# Classification algorithms for email spam detection: A case for energy conservation

## Introduction

Businesses and private individuals use email on daily basis for communication. While users find email very useful, the rising presence of spam emails is a cause of concern for most users. Spam email classification using machine learning is the application of machine learning to automatically classify spam from non-spam (ham -i.e., email wanted by the recipient).

## Scope

Emails have improved communication in businesses and private lives. In building models to detect spam email and improve the email-user experience, there is a strong need for AI practitioners to also consider the energy requirements of the models being used and their effect on the environment.

## Importance

Huge computational energy use by large models has negative effects on the environment contributing to greenhouse effect (Acosta, et al., 2023; Ahmad, et al, 2021). There exist many contributions on the use of machine learning algorithms in spam email detection (Mansoor et al., 2021; Awad and ELseuofi, 2011; Jiang, 2010; Li, W. and Meng, 2015). However, there exist few literatures that have approached email classification with consideration for energy consumption of models and impact on the environment. Consideration for the environment informs the need for lightweight algorithm.

## Background review

Awad and ELseuofi (2011) took a comprehensive approach and examined six classifiers namely – Naive Bayes (NB), KNN, ANN, Support Vector machines (SVM), artificial immune system and rough set classifier. With accuracy as the number of emails that are correctly identified, NB achieved an accuracy of 99.46%. Another study compared SVM with NB (Ma et al., 2020). The result of this study showed that SVM consistently performed better than NB with varying number of training emails recording 95.5% accuracy as against 94.5% of NB. Some other studies have explored the hybrid approach. In their work (Rahman and Ullah, 2020) used three networks: word embedding, convolutional neural network (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) and compared this hybrid model with Random Forest (RF), SVM, NB and LSTM. This network achieved between 98-99% accuracy in comparison to other models which achieved lower. The ability of CNN to excel in feature extraction has made it an attraction in hybrid model construction for email classifiers especially with the rise of spam images. A multi-modal architecture has been employed by merging image and text classifiers (Seth and Biswas, 2017). In this work, the multi-modal architecture provided better accuracy of 98.11% over separate text classifier of 97.54% and image classifier of 85.89%. Bouke et al. (2023) in their work considered a lightweight approach - RF suitable for spam email classification against traditional ML classifiers like SVM, KNN, Log and DT. In this work, RF with word frequency patterns achieved an accuracy of 97% as against other classifiers with lesser accuracy. In addition to the preceding lines, Schwartz et al. (2020) raised the point of considering energy consumption of models as an important factor in model as against the dominant trend of using accuracy as the principal benchmark in model training. This view finds concurrence in the work of Strubell et al. (2020) which also noted the prime place researchers accord accuracy over computational consumption with attendant harm to the environment. They recommend that researchers should focus more computational efficient algorithms and should not only report accuracy but also computational cost of training their models.

## Objective

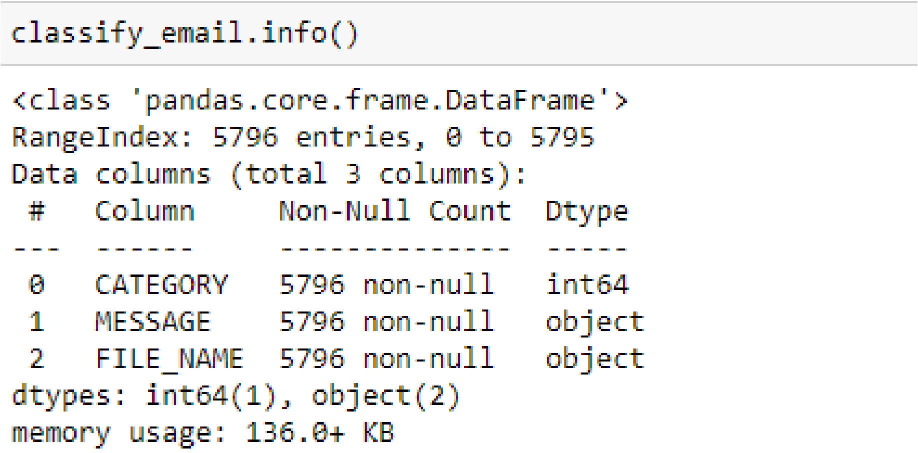
Objective of this study is to further explore the lightweight approach of Bouke et al. (2023), the benchmark for this study, to verify result of this work using a different dataset. Traditional machine learning and deep neural network classifiers will be used for this work. Result will be measured against the 97% accuracy achieved by the benchmark for this study.

## Dataset

The dataset used in this project was sourced from Kaggle.com. It comes under the title Spam Classification for Basic NLP and can be accessed here. It is a 6MB excel file made up of raw mail messages and combinations of plain messages with headers and some with HTML tag. The label column has class imbalance (67/33). Feature column containing emails (texts) and label column containing class/label make this dataset a good candidate for NLP tasks. It is suitable for data preprocessing, vectorization and classification tasks.

## Exploratory Data Analysis

From data exploration, it can be discovered that this data has a total of 3 columns and 5796. There in null row, and MESSAGE (feature) and CATEGORY (label) columns are columns of interest in this classification project.



```
classify_email.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5796 entries, 0 to 5795
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   CATEGORY   5796 non-null   int64 
 1   MESSAGE    5796 non-null   object
 2   FILE_NAME  5796 non-null   object
dtypes: int64(1), object(2)
memory usage: 136.0+ KB
```

**Figure 1: Dataframe summary**

Word cloud (see figure 2) helps us view distribution of words in the feature column

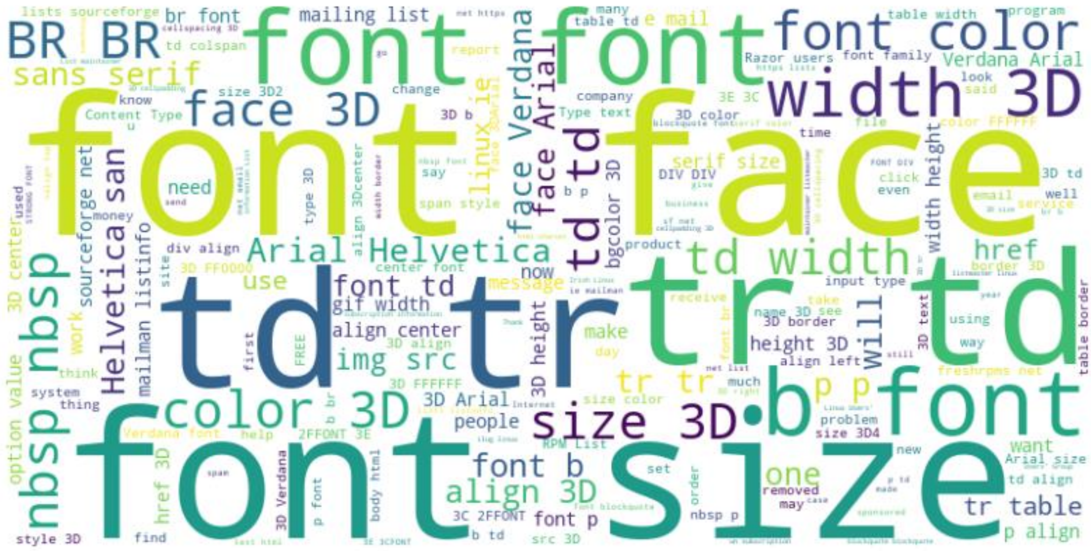


**Figure 2: Wordcloud**

See text length distribution shown in Figure 3 below:

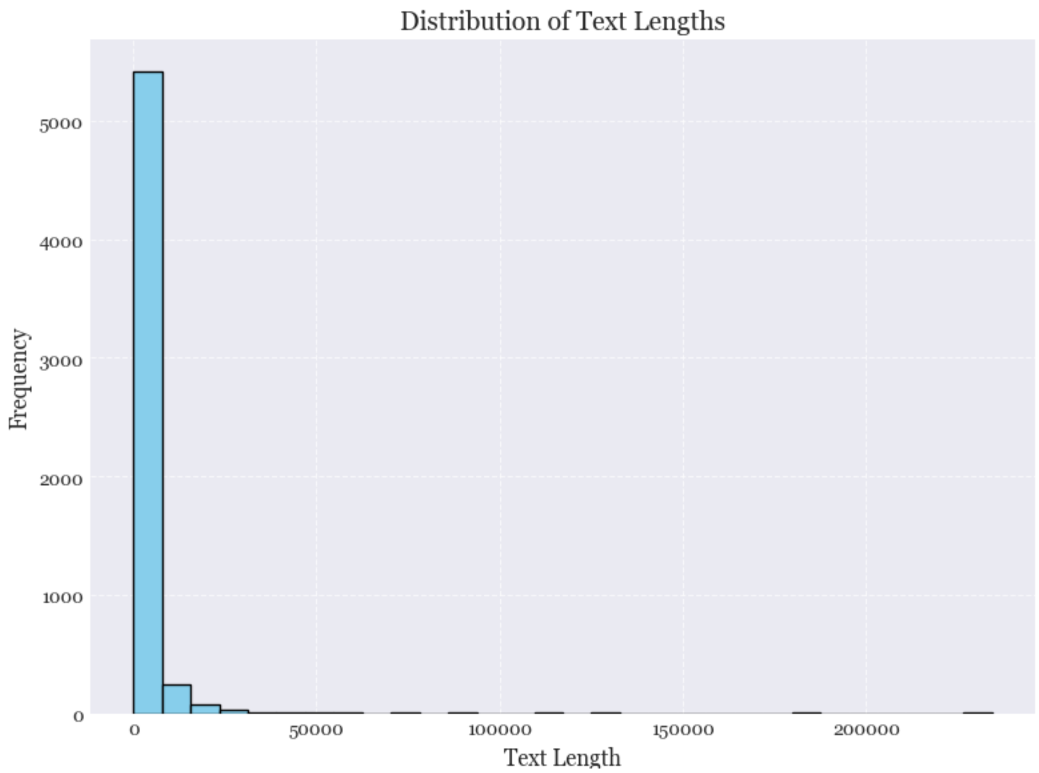
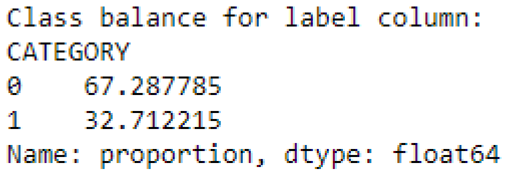


**Figure 3: Word length distribution**

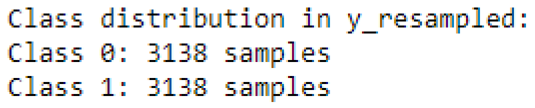
The label column has integer datatype consisting of 0 and 1. 1 is for spam and 0 is for non-spam. Checking for class balance, we have more population in the non-spam class as show below:



```
Class balance for label column:
CATEGORY
0    67.287785
1    32.712215
Name: proportion, dtype: float64
```

**Figure 4: Unbalanced class distribution**

I used SMOTE for class balancing. SMOTE brings about class balance by introducing synthetic instances for the minority class. One merit of using SMOTE is that it checks overfitting. After class balancing, we now have class balance as seen in Figure 5:



```
Class distribution in y_resampled:
Class 0: 3138 samples
Class 1: 3138 samples
```

**Figure 5: Balanced class distribution**

## Data preprocessing

I performed data cleaning in this order

1. Removing special characters and punctuations as they do not have any semantic utility and can introduce noise

2. converted all letters to lowercase to help normailse data and bring consistency in meaning of words
3. Removed stopwords to help bring about focus on meaningful content of texts
4. Lemmitization was done to ensure consistency in inflected forms of words, improve tokenization accuracy and reducing vocabulary size

### Vectorization

Given than machines cannot read words but numbers, vectorization was used to convert text into numbers readable by comuter. I used TF-IDF for vectorization in classification using traditional machine learning model and used SentenceTransformers, a transfer learning model, to perform sentence embedding before classification using deep neural networks. Vectorisation will take care of tokenization.

While TF-IDF vectors are sparse and suitable for traditional machine learning, SentenceTransformers provides transfer learning and offers semantic understanding suitable for deep learning algorithm

### Dimensionality reduction

For this project, I used Truncated Singular Value Decomposition. Truncated SVD is a dimensionality reduction technique suited for sparse data like text data. It is like PCA; and operates on sparse matrices directly. The cumulative explained variance ratio, as seen in figure 7, has changed from 0.016-0.024 to 0-1 after reduction suggesting that the reduced dimensions now explain a larger proportion of the total variance present in the original dataset. Dimensionality reduction will improve model efficiency by reducing high-dimensional text data into fewer, more manageable features, while retaining essential information for accurate classification.

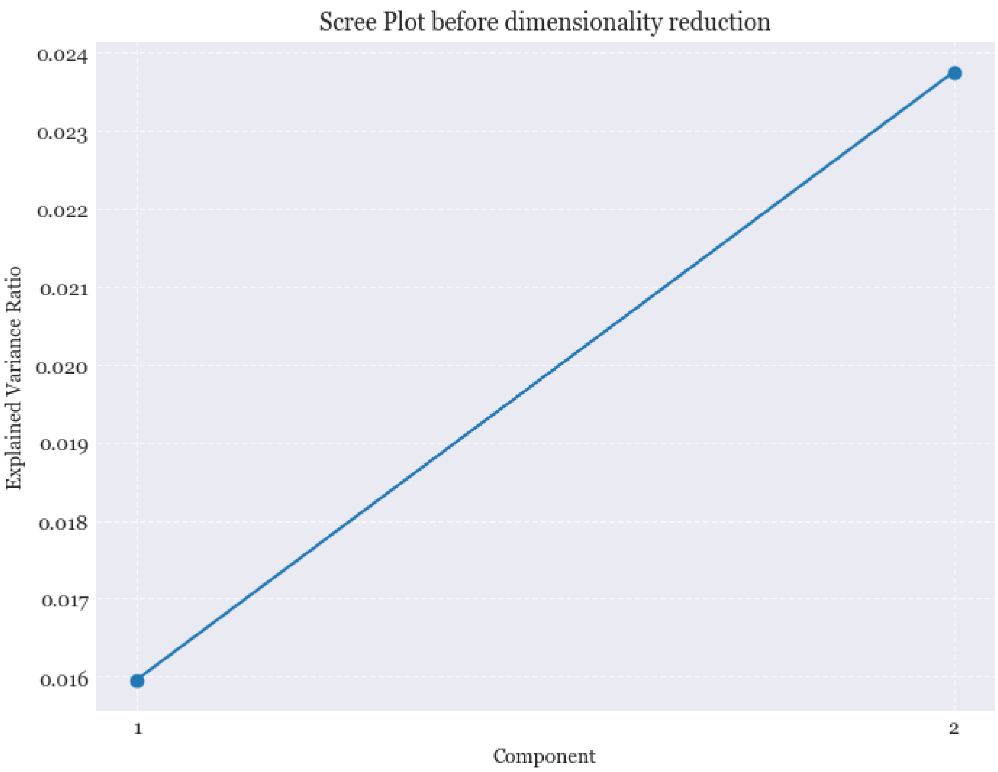


**Figure 6: Before dimensionality reduction**

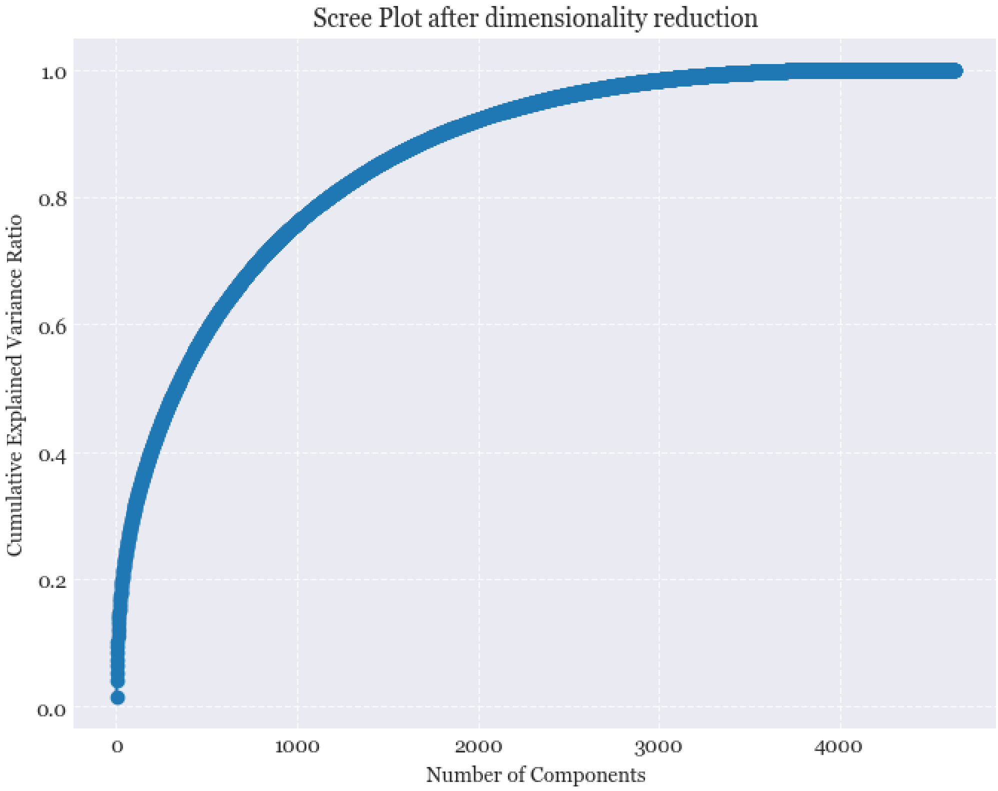


**Figure 7: After dimensionality reduction**

## Traditional Machine learning

### 1. Support Vector Machines:

The SVM algorithm operates by singling out a decision boundary called the hyperplane which best segregate different classes in a dataset. SVM selects the hyperplane which has the widest distance resulting in a hyperplane which generalises better and accurately classifies new data (Yu and Kim, 2012). Once this hyperplane is established, new instances are then classified into matching group (Amarappa and Sathyanarayana, 2010).

SVMs operate better with smaller datasets and high-dimensional data owing to their reliance on support vectors and not the entire dataset. SVM has shown to perform well over algorithm like Naive bayes (Yamamori and Thida, 2020) in spam email classification.

### 2. Random Forest (RF)

RF operates on the concept of ensemble technique. RF works by training many decision trees on a dataset and then arrives at a democratic result by computing average of the prediction of these trees as its prediction. RF is used for both classification and regression and takes care of overfitting.

RF handles high dimensional data very well and is suited for spam email classification with large feature spaces. Earlier works have used random forest in email classification with RF showing superior accuracy over SVM (Taylor and Ezekiel, 2020), and proving to be lightweight algorithm in email classification (Bouke, et al, 2023)

### 3. Naive bayes (NB)

NB is based on the Bayes theorem. Bayes theorem is used to estimate the probability that a given sample belongs to any class. NB is a probabilistic classifier and operates with an assumption of independence between features. It is simple and robust to noise, and these make it suitable for text classification that has high-dimensional feature spaces.

Feng et al. (2016) have used a support vector machine based naive Bayes algorithm for spam filtering.

### 4.Logistic regression (LR)

Logistic Regression is a statistical method used to categorize data into one of two groups. While it is used for binary classification, it can be adapted for multi-class classification. It operates by employing the logistic function to transform a linear combination of input features into a probability value ranging between 0 and 1. It calculates a decision boundary that separates the instances of different classes.

Its simplicity and interpretability make it a good choice for classification tasks like email classification. Wijaya and Bisri (2016) have also explored using LR and Decision Trees in a hybrid implementation for email classification.

**5.K-Nearest Neighbors (KNN)**
KNN uses proximity to make classification. It operates by classifying a new data point based on how its nearest neighbours are classified. It calculates the distance by measuring the Euclidean distance to find the group of nearest neighbours to a data point. It makes no assumptions, and this enables it to detect relationships between features and the target.

KNN's non-assumption of relationship between features and target variable makes it useful in classification where non-linearity is an important consideration. Murugavel and Santhi (2020) have used KNN in classification of emails for spam detection.

***Choice traditional machine learning model for experiment:***

RF and SVM were chosen here mostly for the abundance of literature in comparing the performance of these two algorithms (Taylor and Ezekiel, 2020; Shafi'i, 2018; Jukic , et al., 2015; Tang, et al., 2008) with differing results on superior classification accuracy. While the benchmark for this work did not show the computational consumption of models used as to justify lightweight description of the combination of RF algorithm and word frequency patterns, the work of Murugavel and Santhi (2020) showed RF's resource consumption of 0.57 compares with least KNN 0.54 (which the work proved to be the algorithm with least computational resource consumption).

## Deep Learning Methods

1. **MLP**

Muli-layer perceptron is a modern feedforward artificial neural network. It is made up of an input layer, one or more hidden layers and an output layer. The hidden layers have neurons that learn relationships between input features and class labels in the input data, and the output layer presents the result of the classification.

Their ability to capture non-linear relationship makes them suitable for text classification. Tamilarasan et al. (2022) have used MLP in spam email classification.

2. **LSTM**

Traditionally, Recurrent Neural Networks (RNNs) find long sequences challenging and this leads to vanishing or exploding gradients. Long Short-Term Memory

(LSTM) is a type of RNN that was designed to handle sequential data, and to address the vanishing gradient challenge of RNN.

The principal component of the LSTM that helps it address the vanishing gradient issue is the memory cell. The memory cell allows the network to keep information over long sequences. LSTM employs 3 gates (forget, input and output) to manage the flow of information in and out of the cells.

LSTM's ability to handle sequential data capturing long-term dependencies and contextual information in the process makes it suitable for text classification. Bi-directional LSTM with CNN in email classification has shown to outperform NB, CNN, SVM (Rahman, S.E. and Ullah, 2020).

### 3. GRU

GRU is a simplified version of LSTM. Its architecture is like the LSTM. However, GRU has a simpler architecture (It has 2 gates –update and reset gates). GRU has fewer parameters, making it computationally more efficient and faster to train compared to LSTM.

GRU is a choice algorithm in an environment (like the target of this work) with limited computational resources that needs to work with sequential data and understand long term dependencies. Wanda (2023) has used GRU in email spam detection project.

### 4. CNN

Convolutional Neural Network (CNN) also known as ConvNet is a deep learning algorithm which specializes in learning and extracting features from gridlike input images. CNN is inspired by the visual cortex of animals. A CNN is composed of an input layer, an output layer, and many hidden layers. CNN operates in 2 major steps- feature learning and classification (prediction).

In recent time, CNNs have been used in NLP project (Huang, 2019; Yaseen, 2021). Sainath, et al. (2020) have shown that higher level features can be extracted by the convolution layer. The advent of image spam has made CNN a choice mode over traditional algorithms like SVM, Naive Bayes, Decision Tress (Seth and Biswas, 2017).

### 5. Transformers

Vaswani et al. (2017) in their work "Attention Is All You Need" introduced Transformers. It is a deep learning architecture that is based on attention. Self-attention makes the model to focus on different parts of the input sequence and extract relevant information. Given that words in each corpus of texts are not equally important, attention takes this into consideration. Self-attention calculates attention scores for individual texts within the input sequence by considering the interactions between every pair of words.

Transformers capture long term dependencies and understand context better than RNNs and this makes it an algorithm of choice in contemporary NLP tasks. Variants like Generative Pre-trained Transformers (GPT) and Bidirectional Encoder Representations from Transformers (BERT) exist. Sahmoud and Mikki (2022) have used BERT in spam email classification.

***Choice deep learning model for experiment***:
GRU and LSTM were chosen because of their ability to handle sequential data capturing long-term dependencies and contextual information. These consume lesser energy than BERT variant of Transformers.

## Implementation and Refinement:

**Libraries used**:
I used scikit-learn library for machine learning tasks in Python as well tensorflow which is used for building and deploying machine learning models. I also used pandas for handling data frames and series; NumPy for numerical operations; seaborn for statistical data visualization as well as matplotlib for plotting. Wordcloud used to create visualizations of words from text data; imblearn used for dealing with imbalanced datasets; sentence-transformers -employed for encoding and transforming sentences into dense vector representations; Spacy used for NLP tasks like tokenization.

For all models, dataset was split into train and test (20%). Models were trained and evaluated with train and test but this work is reporting test accuracy and F1. Summary of evaluation result is shown in table 1 as well as the confusion matrix for all four models is shown in figure 7 below.

**SVM**
I defined the SVM model SVC (kernel='linear', C=1.0, gamma='scale') and then trained. Following evaluation, test accuracy of 99% (.99) was archived and F1 score of 99% (.99) was recorded. I tuned the C-parameter setting to .01. Tuning C is essential in handling outliers and test accuracy dropped to 76% while F1 came to

71%. Bring C back to 1, I restored earlier value of C and tuned gamma to .0001 and each accuracy and F1 returned to the initial value of 99%. Gamma controls sensitivity to noise as well as generalization performance.

**RF**

Model was defined rf_model = RandomForestClassifier(n_estimators=100, max_depth=10, class_weight='balanced', random_state=42) and I then moved on to train the model. On evaluation, test accuracy of 95% (.95) was archived and F1 score of 95% (.95) was recorded. Upon tuning the number of trees from 100 to 500, accuracy and F1 each improved to 96%. N_estimator determines the number of trees in the ensemble model. Tree depth parameter controls overfitting. Increasing tree depth from 10 to 100 saw train accuracy/F1 reach 100% with test accuracy and F1 reaching 98% each.

**GRU**

Using GRU for this experiment, a test accuracy of 97% (.97) was achieved and F1 score of 95% (.95) was achieved. I tuned batch size from 128 to 256, a test accuracy of 98% (.98) was achieved and F1 score of 96% (.96) was achieved. With an increase in number of epochs from 10 to 100, test accuracy stayed at 98% while F1 moved to 97% result. I noted that the model overfitted with training accuracy and F1 score reaching 100% despite introduction of early stopping to prevent overfitting. Larger number of epochs are known to increase model performance, but overfitting should be monitored. Tuning batch size is a form of regularization.

**LSTM**

LSTM model achieved a test accuracy of 96% (.96) and F1 score of 95% (.95). I tuned hyperparameters to check for model improvement. I increased number of epochs from 10 to 150 and introduced early stopping to guide against overfitting, and the model produced test accuracy of 97% and F1 score of 96%. I added another layer but the accuracy and F1 remained same at 97% and 96% respectively. Adding another layer was to increase the model's capacity to learn.

| S/N | Algorithm | F1 | Accuracy |
|---|---|---|---|
| 1 | SVM | 0.99 | 0.99 |
| 2 | RF | 0.98 | 0.98 |
| 3 | GRU | 0.97 | 0.98 |
| 4 | LSTM | 0.95 | 0.97 |

**Table 1: Summary of Evaluation Result**

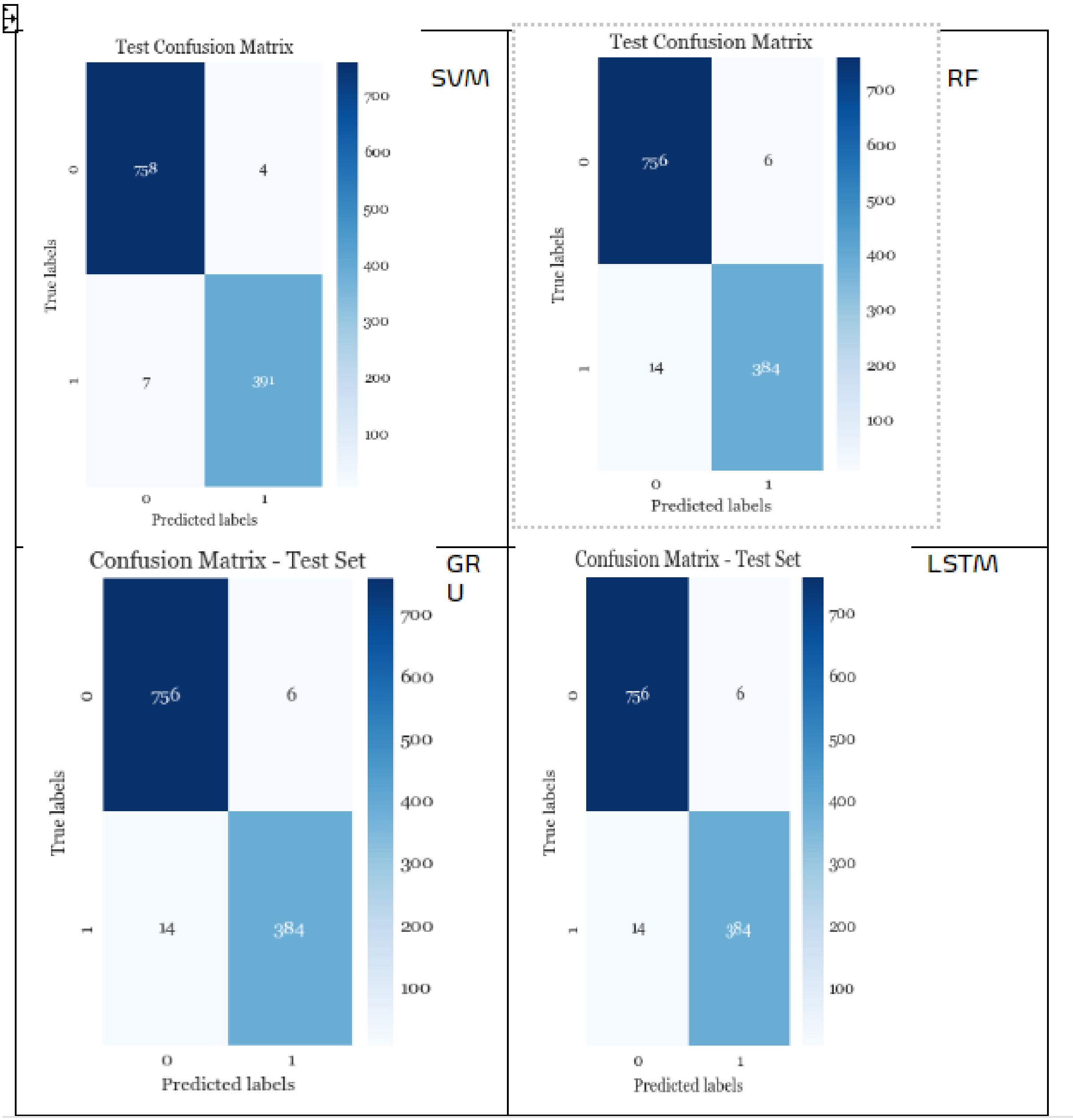


**Figure 8: Confusion matrix for the 4 models**

## Conclusion

This study has demonstrated that both traditional machine learning and deep learning methods can be employed for email NLP text classification and have gone ahead to show that with spam email classification. Improving accuracy of models to detect spam from emails desired by users helps life and business. In this experiment, SVM with 99% accuracy and F1 score respectively has shown to be the best model with RF following with 98% for each of accuracy and F1 score. While RF with many trees can consume memory, SVM is known to consume more computational resources (Marquez, et al. (ed), 2024). Murugavel and Santhi (2020)'s work has earlier proven RF as low in computational resources consumption. This makes RF a choice model in our world where there is need to take urgent steps to preserve the environment. RF model here with 98% accuracy performed better than the baseline model for this work which had 97% accuracy. Future studies should focus on experimenting on tweaking hyperparameters of RF for improved performance as well as preprocessing steps including improved embedding techniques that can help introduce low computational resource usage and performance improvement.

## References

Acosta, A.G., Jarrín, M.T. & Riordan, S. (2023) The Environmental and Ethical Challenges of Artificial Intelligence. *Observer Research Foundation*.

Awad, W.A. & ELseuofi, S.M. (2011) Machine learning methods for e-mail classification. *International Journal of Computer Applications*, *16*(1), pp.39-45.

Ahmad, T., Zhang, D., Huang, C., Zhang, H., Dai, N., Song, Y. & Chen, H. (2021) Artificial intelligence in sustainable energy industry: status quo, challenges and opportunities. *Journal of Cleaner Production*, *289*, 125834.

Amarappa S. & Sathyanarayana S. V. Data classification using Support vector Machine (SVM), a simplified approach. *International Journal of Electronics and Computer Science Engineering* 435

Bouke, M. A., Abdullah, A., Abdullah, M. T., Zaid, S. A., El Atigh , H., & ALshatebi, S. H. (2023). A lightweight machine learning-based email spam detection model using word frequency pattern. *Journal of Information Technology and Computing*, *4*(1), 15–28.

Jukic, S., Azemovic, J., Keco, D. & Kevric, J. (2015) Comparison of machine learning techniques in spam e-mail classification. *Southeast Europe Journal of Soft Computing*, *4*(1).

Jiang, E.P., 2010. Content-based spam email classification using machine-learning algorithms. *Text Mining: Applications and Theory*, 37-56.

Huang, T. (2019). A CNN model for SMS spam detection. In *2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE),* 851-85110.

Ma, T.M., Yamamori, K. & Thida, A. (2020) A comparative approach to Naïve Bayes classifier and support vector machine for email spam classification. In *IEEE 9th Global Conference on Consumer Electronics (GCCE),* 324-326.

Mansoor, R.A.Z.A., Jayasinghe, N.D. & Muslam, M.M.A. (2021) A comprehensive review on email spam classification using machine learning algorithms. In *International Conference on Information Networking (ICOIN)* 327-332.

Marquez, F. P. G., Eken, S., Hameed, A. A., Jamil, A., & Ramírez, I. S. (2024) *Computing, Internet of Things and Data Analytics. Selected Papers from the International Conference on Computing, IoT and Data Analytics (ICCIDA).* Switzerland: Springer Nature.

Murugavel, U. and Santhi, R., 2020. K-Nearest neighbor classification of E-Mail messages for spam detection. *ICTAT Journal on Soft Computing*, *11*(1), 2218-2221.

Rahman, S.E. & Ullah, S. (2020) Email spam detection using bidirectional long short term memory with convolutional neural network. In *IEEE Region 10 Symposium (TENSYMP),* 1307-1311.

Sainath, T.N., Senior, A.W., Vinyals, O. & Sak, H. (2020) Convolutional, long short-term memory, fully connected deep neural networks. Google LLC, U.S. Patent 10,783,900.

Seth, S. & Biswas, S. (2017) Multimodal spam classification using deep learning techniques. *13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 346-349.

Sahmoud, T. & Mikki, D.M. (2022) Spam detection using BERT. Available online: https://doi.org/10.48550/arXiv.2206.02443 [Accessed 25/04/2024].

Shafi'i, M.A., Maryam, S., Oluwafemi, O., Ismaila, I. & John, K.A. (2018) Comparative analysis of classification algorithms for email spam detection. Available online: https://doi.org/10.5815/ijcnis.2018.01.07 [Accessed 23/04/2024].

Schwartz, R., Dodge, J., Smith, N.A. and Etzioni, O. (2020) Green ai. *Communications of the ACM*, *63*(12), 54-63.

Singh, M. & Pamula, R. (2018) Email spam classification by support vector machine. In *International Conference on Computing, Power and Communication Technologies (GUCON),* 878-882.

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645-3650.

Tamilarasan, S.M., Hithasri, M. & Pille, K., 2022. Email spam detection using multilayer perceptron algorithm in deep learning model. In *Information and Communication Technology for Competitive Strategies (ICTCS 2021) ICT: Applications and Social Interfaces,* 581-587. Singapore: Springer Nature Singapore.

Tang, Y., Krasser, S., He, Y., Yang, W. & Alperovitch, D. (2008) Support vector machines and random forests modeling for spam senders behaviour analysis. In *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference,*1-5.

Taylor, O.E. & Ezekiel, P.S. (2020) A model to detect spam email using support vector classifier and random forest classifier. *Int. J. Comput. Sci. Math. Theory*, *6*(1), 1-11.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, Kaiser, L. & Polosukhin, I (2017) Attention Is All You Need. Available online: https://doi.org/10.48550/arXiv.1706.03762 [Accessed 25/04/2024].

Wanda, P. (2023) GRUSpam: robust e-mail spam detection using gated recurrent unit (GRU) algorithm. *International Journal of Information Technology*, *15*(8), 4315-4322.

Wijaya, A. and Bisri, A. (2016). Hybrid decision tree and logistic regression classifier for email spam detection. In *8th international conference on information technology and electrical engineering (ICITEE),* 1-4.

Yaseen, Q. (2021) Spam email detection using deep learning techniques. *Procedia Computer Science*, *184*, 853-858.

Yu, H., Kim, S. (2012). SVM Tutorial — Classification, Regression and Ranking. In: Rozenberg, G., Bäck, T., Kok, J.N. (eds) Handbook of Natural Computing. Springer, Berlin, Heidelberg.