OPEN ACCESS
Indo-JC
Indonesian Journal on Computing
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 6, Issue. 1, April 2021. pp. 85-94
doi:10.34818/indojc.2021.6.1.452

Implementation of Naïve Bayes and Gini Index for Spam Email Classification

Fikri Rozan Imadudin [1], Danang Triantoro Murdiansyah [2], Adiwijaya [3]

School of Computing, Telkom University
Jl. Telekomunikasi No.1, Bandung 40257, Indonesia

[1] fikrirozan@students.telkomuniversity.ac.id
[2] danangtri@telkomuniversity.ac.id
[3] adiwijaya@telkomuniversity.ac.id

Abstract

Email is one of communication tools still used by people today. Currently, email still has a challenge to solve, that is spam email. Spam email is email that can annoy and endanger the recipient of the spam email. In this study, Multinomial Naïve Bayes and Complete Gini Index Text are used for spam email filtering. Multinomial Naïve Bayes serves as a classifier whether email is spam or not, while Complete Gini Index Text functions as a selector of features. The result of this study is that the combination of Multinomial Naïve Bayes and Complete Gini Index Text using 80,000 features can produce a 0.39% better accuracy than Multinomial Naïve Bayes by using 184,161 features without Complete Gini Index Text.

Keywords: Complete Gini-Index Text, Multinomial Naïve Bayes, Email Classification

Abstrak

Email adalah salah satu alat komunikasi masih sering digunakan oleh orang-orang pada saat ini. Saat ini email masih memiliki tantangan untuk diselesaikan, yaitu email spam. Email spam merupakan email yang dapat menganggu dan membahayakan penerima email spam tersebut. Pada penelitian ini digunakan Multinomial Naïve Bayes dan Complete Gini Index Text untuk filterisasi email spam. Multinomial Naïve Bayes berfungsi sebagai pengklasifikasi apakah email termasuk email spam atau bukan, sedangkan Complete Gini Index Text berfungsi sebagai seleksi fitur yang ada pada email. Hasil dari penelitian ini adalah penggabungan Multinomial Naïve Bayes dan Complete Gini Index Text dengan menggunakan 80.000 fitur dapat menghasilkan akurasi yang lebih baik 0.39% daripada Multinomial Naïve Bayes dengan menggunakan 184.161 fitur tanpa Complete Gini Index Text.

Kata Kunci: Complete Gini-Index Text, Multinomial Naïve Bayes, Klasifikasi Email

I. Introduction

EMail, short for electronic mail, is a message that may contain text, files, images, or other attachments sent through a network to a specified individual or group of individuals. In the 2018s, the number of emails received and send every day reached 281 billion with 3.8 million users. Instead of benefits, email has a challenge to solve, that is spam email. Spam emails are propagated by the spammers for simple marketing purposes to unfold more malicious activities such as financial disruption and reputational damage, both in personal and institutional front [1]. Email spamming refers to the act of distributing unsolicited messages, optionally sent in bulk, using email; whereas emails of the opposite nature are known as ham, or useful emails [2]. In the 2017s, 54,6% of incoming emails considered spam, which has fraud, fake content, pornography, etc.

Received on January, 2021. Accepted on Mei, 2021

In email filtering, before email enter the folder, the email is checked so that spam-type emails enter the spam folder by using the classification machine / system. Before email is classified, some non-essential sentences reduced using preprocessing techniques which include stop-words removal, alphanumeric-words removal, stemming and lemmatization. Next step, the email is processed using machine learning algorithm.

Email is one of the text classifications which has many numbers of corpus. H. Park et al. develop the Gini-index text feature selection [3]. Their research resulted in a new feature selection called Complete Gini-Index Text (GIT), used to optimization text categorization for large number of features that are previously using the Gini-index text feature selection. The feature selection reduces the amount of feature. The feature selection also increases speed during classification and can reduce features less relevant to the category to be classified. There have also been many studies conducted to classify emails and texts using the Naïve Bayes, Super Vector Machine, AdaBoost, Logistic Regression. As for the classification using a variety of feature selection such as Mutual Information, Information Gain, Chi-Square, etc. [3][4][5][6][7].

In this study, an email classification system is built using multinomial naive bayes model. In addition to the model, the Complete Gini Index Text feature selection is also used as well. The email dataset used is in English language, provided by Enron [8]. The email classification system will classify whether the email is spam email or not spam email (ham). The purpose of this study is to improve the accuracy and performance of the classification algorithm/model by using the Complete Gini-Index Text method. The results also compared to other feature selection, that is Chi-Square [9].

II. LITERATURE REVIEW

There have been many studies to solve the problem of spam. As in previous studies, Gad et al. [4] using Lingspam data as training data to be used in classifying into spam and ham. They use Mutual Information as a feature selection to increase speed and accuracy in spam classification. Before that, the data converted into a vector space model (VSM) and then entered to the feature selection. Furthermore, it classified using the Logistic Regression method, J4.8, Bagging, Naïve Bayes and AdaBoost. From the results of their research that Mutual Information gets the best results when using the logistic regression method.

Isaac et al. [10] applied Bayesian using Enron-spam and Lingspam data to calculate the values of many sets of keywords and context-matching keywords so that the scores of each email obtained. They compare it with a set of keywords that applied with Bayesian with an average accuracy of 92% when used in data Enron-spam.

M. Singh [11] introduced classification spam e-mails using the Intelligent Water Drops algorithm that implemented with Naïve Bayes. The algorithm IWD they use is to select a subset of features and eliminating redundant features. They implemented the algorithm in the UCI dataset which had 1813 spam data and 2788 ham. The results they get compared with genetic algorithms and ant colony which method used as.

Yang et al. [6] implemented a feature selection algorithm based on Binomial hypothesis testing that applied to spam filtering. They use pu1, pu2, pu3, Lingspam, and CSDMC2010 datasets. The data applied by using the Information gain feature, Improved Gini Index, CHI2, Poisson Distribution, and Binomial hypothesis testing.

Sebastian S.R Gomes et al. [7] compared the Naïve Bayes method and the Hidden Markov Model in his research. They compare the accuracy and performance of several pre-processing techniques such as stemming, stop-words removal and lemmatization. From the results of the comparison, it is found that each combined preprocessing process can produce good performance depending on the machine learning algorithm used. The highest accuracy of their research results obtained by 91.28%, for the Hidden Markov Model algorithm using the Stemming process. The results for Naïve Bayes were 79.19%, with a stop-words removal process using lemmatization.

Thae M. M et al. [12] proposes Support Vector Machine and Naïve Bayes Classifier, to classify the emails into spam or ham based on the content or the body of the emails. Support Vector Machine can be used to represent an email in vector space in which each feature means one dimension. In Naïve Bayes Classifier, independent words are considered as features. The two methods are compared in terms of precision, recall, and F-measure, with the aim of finding the best method.

A. Complete Gini Index Text

The Gini-index text has a problem in the value of Gini, which is difficult in determining good subset features. This difficulty is due to documents classified as having unequal features and classes. So it developed into the Complete Gini-Index Text in the 2010s by H. Park et al. [3] by entering some formulations into the Gini-Index Text.

The GIT (Gini-Index Text) feature selection has three equations namely GIT_A (W, C_k), GIT_B (W, C_k) and GIT_C (W, C_k). W defined as the feature set W = {w_1, w_2, w_3, ..., w_j} and C_k is the spam class or not spam class. The GIT equation defined as follows.

$$GIT_A(W, C_k) = P(C_k|W)^2) \quad (1)$$

$$GIT_B(W, C_k) = \left|\frac{P(C_k|W)^2}{log_2 P(W)}\right| \quad (2)$$

$$GIT_c(W, C_k) = \frac{P(C_k|W)^2}{|log_2 P(C_k|W)^2|} \quad (3)$$

Where $P(C_k|W)$ is a conditional probability C_k for all features, and P(W) is probability of all features W.

B. Naïve Bayes

In-text classification and sentimental analysis, the Naïve Bayes algorithm is the most popular algorithm used. There are several types of Naïve Bayes including Naïve Bayes Gaussian, Naïve Bayes Multivariate Bernoulli, multinomial Naïve Bayes. Of the various types of naïve Bayes processed using different forms of data obtained at the pre-processing stage [13]. The naïve bayes classification assumes that a feature does not depend on other features, which called the naïve assumption. Naïve Bayes is generally defined as follows:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (4)$$

Where, X is the set of features, X = {x1, ..., xj} and C represents the set of classes or category C = {c_1, ..., c_k}.

C. Multinomial Model

Multinomial naïve bayes has been successfully used in various classification texts, categorical texts, sentimental analysis, and email classifications. In the multinomial naïve bayes classification email achieved the best accuracy compared to other naïve bayes models. Multinomial naïve bayes assumes that a document X does not depend on the category or class C [13][14].

$$argmax_{C\epsilon C1 \ldots Ck} P(C|X) = argmax_{C\epsilon C1 \ldots Ck} P(C) \prod_{i=1}^{m} P(W_i|C) \quad (5)$$

Probability of W_i in class C define as:

$$P(W_i|C) = \frac{1+N(W_i|C)}{|V|+N(C)} \quad (6)$$

Where $N(W_i|C)$ is the number of occurrences of W_i features in documents with class C, |V| is the number of vocabulary terms for all documents and N (C) is the sum of all features of W in the category of spam and not spam.

D. Performance Measurement

After the classification, we will measure how well the classification machine works well. To measure the result will use a confusion matrix as follows.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1\ Score = \frac{2\ Presisi\ Recall}{Presisi+Recall} \quad (10)$$

Accuracy is the result of the accuracy of the classification, precision is a measure of the success of a document regarded as spam, recall is the success rate of a spam-indicated document by the classification engine, and F1 Score is to measure the average recall and precision of classification results. Where TP (True Positives) is number of documents from spam and classified as spam, TN (True Negatives) is number of documents not spam and classified not spam, FP (False Positives) is number of spam documents and classified as not spam, and FN (False Negatives) is number of documents not spam and classified as spam.

III. RESEARCH METHOD

The classification system starts by inputting training data and inputting testing data. Then the pre-processing process consists of stop-word removal, stemming, and lemmatization. After that, the feature selected using GIT. Data that has been selected features continued with the classification process using the Naïve Bayes multinomial to produce a multinomial naive Bayes model as a model for obtaining accuracy calculations. Fig. 1 is a flowchart as an illustration of the classification system.

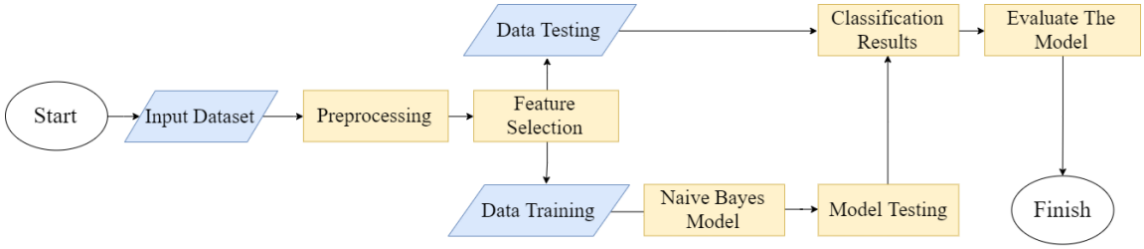

Fig. 1. Chart that illustrates the classification machine process.

A. Dataset

Dataset is an important part of machine learning, especially in email classification. Building the machine learning system, we use a dataset that has been published and labeled first as spam and ham. The dataset we used is Enron-spam [8], consisting of data Enron 1 to Enron 6. The Enron data spam processed with pre-processing techniques, so the data only consists of the subject and contents of the email. The total data from Enron-spam is 33716, used as training and testing data. The data contains 17171 spam data and 16545 ham data. The following Fig. 2 shows a comparison of data from Enron-spam.

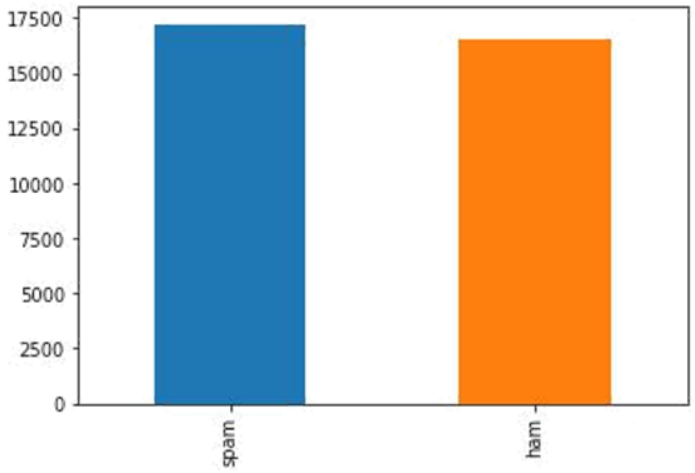

Fig. 2. illustrates the number of spam and ham emails.

B. Pre-Processing

Pre-processing is the process when the document or corpus being process to reduce unimportant words or less use of information when classified. There are several preprocessing techniques including cleaning, stopwords removal, stemming and lemmatization.

C. Cleaning and Stop-Word Removal

The cleaning process is the initial stage of pre-processing. This process is done to change the capital letters into lowercase letters and delete punctuation as well as numbers in the data. This cleaning process is useful to ease or speed up the process of classification.

Stop-Words Removal is a common word in a sentence that generally contains stop words or conjunctions etc. The word in making the model is considered as less useful information to use because it can reduce performance when machine learning is done. That is what causes the stop-word to be deleted.

The stop-word removal process is done after the tokenization process. The process of tokenization is the process of separating words in a sentence, the words are called tokens [15]. After separating words will be identified, whether it is a stop-word or not. After the stop-word is indicated, the word will be deleted. Following are examples of stop-words that will be deleted, namely, is, the, a, I, and, him, which, etc.

D. Stemming and Lemmatization

English generally contains several prefix and suffix sentences such as ed, ing, es, etc. This stemming process changes a word that has the word ending to its basic word form. For example, cats become cat, eats to eat, sleeps to sleep, studies to study.

In English, the most popular algorithm used for the stemming process is the porter algorithm [11]. This algorithm is very often used in the pre-processing for text classification. Lemmatization is almost similar to stemming. This process reduces words but uses a dictionary, not based on each suffix word such as ice, ing, ed, etc.

E. Preprocessing Result

After pre-processing, data that is not useful for classification are omitted and the size of the data will be small, making it more efficient for the model to process data. Fig. 3 and Fig. 4 show the results of pre-processing.

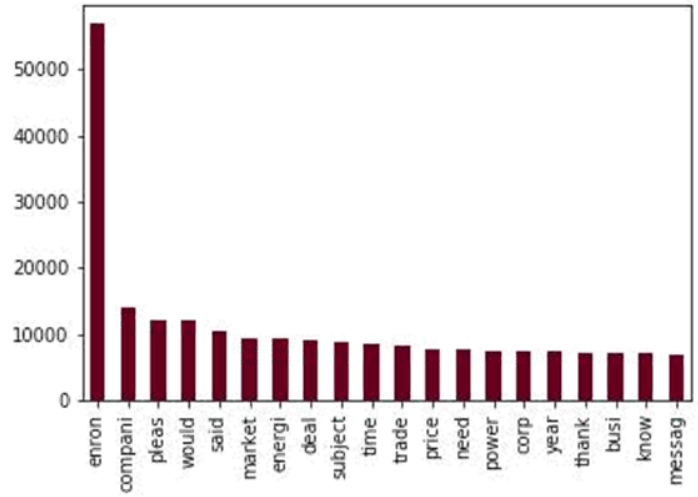

Fig. 3. 20 words frequently appear on the ham label.

Fig. 3 shows that the words most often appears in the ham label after successfully being process using cleaned, stop-word removal, stemming and lemmatization is Enron. The Enron most often appear because the Enron spam dataset contains many words of Enron company word.

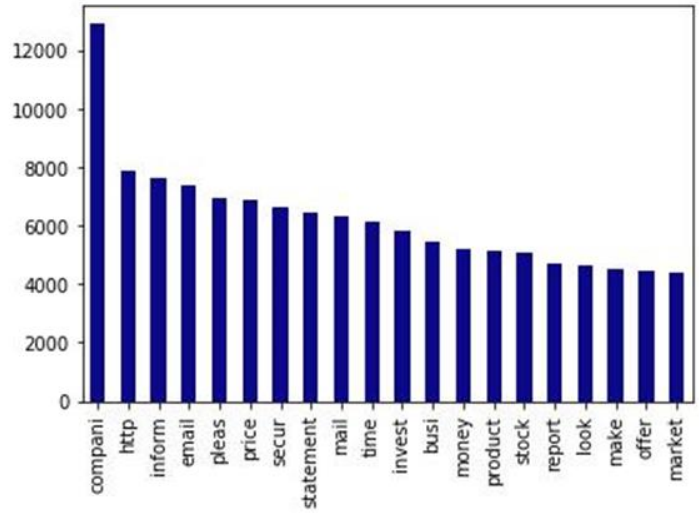

Fig. 4. 20 words frequently appear on the spam label.

Fig. 4 shows that the compani word most often appears in spam label after being the process. We can see also the most word appear is about promotion like money, price, invest, offer, etc.

F. Feature Selection

The feature selection has input in the form of training data which is Enron spam. The feature selection will be selected as the best features used in the classification process. Choosing the best features allows the model to run faster and can also add accuracy in its classification. Before this process is carried out, the pre-processed data will be formed into bag-of-words and after that, the data will be in the form of a matrix. Because the matrix derived from these bag-of-words takes up too much memory, this matrix is formed into a sparsing matrix, which

is a compressed matrix. In conducting this research the writer uses the sklearn and numpy modules to create a sparsing matrix form [16][6]. The selection of features in GIT is done by selecting the best K features taken from the maximum value. After the feature selection is performed, then a Naïve Bayes multinomial model will be built. The Naïve Bayes multinomial model is obtained from data that has been selected, which will then be tested by cross-validation and testing by testing data.

IV. Results And Discussion

In this study, the author uses 33716 data Enron spam and uses 80000 features to be performed as data training and testing. The features we used in this classification are obtained from the best results after benchmarking all the features with multiples of 5000. The evaluation was carried out using the 6 fold cross-validation technique. The cross-validation process is carried out with the initial step of choosing the best features using GIT_A, GIT_B, GIT_C, and Chi-square. A comparison with Chi-square refers to research [9], which also compares GIT feature selection with other feature selection. In the second step, the authors carry out the development of the model by using multinomial naïve bayes as a learning model. Then we will compare each accuracy and F1 score for each method.

A. Result for All Features

Doing this experiment first, we will see the results of the experiment for all features base on multiples of 5000 features. This experiment was carried out on GIT_A, GIT_B, GIT_C, and Chi-Square (CHI2). This experiment will see the accuracy and F1 score movement of the multiples of 5000 features so that the optimal number of features is obtained.

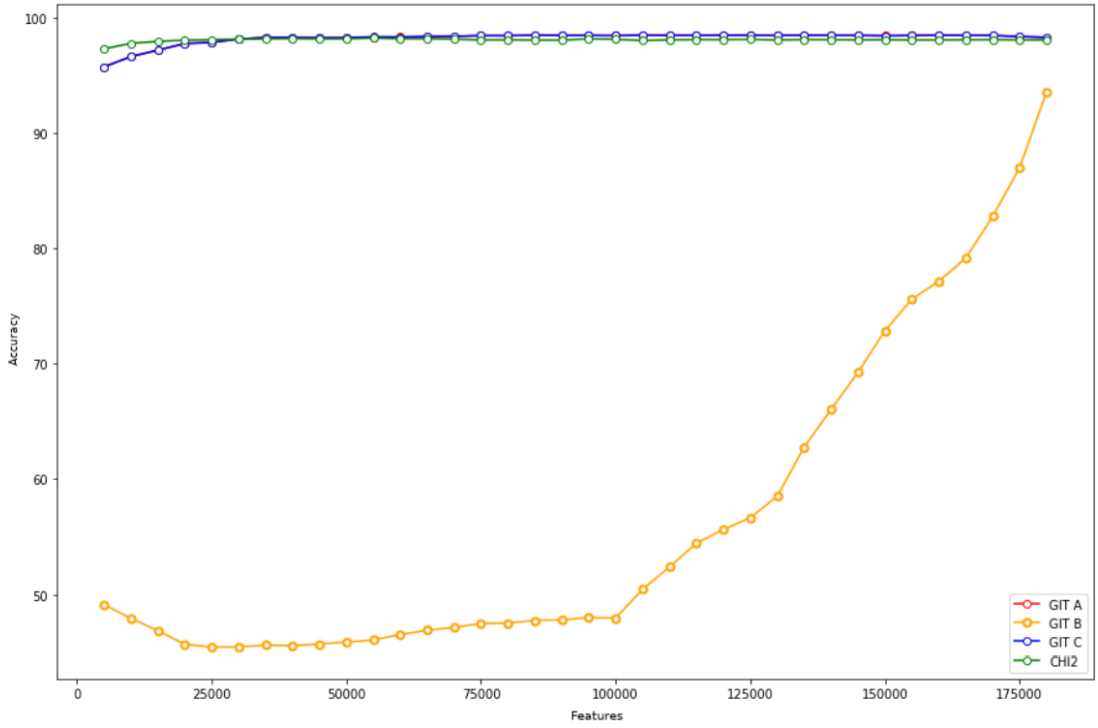

Fig. 5. Graph that shows the accuracy of all features with a threshold 0.5 on GIT_A, GIT_B and GIT_C.

Fig. 5 shows that GIT_A and GIT_C are better than GIT_B and Chi-Square (CHI2). With K-Best parameters, GIT_A and GIT_C successfully determine the optimum features when the features are 80000 with an accuracy value of 98.46%. Chi-Square (CHI2) when compared to GIT_B get better results and get optimum results when it reaches the number of features 60000 with a value of 98.20%.

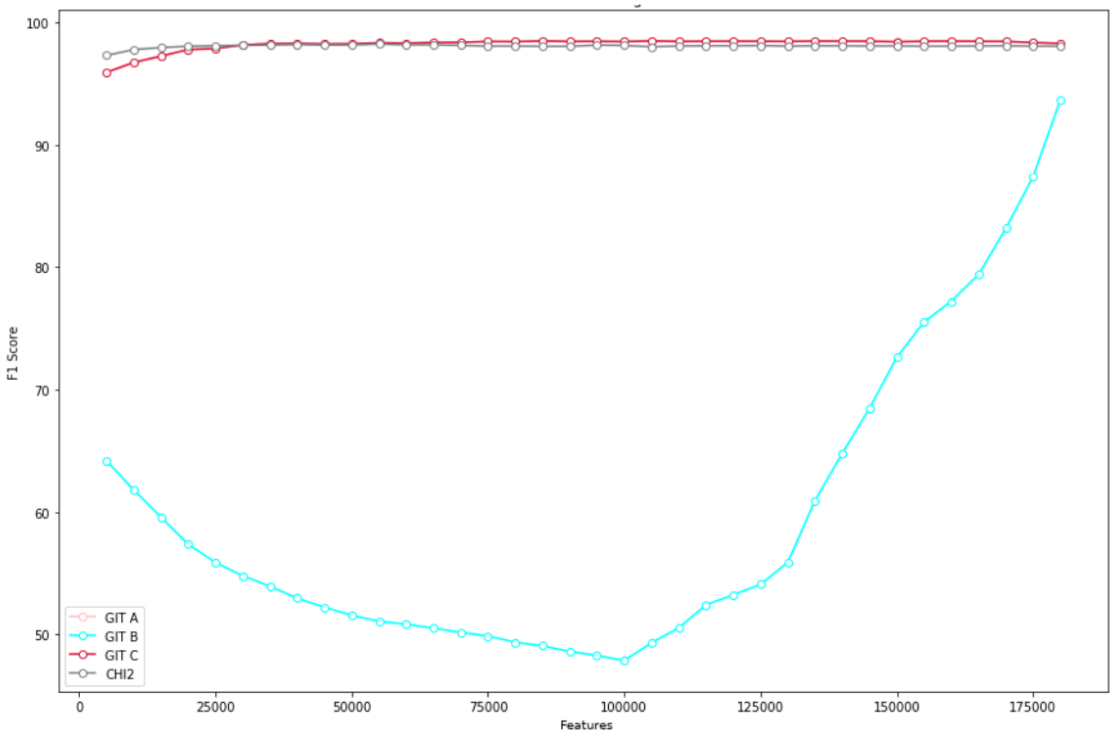

Fig. 6. A graph showing the results of the F1 score of all features with a threshold of 0.5 on GIT_A, GIT_B and GIT_C.

In Fig. 6, we can see the results of the F1 score from the benchmark multiple of 5000. Based on the results of the calculation, the results achieved are highest by using the feature selection GIT_A and GIT_C when compared with GIT_B and Chi-Square (CHI2). The calculation shows the maximum score when it is vulnerable from 75000 to 125000. While Chi-Square experiences a maximum peak when the feature is between 50000 and 75000.

B. Experiment Result in 6 Fold Cross-Validation

In this experiment, we use only 80000 features, obtained from the most optimum number of features. This experiment uses a 6 fold cross-validation technique to test how well the model is built. Cross-validation techniques are commonly used to test how well a model is used on a model based on the average of the result F1 score and accuracy score.

TABLE I
THE RESULTS OF THE CALCULATION OF THE AVERAGE ACCURACY USING 6 FOLD CROSS-VALIDATION

MNNB	GIT_A	GIT_B	GIT_C	Chi-Square	Without Feature Selection
Fold 1	98.54%	47.56%	98.45%	98.20%	97.95%
Fold 2	98.33%	48.54%	98.56%	97.95%	97.83%

Fold 3	98.38%	46.50%	98.31%	97.97%	97.94%
Fold 4	98.61%	48.00%	98.26%	98.01%	98.24%
Fold 5	98.31%	47.18%	98.38%	97.81%	98.10%
Fold 6	98.33%	47.37%	98.52%	98.15%	98.10%
Average	98.42%	47.52%	98.40%	98.01%	98.03%
Number of Feature	80000	80000	80000	80000	184161

From Table I can be seen that GIT_A and GIT_C successfully selected features and reduced features that were less representative to use with MNNB. GIT_A received 98.61% with the highest achievement being in the 4th fold. The GIT_A result obtained by applying a threshold of 0.5. As seen in the table there is also GIT_C which shows a second good result when it is on the 2nd fold, namely 98.56%, with the same threshold as GIT_A. When calculating the average accuracy between folds, GIT_A gets a higher result compared to the others and for GIT_C get second place. The smallest result that is equal to 46.50% obtained from GIT_B. GIT_B and Chi-Square when used 80000 features get poor results when applied and cannot exceed results without feature selection.

TABLE II
THE RESULTS OF THE CALCULATION OF THE AVERAGE OF F1 SCORE USING 6 FOLD CROSS-VALIDATION

MNNB	GIT_A	GIT_B	GIT_C	Chi-Square	Without Feature Selection
Fold 1	98.57%	49.56%	98.48%	98.22%	97.98%
Fold 2	98.39%	50.80%	98.61%	97.96%	97.89%
Fold 3	98.36%	48.74%	98.33%	98.02%	97.96%
Fold 4	98.66%	49.67%	98.27%	98.03%	98.28%
Fold 5	98.33%	48.51%	98.39%	97.86%	98.06%
Fold 6	98.32%	48.97%	98.55%	98.17%	98.13%
Average	98.44%	49.37%	98.44 %	98.03%	98.05%
Number of Feature	80000	80000	80000	80000	184161

From Table II we can see the 4th fold is the highest result, 98.66%, achieved by GIT_A. Also GIT_C 98.61% at 2nd fold. In the table, we can see the average GIT_A and GIT_C have the same value equal to 98.44%. We found that the average can exceed the naïve bayes multinomial without using feature selection which using only 80000 features. We found again just only using 80000 features, GIT_B and Chi-Square get worse results compared to scenario without using the feature selection, which F1 score is 49.37% and 98.03%.

V. CONCLUSION

The study has empirically proved that Multinomial Naïve Bayes combined with Complete Gini-Index Text can improve the accuracy of the classification. The calculation of accuracy and F1 Score have been done, obtained results with different values for each GIT_A, GIT_B, and GIT_C. The best accuracy results obtained by GIT_A, 98.61%, and have a difference of 0.37% compared to without a feature selection. In term of F1 Score, GIT_A also has the highest score with score of 98.66%, on the 4th fold. GIT_A is compared to Chi-Square, the GIT_A has better accuracies, with difference of 0.41%. GIT_A also has better F1 Score than Chi-Square, with difference of 0.44%.

REFERENCES

[1] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," *IEEE Access*, vol. 7, pp. 168261–168295, 2019.
[2] O. Saad, A. Darwish, and R. Faraj, "A survey of machine learning techniques for Spam filtering," *Int. J. Comput. Sci. Netw. Secur.*, vol. 12, no. 2, p. 66, 2012.
[3] H. Park, S. Kwon, and H.-C. Kwon, "Complete gini-index text (git) feature-selection algorithm for text classification," in *The 2nd international conference on software engineering and data mining*, 2010, pp. 366–371.
[4] W. Gad and S. Rady, "Email filtering based on supervised learning and mutual information feature selection," in *2015 Tenth International Conference on Computer Engineering \& Systems (ICCES)*, 2015, pp. 147–152.
[5] R. Aggarwal and N. Singh, "A New Hybrid Approach for Network Traffic Classification Using Svm and Naïve Bayes Algorithm," 2017.
[6] J. Yang, Y. Liu, Z. Liu, X. Zhu, and X. Zhang, "A new feature selection algorithm based on binomial hypothesis testing for spam filtering," *Knowledge-Based Syst.*, vol. 24, no. 6, pp. 904–914, 2011.
[7] S. R. Gomes *et al.*, "A comparative approach to email classification using Naive Bayes classifier and hidden Markov model," in *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, 2017, pp. 482–487.
[8] "Enron-Spam dataset | Kaggle." [Online]. Available: https://www.kaggle.com/wanderfj/enron-spam. [Accessed: 03-May-2021].
[9] J. Sun, X. Zhang, D. Liao, and V. Chang, "Efficient method for feature selection in text classification," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.
[10] B. Issac, W. J. Jap, and J. H. Sutanto, "Improved Bayesian anti-spam filter implementation and analysis on independent spam corpuses," in *2009 International conference on computer engineering and technology*, 2009, vol. 2, pp. 326–330.
[11] M. Singh, "Classification of spam email using intelligent water drops algorithm with naive bayes classifier," in *Progress in Advanced Computing and Intelligent Engineering*, Springer, 2019, pp. 133–138.
[12] T. M. Ma, K. YAMAMORI, and A. Thida, "A Comparative Approach to Na{\"\i}ve Bayes Classifier and Support Vector Machine for Email Spam Classification," in *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, 2020, pp. 324–326.
[13] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[14] K. S. Jones and P. Willett, *Readings in information retrieval*. Morgan Kaufmann, 1997.
[15] A. S. Manisha and M. D. R. Jain, "Data pre-processing in spam detection," *Int. J. Sci. Technol. Eng.*, vol. 1, no. 11, p. 5, 2015.
[16] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?," in *CEAS*, 2006, vol. 17, pp. 28–69.