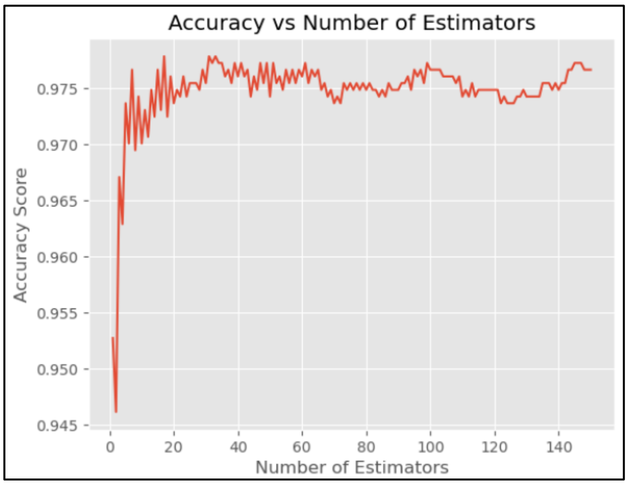
Natural Language Analysis

In the realm of natural language analysis, preparing the data properly is important for the success of any model. For the task of SPAM SMS detection, the initial dataset comprised around 85% HAM messages and 15% SPAM messages, reflecting a significant class imbalance. The preprocessing section for these text messages involved converting all text to lowercase, removing punctuation, and eliminating common stopwords. These steps help reduce the noise from the dataset, focusing on the more meaningful content of the messages.

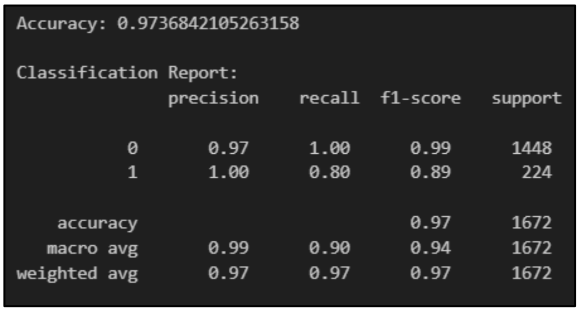
One of the primary feature extraction techniques we applied to the dataset is Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF is an algorithm that counts words' weight by considering the word's frequency and in how many files the word can be found (Hakim et al., 2014). This dual approach helps to amplify the influence of terms that are unique to the documents, with the advantages of distinguishing words that are more likely to be important and are not just random words that commonly appear in texts and handling large datasets with diverse vocabulary very well. However, the algorithm also comes with disadvantages of treating each word independently, ignoring the content and context of the words, and having results that are usually high-dimensional and sparse vectors, which in some situations could be computationally inefficient

On the other hand, Word2vec, is a neural network that includes two learning models: Continuous Bag of Words (CBOW) and Skip-gram. CBOW predicts the word given its context while Skip-gram do it the opposite (Ma, L., & Zhang, Y., 2015). By training the model on the corpus of SMS messages, each word is viewed as a vector in a defined space where the vector shows similarities to other words. The advantages of Word2vec come with the richness in the semantic area that captures the relationships between words, the efficiency of not having so much dimensional structure compared with TF-IDF, and the ability to fit in and deal with all kinds of natural language problems; meanwhile, it also has the disadvantages of not being able to provide dynamic impact on the work. Once trained, the model cannot interpret any new word without adjustment.

To utilize the features extracted from TF-IDF and Word2vec, the Random Forest classifier was chosen to categorize messages. One key thing to mention is using Grid search to find the best n_estimator number for the best accuracy and efficiency. The result of the model with TF-IDF is as follows:

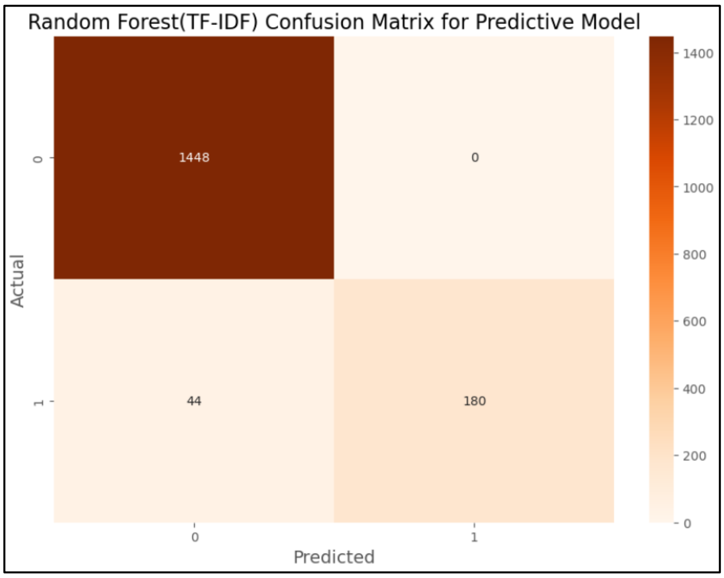


(Diagram 1: Result of Grid Search of RF with TF-IDF)



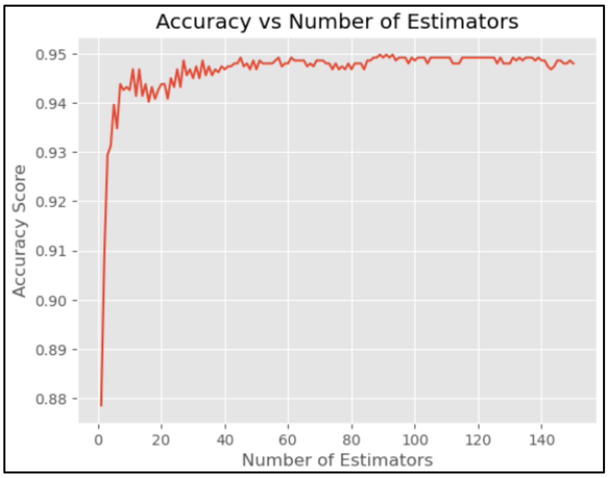
```
Accuracy: 0.9736842105263158

Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99      1448
           1       1.00      0.80      0.89       224

    accuracy                           0.97      1672
   macro avg       0.99      0.90      0.94      1672
weighted avg       0.97      0.97      0.97      1672
```

(Table 1: Result of Random Forest with TF-IDF)

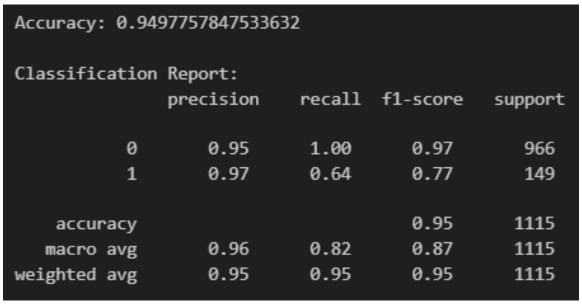


(Diagram 2: Confusion Matrix of Random Forest with TF-IDF)

The model achieved a notable accuracy of 97.37% with n_estimators as 37 in 4.0 seconds, demonstrating strong predictive capabilities. Alternatively, the result of the model with Word2vec is shown as follows:

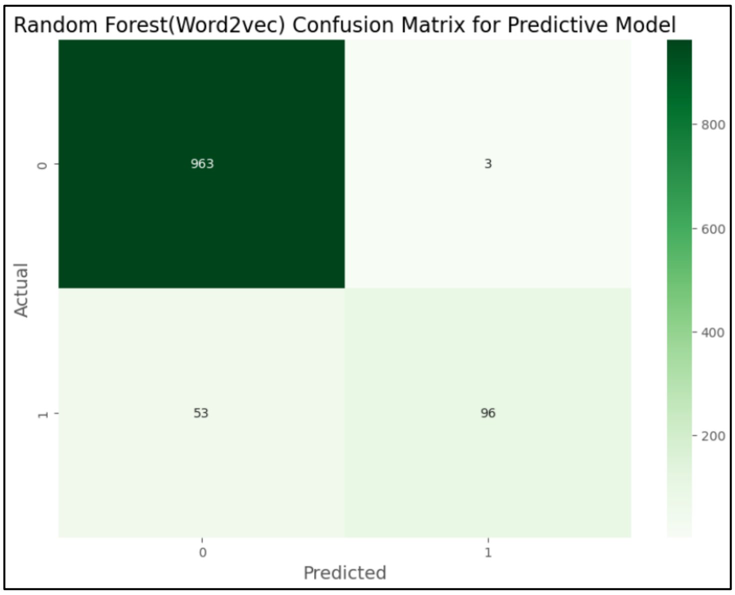


(Diagram 3: Result of Grid Search of RF with Word2vec)



```
Accuracy: 0.9497757847533632

Classification Report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97       966
           1       0.97      0.64      0.77       149

    accuracy                           0.95      1115
   macro avg       0.96      0.82      0.87      1115
weighted avg       0.95      0.95      0.95      1115
```

(Table 2: Result of Random Forest with Word2vec)



(Diagram 4: Confusion Matrix of Random Forest with Word2vec)

With an accuracy of 95.16% in 2.9 seconds of training time with n_estimators as 47, the result is just slightly lower than TF-IDF, showing good performances the classifier can have with these two kinds of feature extraction, especially when both the recall rate and f1 score are also coming with a high rate.

Moving on to the deep learning model, Convolutional Neural Networks (CNN) were chosen. For the architecture, both the model with TF-IDF and the model with Word2vec consist of a convolutional layer with 128 filters and a kernel size of 5, using ReLU activation, and an initial input shape parameter to fit the dimensions, designing to capture the most informative features. A global max pooling layer is added to reduce the dimension of the data, ensuring only significant features are passed through. Two dense layers, with the final layer using a sigmoid activation function to output the probability of being SPAM or HAM. A batch normalization layer and a dropout layer are also added to the model with Word2vec to fight imbalanced problems within the data. Both models were compiled with the Adam optimizer and used binary cross-entropy as loss function, ideal for binary classification problems.

One key thing to mention is to fight the imbalanced problem within the data, both datasets were resampled using SMOTE for oversampling the minority class, SPAM, and Tomek Links for cleaning some samples that may be misunderstood, aiming to provide a more balanced and improved model.

With all models conducted, the results can be listed as followed:

Model	**Accuracy**	**Training Time**
RF (TF-IDF)	97.37%	4.0s
RF (Word2vec)	95.16%	2.9s
CNN (TF-IDF)	70.93%	56.1s
CNN (Word2vec)	54.35%	13.4s

References:

Hakim, A. A., Erwin, A., Eng, K. I., Galinium, M., & Muliady, W. (2014, October). Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach. In *2014 6th international conference on information technology and electrical engineering (ICITEE)* (pp. 1-4). IEEE.

Ma, L., & Zhang, Y. (2015, October). Using Word2Vec to process big text data. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 2895-2897). IEEE.