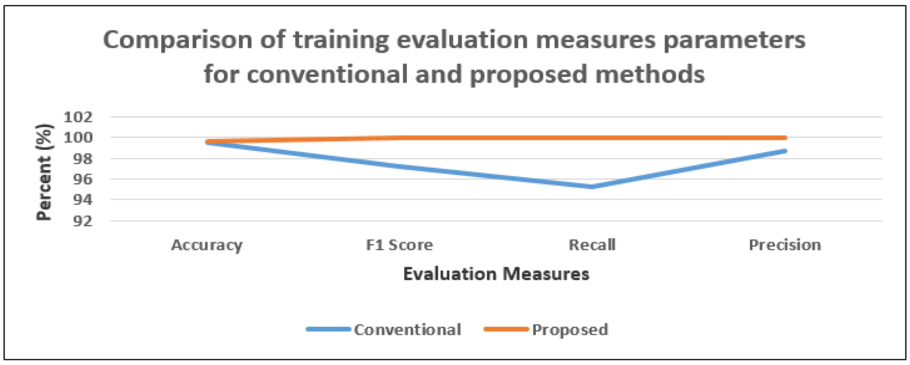


Figure 5. Accuracy curve of proposed method over Naïve Bayes (conventional method) for training dataset

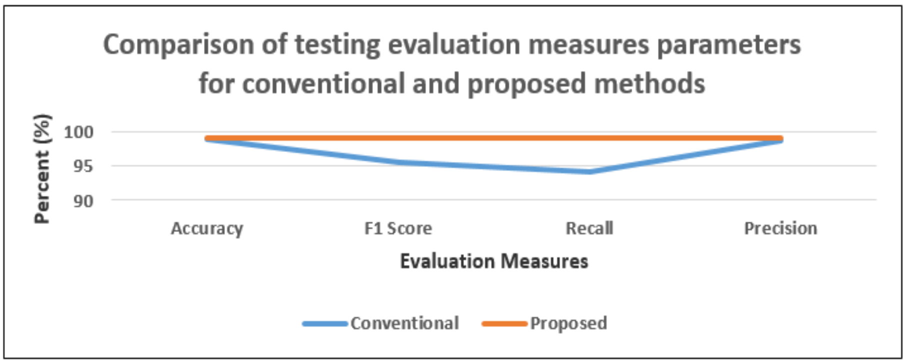


Figure 6. Accuracy curve of proposed method over Naïve Bayes (conventional method) for testing dataset

6. CONCLUSION

In this paper, the preprocessing method of stop words, TF-IDF and stemming algorithm is discussed in detection of spam e-mails using the Naïve Bayes classifier. The study is focused on how the algorithms can be applied on a number of e-mails consisting of both ham and spam e-mails. First, the working principle and steps which should be followed for implementation of stop words, TF-IDF and stemming algorithm are discussed. From the Kaggle spam e-mails dataset, the training data showed an accuracy of 99.67% while the test evaluation gives 99.03% accuracy which is quite accurate compared to the only Naïve Bayes (conventional method) classifier where accuracy of training data is 99.45% while accuracy of testing data is 98.85%. We also found that, as the proposed algorithm was executed with high speed on multi-core GPU of Google Colab environment. The time taken on the CPU and GPU for the training dataset is 2.029 seconds and 1.361 seconds respectively while that on the testing dataset is 2.280 seconds and 1.978 seconds respectively.

REFERENCES

[1] S. B. Nadaf, and A. D. Gujar, “A Survey Paper on Spam Mail Detection Using RFD,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 4, no. 1, pp. 46-48, 2016.
[2] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25-29, Jul. 2018.
[3] S. Ahmed and F. Mithun, “Word Stemming to Enhance Spam Filtering,” *CEAS*, 2004.
[4] Rekha, and S. Negi, “A Review on Different Spam Detection Approaches,” *International Journal of Engineering Trends and Technology (IJETT)*, vol. 11, no. 6, pp. 315-318, 2014.
[5] N. F. Rusland, N. Wahid, S. Kasim, H. Hafit, “Analysis of Naive Bayes algorithm for email spam filtering across multiple datasets,” *IOP conference series: materials science and engineering*, vol. 226, no. 1, 2017, Act. no. 012091.
[6] Islamiyah, Nataniel, Dengen, E. Maria, “Naïve Bayes Classifiers For Tweet Sentiment Analysis Using GPU,” *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 5C, 2019.
[7] S. K. Sahay and M. Chaudhari, “An Efficient Detection of Malware by Naive Bayes Classifier Using GPGPU,” *Advances in Computer Communication and Computational Sciences*, Springer, Singapore, pp. 255-262, 2019.

[8] M. Jaiswal, A. Sahu, Md T. Zafar, “Effectively Diagnosing Malaria by Optimizing the Hyperparameters of CNN using Genetic Algorithm on the Multi core GPU,” *International Journal of Recent Technology and Engineering (IJRTE),* vol. 8, no. 6, pp. 2983-2991, 2020.

[9] D. K. Renuka and T. Hamsapriya, “Email classification for Spam Detection using Word Stemming,” *International Journal of Computer Applications,* vol. 975, pp. 8887, 2010.

[10] J. R. Méndez, E. L. Iglesias, F. Fdez-Riverola, F. Díaz, J. M. Corchado, “Tokenising, Stemming and Stopword Removal on Anti-spam Filtering Domain,” *Springer, Berlin, Heidelberg,* 2006.

[11] R. Matsumoto, D. Zhang and M. Lu, “Some empirical results on two spam detection methods,*”Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration,* 2004, pp.198-203.

[12] B. Issac and W. J. Jap, “Implementing spam detection using Bayesian and Porter Stemmer keyword stripping approaches,” *TENCON 2009-2009 IEEE Region 10 Conference,* 2009, pp. 1-5.

[13] N. Saidani, K. Adi, M. S. Allili, “A Semantic-Based Classification Approach for an Enhanced Spam Detection,” *Computers and Security,*vol. 94, 2020, Act. no. 101716.

[14] W. Etaiwi and G. Naymat, “The Impact of applying Different Preprocessing Steps on Review Spam Detection,”*Procedia computer science,* vol. 113, pp. 273-279, 2017.

[15] Martin F. Porter, “An algorithm for suffix stripping,” *program,* vol. 40, pp. 211-218, 1980.

[16] B. Pragna, M. Rama Bai, “Spam Detection using NLP Techniques,” *International Journal of Recent Technology and Engineering (IJRTE),* vol. 8, no. 2S11, pp. 2423-2426, 2019.

[17] Asmeeta Mali, “Spam Detection Using Baysian with Pattren Discovery,” *International Journal of Recent Technology and Engineering (IJRTE),*vol. 2, no. 3, pp. 139-143, 2013.

[18] F. Jiang, Z. Zhang, P. Chen, Y. Lio, “Naïve Bayes Text Categorization Algorithm Based on TF-IDF Attribute Weighting,” *CSAI '18, Shenzhen, China© 2018 Association for Computing Machinery (ACM),* Dec. 2018.

[19] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, O. Rusli, Rudy, “News article text classification in Indonesian language,” *2nd Int. Conf. on Comp. Sci. and Comp. Intell., ICCSCI,* Bali, Indonesia, Oct. 2017.

[20] Trstenjak, B., Mikac, S., and Donko, D., “KNN with TF-IDF based framework for text categorization,” in *Procedia Engineering,* vol. 69, pp. 1356-1364, 2014.

[21] Bafna, P., Pramod, D., and Vaidya, A., “Document clustering:TF-IDF approach,” *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT),* Chennai, 2016, pp. 61-66.

[22] Rathi, M. and Pareek, V., “Spam Mail Detection through Data Mining A Comparative Performance Analysis,” *I.J. Modern Education and Computer Science,* vol. 5, no. 12, pp. 31-39, 2013.

[23] U. Bhardwaj and P. Sharma, “Email Spam Detection using Ensemble Methods,” *International Journal of Recent Technology and Engineering (IJRTE),*vol. 8, no. 3, pp. 4148-4153, Sep. 2019.

[24] “TF-IDF: Vector representation of Text,” *Learn Natural Language Processing: From Beginner to Expert*. [Online], Avalaible: https://www.commonlounge.com/discussion/99e86c9c15bb4d23a30b111b23e7b7b1

[25] V. Garnepudi, “Spam Mails Dataset,” *kaggle,* [Online], Available: https://www.kaggle.com/venky73/spam-mails-dataset

[26] “Colaboratory,” [Online], Available: https://research.google.com/colaboratory/faq.html

BIOGRAPHIES OF AUTHORS

Manjit Jaiswal is currently working as an Assistant professor in Computer Science and Engineering Department, School of Studies in Engineering and Technology, Guru Ghasidas Vishwavidyalaya, A Central University Bilaspur,Chhattisgarh,India. He received Master of Technology(M.Tech.) degree in 2012 from Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India. He has published more than 12 papers in reputed journals and conference like IEEE, UGC approved etc. His research interest fields are Parallel Computing,Machine Learning and Deep Learning.He has more than 8 years of teaching experience.

Sukriti Das is currently pursuing Bachelor of Technology in Computer Science and Engineering from School of Studies in Engineering and Technology, Guru GhasidasVishwavidyalaya, Bilaspur, India. She is currently studying in final year. Her research interest fields are Machine Learnisng, Computer Vision, and Network Security.

Khushboo is currently pursuing Bachelor of Technology in Computer Science and Engineering from School of Studies in Engineering and Technology, Guru GhasidasVishwavidyalaya, Bilaspur, India. She is currently studying in final year. Her research interest fields are Machine Learning, Computer Vision, and Network Security.