International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Optimization of Naïve Bayes Classifier for Spam E-Mail Detection

Sai Charan Lanka [1], Kodali Pujita [2], Kommana Akhila [3], Shayan Mondal [4], P. Vidya Sagar [5], Suneetha Bulla [6]

Submitted: 29/01/2024 **Revised:** 07/03/2024 **Accepted:** 15/03/2024

Abstract: E-Mail, a popular and official communication platform, widely used method of communication that facilitates exchange of information between individuals or organizations in a convenient and efficient way to send and receive any kind of information instantly from any corner of the world. But, due to this drastic growth of the usage of E-Mail, spammers are using this platform to perform frauds through mails that mails are known as Spam Mails. Spam Mails can be detected and identified using various approaches. Among those approaches Machine Learning is widely used. In Machine Learning, Naïve Bayes Classifier stands out with the highest accuracy this is due to "Low False Positive Error Rate". Although Naïve Bayes Classifier gives us the best accuracy among all other Machine Learning models, we can still optimize it to give a better accuracy.

Keywords: *information, facilitates, Spam Mails, Machine Learning, optimize.*

1. Introduction

Spam E-Mails, a rising problem across the electronic communication platform. This spam E-mails can be defined as "Usage of E-Mails in order to send mails that are not at all useful or informative to a large group of recipients using E-mail as a platform". Due to the drastic growth of E-Mail communication, spammers are using this platform to perform scams and frauds by sending some mails containing URLs which consists malicious viruses which will breach into recipient's devices resulting in the compromise of the sensitive data of the user.

Although many popular E-Mail domain providers provide some in-built Spam Mail detectors, spammers with better technology can overcome those detectors.

Hence Machine Learning can be used and incorporated in Spam Detection. There are a wide range of Machine Learning Algorithms, but Naïve Bayes Classification Algorithm gives us a better performance with the highest accuracy score. This is due to "Low False Positive Error for Spam Detection Rates".

1,2,3,4 Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.
5,6 Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India.
1 charanlanka6@gmail.com
2 kodalipujita@gmail.com
3 akhila61663@gmail.com
4 shayanmondal26@gmail.com
5 pvsagar20@gmail.com
6 suneethabulla@kluniversity.in

2. Literature Review

According to SC Lanka et al., [1] said that email and how it's a popular way for people to communicate online. However, because email usage is expanding, so is the number of spam emails. Spam emails are unsolicited and sometimes harmful messages sent to an extensive list of individuals. This can be an issue because some spam emails may attempt to deceive individuals into disclosing personal information. To tackle this issue, the writer suggests using a technology called "machine learning" to help identify and filter out spam emails. Machine learning is a way for computers to learn from data and make decisions without being explicitly programmed. By using machine learning, researchers can find the best computer program (algorithm) that can accurately detect and block spam emails with the highest accuracy possible. This will help protect people from falling for scams and frauds that might come through spam emails.

According to Renuka et al., [2] They proposed system uses electronic mail (Email) to exchange documents over the Internet. However, it faces the issue of spam, which is harmful and irrelevant data sent without request. To tackle this problem, the system employs a spam classification method called Naive Bayes classifier. This classifier is based on conditional probability and works well for complex classification tasks. Additionally, the system utilizes a feature selection technique known as hybrid Ant Colony Optimization, which improves efficiency and produces effective results for the proposed system, as described in the paper.

According to F Hossian et al., [3] They developed a model for categorizing emails as spam or ham (non-spam). We used DBSCAN and Isolation Forest algorithms to discover extreme values outside of a certain range. The model was created utilizing both deep learning and machine learning approaches in order to conduct a comparative study. Finally, we developed an ensemble approach for combining the results of many classifiers.

According to Agarwal et al.,[4] They said that nowadays, email communication has become a cheap and easy option for official users due to widespread internet access. However, the ease of email usage has led to the problem of spam emails, which are unwanted and useless bulk messages sent to others. These spam emails not only consume mailbox memory but also make it difficult to find useful information. The paper proposes a solution to this problem by integrating machine learning with the Naive Bayes approach, computational intelligence with Particle Swarm Optimization, and artificial intelligence with Particle Swarm Optimization (PSO). PSO successfully optimizes the parameters of the Naive Bayes algorithm, which is used to determine if an email is spam or not. The suggested method is assessed using performance metrics on the Ling spam dataset. The results suggest that employing PSO instead of the Naive Bayes method alone enhances performance.

According to P Sharma et al., [5] They introduce a hybrid bagging approach for detecting spam emails, using Nave Bayes and J48 are two machine learning methods. Each algorithm is fed a separate set of data from the dataset. Three studies are compared in terms of performance metrics. The first two trials employ individual Nave Bayes and J48 algorithms, respectively, while the third employs the hybrid bagged technique to create the suggested SMD system. The total accuracy of the hybrid bagged approach based SMD system is 87.5%.

According to Faris et al., [6] They present a dual-stage spam filtering approach. The first stage involves the use of PSO for Wrapper Feature Selection, which aids in the selection of the most relevant characteristics from a huge range of measured data. The selected characteristics from the first stage are used to build a spam filtering model based on Random Forest in the second step. Our trials on real-world spam data show that our strategy beats five popular machine learning algorithms in the literature. Furthermore, we assess our proposed spam filtering technique using four cost functions, and the findings show that the PSO-based Wrapper with Random Forest is an effective spam detection strategy.

According to Z Hassain et al.,[7] They present a feature selection method for detecting e-mail spam, which combines optimization algorithms and machine learning classifiers. Usage of the Binary Whale Optimization and Binary Grey Wolf Optimization algorithms for feature selection, as well as the K-Nearest Neighbor and Fuzzy K-Nearest Neighbor techniques as classifiers, in this study. The proposed technique was evaluated on the "SPAMBASE" dataset from the UCI Machine Learning Repositories to determine how accurate it is, and on this collection of data, the experimental results reveal an astounding accuracy performance of 97.6%. These findings demonstrate that our approach outperforms other methods, making it a suitable and effective solution for e-mail spam detection.

According to WA Awad et al.,[8] spam Email categorization, they present an overview of different common machine learning algorithms. The methods are described in detail, and their performance is evaluated using the Spam Assassin spam corpus dataset.

According to H Bhuiyan el at.,[9] presents a study of several email spam filtering systems that employ Naive Bayes, SVM, K-Nearest Neighbor, Bayes Additive Regression, KNN Tree, and rules are examples of Machine Learning Techniques. The emphasis is on categorizing, evaluating, and comparing different systems in order to summarize their overall accuracy rates.

According to W Peng et al.,[10] They devised an innovative way for improving the Naive Bayes Spam Filter's accuracy, allowing it to identify text changes and properly categorize emails as spam or ham. Furthermore, a correlation between email length and spam score was identified, demonstrating that Bayesian Poisoning, a disputed phrase, is an actual event used by spammers.

3. Methodology

The Methodology obtained to implement Naïve Bayes Classifier is the standard “Machine Learning Life Cycle”. This life cycle is a standard approach to implement any Machine Learning Algorithm. It is a five-step process.

This is a step-by-step procedure in which each step must be followed exactly by the next phase.

3.1. Life Cycle of Machine Learning

The below mentioned stages are the one in which machine goes through.

3.1.1. Data Gathering:

For any Machine Learning Model, data is an important pillar. This step of Machine Learning Life Cycle involves the collection of appropriate data according to our requirement and in a proper format.

3.1.2. Data Preparation:

The data collected in the previous step must be prepared in a format such that the data is readable and analysable. This involves many strategies such as eliminating null values, outliers, and so on.

3.1.3. Exploratory Data Analysis:

After preparing the data, we will analyse it utilizing visualization techniques to acquire insights from the processed data so that we may have a better knowledge of the target variable. Also, we will divide the dataset into testing and training datasets for the purpose of model testing and model training respectively.

3.1.4. Data Modelling:

After gaining some valuable insights from the Exploratory Data Analysis step, here we will model the data using training dataset using a specific algorithm (in this case we will apply Naïve Bayes Classification Algorithm).

3.1.5. Model Evaluation:

The model therefore modelled using training dataset needs to be tested using testing dataset to test the model’s performance.

3.2. Dataset:

For any Machine Learning model, we require data for modelling purposes. Hence, we have gone through various online sources for collection of the data. Finally, we collected the data from the famous data repository called “Kaggle”.

The dataset name is “Spam.csv”, it contains 2 columns and around 5500 records in it. The first column consists of the group of words randomly taken from the real-time mails and second column consists of the class of the mail corresponding to the word with the class label spam and ham.

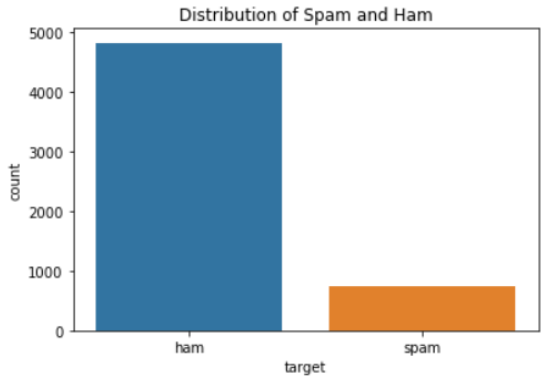

Fig. 1. Dataset's Spam Label and Ham (Non-Spam) Label distribution.

3.3. Classification:

Spam Based E-Mail Detection employs classification since the goal variables, Spam and Ham are categorical variables. As a result, categorization technique is implemented for this use-case.

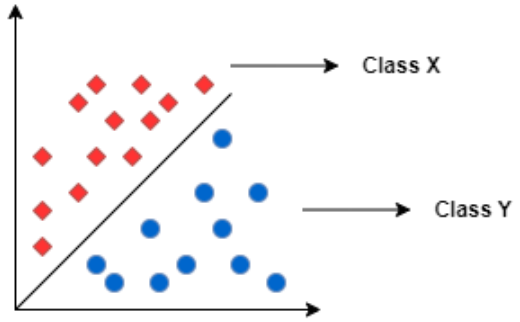

Fig. 2. Classification

Classification is a Machine Learning approach, used when the target variable is categorical in nature, and it classifies the target variable into a specific class label.
For Example: Classification of Gender.

For the use-case “Spam E-Mail Detection”, Naïve Bayes Classification Algorithm is a widely used technique as it performs with a good accuracy score among all other Machine Learning Techniques. This is due to this algorithm exhibits “Low False Positive Error Rate”. This differs the Naïve Bayes Classification Algorithm from other Machine Learning Algorithms.

3.3.1. Naïve Bayes Classification Algorithm:

Naive Bayes is a basic, yet effective classification technique based on Bayes' theorem and the assumption of feature conditional independence.

By integrating prior probabilities with the probabilities of the characteristics occurring in each class, the Naive Bayes classifier determines the likelihood of a given instance belonging to each class.

3.3.2. Bayes Theorem:

It is used to determine the likelihood of a hypothesis given past knowledge. Conditional probability determines it.

$$P\left(\frac{A}{B}\right) = P\left(\frac{B}{A}\right) * \frac{P(A)}{P(B)}$$

Where,

P(A|B) denotes Posterior Probability: The likelihood of hypothesis A on observed event B.

P(B|A) denotes Likelihood Probability: The likelihood that a hypothesis is true based on the evidence.

3.4. Optimization Techniques:

To optimize the Naïve Bayes Classification Algorithm, we have so many techniques some of them are K-Cross Fold Validation, Regularization, Handling Imbalance Data and so on.

K-Cross Fold Validation, L1 Regularization, L2 Regularization and Handling Imbalance Data are the techniques that are being implemented in this paper to optimize Naïve Bayes Classification Algorithm.

3.4.1. K-Cross Fold Validation:

It is a Machine Learning method for assessing a model's performance on unseen data to enhance the model's overall performance.

It entails dividing the given dataset into numerous subsets or folds, utilizing one of these folds as the testing dataset and the rest as the training dataset to build and evaluate the model. This process is repeated iteratively until all the folds are completed.

This technique works by dividing the dataset into multiple subsets and for an iteration, one of those subsets is taken as the validation dataset and the rest of the subsets as training dataset then, the model is trained and evaluated. This process is repeated iteratively until each data subset is taken as validation dataset. Here k represents the number of subsets or folds.

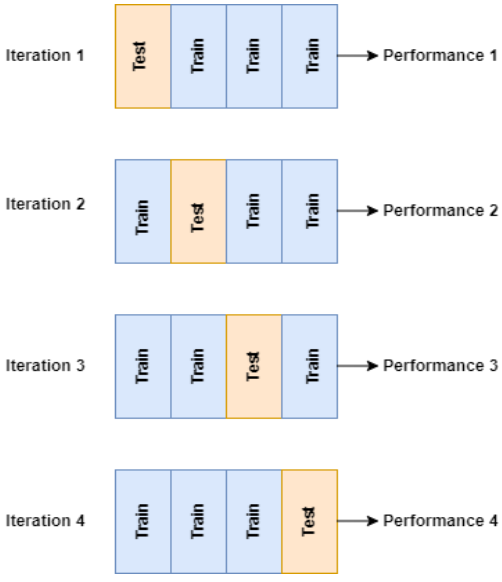

Fig 3. K- Cross Fold Validation

The overall performance can be calculated by computing the average of the Performance obtained at each iteration.

$$\text{Overall Performane} = 1/k \sum_{i=1}^{k} Performance$$

The primary goal of K-Cross Fold Validation is to prevent model overfitting, which occurs when a model is well-trained on a training dataset but fails to perform effectively on untrained data.

3.5. Regularization:

It is a Machine Learning approach used to prevent a model from overfitting. It arises when the model is very sophisticated and overly fits the training dataset, resulting in poor generalization to unseen data.

When the variance of the data is too high, overfitting occurs. Regularization is accomplished by introducing a penalty term or a complexity term into a complex model.

3.5.1. L1 – Regularization:

It is a regularization technique used to prevent a model from overfitting. It introduces a penalty term that is proportional to the absolute value of the loss function's weights. As a result, the weights are sparse, with some weights being exactly zero during training. This helps in the reduction of the complexity of the model. The strength of regularization is controlled by a hyperparameter lambda, which determines amount of shrinkage of weights. It is also known as “Lasso Regularization”.

$$\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_p - y_o)^2 + \lambda \sum_{i=1}^{n} |m_i|$$

Where, λ is the Penalty or Complexity Term.

3.5.2. L2 – Regularization:

It is a regularization technique used to prevent a model from overfitting. It introduces a penalty component that is proportional to the square of the weights in the loss function. As a result, the weights are low but not quite zero. This helps in the reduction of the complexity of the model.
The strength of regularization is controlled by a hyperparameter lambda, which determines amount of shrinkage of weights. It is also known as “Ridge Regularization”.

$$\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_p - y_o)^2 + \lambda \sum_{i=1}^{n} m_i^2$$

Where, λ is the Penalty or Complexity Term.

3.6. Handling Imbalance Data:

Imbalanced data occurs when the labels of the target variable are not distributed equally. Handling Imbalance Data is crucial to prevent biased models that tend to Favor the majority class.

The commonly used Handling techniques are.

- Oversampling
- Undersampling

3.6.1. Oversampling:

It entails increasing the proportion of samples from the majority class to the proportion of samples from the minority class.

```
Number of samples in each class before Oversampling:
Spam (positive class): 514
Non-Spam (negative class): 3386

Number of samples in each class after Oversampling:
Spam (positive class): 3386
Non-Spam (negative class): 3386
```

Synthetic Minority Over-sampling approach (SMOTE) is a prominent over-sampling approach. In this technique it generates synthetic samples of minority class based on characteristics of existing samples.

3.6.2. Under Sampling:

It entails lowering the proportion of samples from the majority class to the proportion of samples from the minority class.

```
Number of samples in each class before Undersampling:
Spam (positive class): 514
Non-Spam (negative class): 3386

Number of samples in each class after Undersampling:
Spam (positive class): 514
Non-Spam (negative class): 514
```

This strategy may aid in class distribution balancing, but it may result in the loss of useful information existing in the majority class.

4. Algorithm And Implementation

4.1. Algorithm:

Step 1: Using Pandas Python Library, upload the d dataset.
Step 2: Determine whether the dataset is compatible with the relevant encoding format.
2.1: If the dataset is compatible with the specified encoding format, go to Step 4.
2.2: If the dataset does is not compatible the desired encoding, go to Step 3.
Step 3: Step 1 should be repeated after changing the encoding format.
Step 4: Data Pre-Processing to be performed.
4.1: Look for values that are missing and try to fill them in or eliminate them.
4.2: Remove unnecessary columns.
4.3: Remove the matching data.
Step 5: Data that has been previously processed should be divided:
5.1: Separate the data into training and testing sets.
5.2: In a Seventy: Thirty ratio, divide the data.
Step 6: Model Development:
6.1: Incorporate the data into the Naive Bayes model.
6.2: Using testing data, make projections based on the learned model.
Step 7: Assess the Model's Performance:

7.1: Using the trained model's predictions, demonstrate the way each trained classification model performs on the testing data using performance metrics.

4.2. Flow Chart:

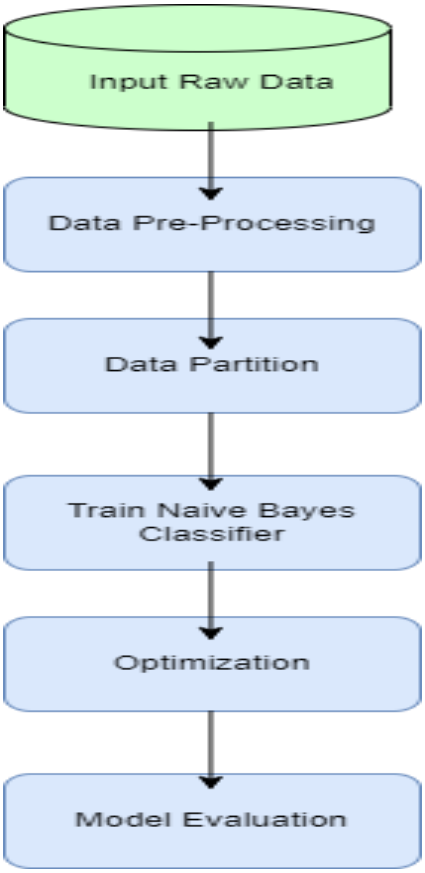

Fig 4. Flow Chart

4.3. Implementation

The Naive Bayes Classification Algorithm is implemented using the Jupyter Notebook IDE followed by the optimization techniques used to optimize the existing algorithm.

In terms of the collection of data required for the modelling, searched a lot of online data repositories but, finally found the required dataset in a well-known online data repository "Kaggle". The dataset name is "Spam.csv".

The present paper's implementation is entirely in Python with the help of in-built libraries like NumPy, Keras, Matplotlib, seaborn etc.

The implementation starts by uploading the dataset in a supported format of encoding (UTF-8). Then the dataset is processed and prepared as per the requirement. The model is then partitioned into training and validation datasets, and the model is trained using the Nave Bayes Classification Algorithm on the training dataset. After the modelling the trained model is evaluated using validation dataset and unseen data. Then, we apply optimization techniques to the pre-trained model and evaluate each optimization technique along with the trained model.

Then, we perform comparative analysis with the standard Naïve Bayes Classifier to the Optimization Techniques performed and conclude which Optimization Technique is better among others.

5. Results

The model is trained and evaluated on the training and validation datasets, respectively, using the Optimization Techniques used. Then, the model is compiled with a number of epochs. Then the model's solo performance and model's performance with Optimization Techniques are evaluated against validation dataset.

The performance of the models is calculated by the means of performance metrics i.e., Accuracy, Precision and F1-Score.

The table "Table I" displays the performance of the Standard Nave Bayes Classification Algorithm in detecting spam e-mail.

Table 1. Performance of Standard Nave Bayes Classification Algorithm

Model	*Accuracy*	*Precision*	*F1-Score*
Naïve Bayes Classification Algorithm	98.253	94.003	94.000

The table "Table II" provides a comparison of the performance of all Optimization Techniques which are applied to Standard Naïve Bayes Classification Algorithm. From the below table "Table II", the technique "K-Cross Fold Validation" gives a better accuracy than other techniques.

Table 2. Performance of Various Optimization Techniques

Optimization Technique	*Accuracy*	*Precision*	*F1-Score*
K-Cross Fold Validation	98.445	96.001	94.323
L1- Regularization	98.262	96.292	92.650
L2 - Regularization	98.026	97.169	92.584
Oversampling	98.266	93.965	93.763
Undersampling	97.009	86.166	89.711

Among all the Optimization Techniques, K-Cross Fold Validation optimizes the best of the Standard Naïve Bayes Classification Algorithm followed by Oversampling, L1-Regularization, L2-Regularization, and Undersampling.

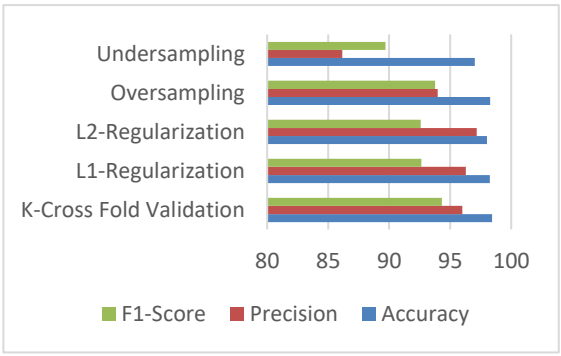

Fig 5. Comparative Analysis of the Standard Naïve Bayes Classifier with Optimization Techniques

6. Conclusion

In comparison to other Machine Learning Methods, the Nave Bayes Classification Algorithm consistently provides the highest

accuracy in the context of Spam Detection due to its unique property of “Low False Positive Error Rate”. Optimization of this Classification Algorithm is a bit difficult due to Independence Assumption property of Naïve Bayes Classification Algorithm and it may lead to overfitting of the model. Hence, we used the regularization techniques in order to optimize the model. Out of all the Optimization Techniques used “K-Cross Fold Validation” outstands optimizing the best of the standard version of Naïve Bayes Classification Algorithm.

Although this research implementation is content based i.e., this approach can be used when we have only the content present in the mail. At the present point of time the research work done is bounded and there is much scope of improvement.

7. References

.

[1] Lanka, S. C., Akhila, K., Pujita, K., Sagar, P. V., Mondal, S., & Bulla, S. (2023, March). Spam based Email Identification and Detection using Machine Learning Techniques. In 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) (pp. 69-74). IEEE.

[2] Renuka, D. K., Visalakshi, P., & Sankar, T. J. I. J. C. A. (2015). Improving E-mail spam classification using ant colony optimization algorithm. Int. J. Comput. Appl, 22, 22-26.

[3] Hossain, F., Uddin, M. N., & Halder, R. K. (2021, April). Analysis of optimized machine learning and deep learning techniques for spam detection. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) (pp. 1-7). IEEE.

[4] Agarwal, K., & Kumar, T. (2018, June). Email spam detection using integrated approach of Naïve Bayes and particle swarm optimization. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 685-690). IEEE.

[5] Sharma, P., & Bhardwaj, U. (2018). Machine Learning based Spam E-Mail Detection. International Journal of Intelligent Engineering & Systems, 11(3).

[6] Faris, H., Aljarah, I., & Al-Shboul, B. (2016). A hybrid approach based on particle swarm optimization and random forests for e-mail spam filtering. In Computational Collective Intelligence: 8th International Conference, ICCCI 2016, Halkidiki, Greece, September 28-30, 2016. Proceedings, Part I 8 (pp. 498-508). Springer International Publishing.

[7] Hassani, Z., Hajihashemi, V., Borna, K., & Sahraei Dehmajnoonie, I. (2020). A classification method for E-mail spam using a hybrid approach for feature selection optimization. Journal of Sciences, Islamic Republic of Iran, 31(2), 165-173.

[8] Awad, W. A., & ELseuofi, S. M. (2011). Machine learning methods for spam e-mail classification. International Journal of Computer Science & Information Technology (IJCSIT), 3(1), 173-184.

[9] Bhuiyan, H., Ashiquzzaman, A., Juthi, T. I., Biswas, S., & Ara, J. (2018). A survey of existing e-mail spam filtering methods considering machine learning techniques. Global Journal of Computer Science and Technology, 18(2), 20-29.

[10] Peng, W., Huang, L., Jia, J., & Ingram, E. (2018, August). Enhancing the naive bayes spam filter through intelligent text modification detection. In 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE) (pp. 849-854). IEEE.

[11] Al-Rawashdeh, G., Mamat, R., & Abd Rahim, N. H. B. (2019). Hybrid water cycle optimization algorithm with simulated annealing for spam e-mail detection. IEEE Access, 7, 143721-143734.

[12] C. M. Shaik, N. M. Penumaka, S. K. Abbireddy, V. Kumar and S. S. Aravinth, "Bi-LSTM and Conventional Classifiers for Email Spam Filtering," 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2023, pp. 1350-1355, doi: 10.1109/ICAIS56108.2023.10073776.

[13] S. Gibson, B. Issac, L. Zhang and S. M. Jacob, "Detecting Spam Email With Machine Learning Optimized With Bio-Inspired Metaheuristic Algorithms," in IEEE Access, vol. 8, pp. 187914-187932, 2020, doi: 10.1109/ACCESS.2020.3030751.

[14] Hosseinalipour, A., Ghanbarzadeh, R. A novel approach for spam detection using horse herd optimization algorithm. Neural Comput & Applic 34, 13091–13105 (2022). https://doi.org/10.1007/s00521-022-07148-x

[15] Ashraf S. Mashaleh, Noor Farizah Binti Ibrahim, Mohammed Azmi Al-Betar, Hossam M.J. Mustafa, Qussai M. Yaseen,Detecting Spam Email with Machine Learning Optimized with Harris Hawks optimizer (HHO) Algorithm,Procedia Computer Science,Volume 201,2022,Pages 659-664,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2022.03.087.