

Automatic Complaint Classification System Using Classifier Ensembles

M. Ali Fauzi

Abstract—Sambat Online is an online complaint system run by the city government of Malang, Indonesia. Because most citizens do not know to which work units (Satuan Kerja Pemerintah Daerah [SKPDs]) their complaints should be sent, the system administrator must manually sort and classify all of the incoming complaints with respect to the appropriate SKPDs. This study empirically evaluated the application of an automated system to replace the manual classification process. The experiments, which used Sambat Online data, involved five individual classification algorithms— Naïve Bayes, Maximum Entropy, K-Nearest Neighbors, Random Forest, and Support Vector Machines—and two ensemble strategies—hard voting and soft voting. The results show that the Multinomial Naïve Bayes classifier achieved the best performance, an 80.7% accuracy value, of the five individual classifiers. The results also indicate that generally all of the ensemble methods performed better than the individual classifiers. Almost all of them had the same accuracy level of 81.2%. In addition, the soft voting strategy had slightly higher accuracy than the hard one when all five classifiers were used. However, when the three best classifier combinations were used, both had the same level of accuracy.

Keywords — Ensemble Learning, E-Government, Machine Learning, Hard Voting, Soft Voting, Complaint classification.

I. INTRODUCTION

THE rapid development of information and communication technologies (ICT) has been changing not only daily life but also the interactions between citizens and governments. The use of ICT in government in the form of e-government is currently very popular in Indonesia. The city of Malang is an example. One of the e-government implementations in Malang is Sambat Online, an online complaint system. Sambat Online provides a platform for citizens to share their aspirations, to ask questions, to offer suggestions, to make criticisms, and to give their opinions about the problems experienced in Malang.

The city's Dinas Komunikasi dan Informasi (Communication and Information Department [Diskominfo]) has served as the Sambat Online system administrator. Because most citizens do not know to which work units (Satuan Kerja Pemerintah Daerah [SKPDs]) their complaints should be sent, the system administrator must sort and classify all the incoming complaints manually and forward them, manually also, to the appropriate

SKPDs. This work is very expensive and time-consuming. The system administrator has to spend a great deal of time reading and classifying the large number of incoming complaints.

The rapidly growing number of digital unstructured text documents available online has created the need for text mining techniques, which have been gaining in popularity, to perform flexible analyses of the content in these documents [1]. These techniques have been applied in many areas, including news classification (e.g., [2]–[3]), literature searches (e.g., [4]–[5]), essay grading (e.g., [6]–[7]), sentiment analyses (e.g., [8]–[10]), medical research (e.g., [11]–[12]), and forensic science [13]. One of the most well-known text mining techniques is automatic text classification using machine learning. Text classification can be defined as the prediction of one or more suitable categories from a predefined set of unstructured texts written in a natural language (e.g., English or Bahasa). Text classification is currently a major research area with applications for many fields.

In the approach taken in this study, the Sambat Online situation was considered a text classification problem. The most used text classification technique has been the bag-of-words (BOW) model. Features are represented in a vector space model (VSM) and subsequently classified by traditional machine-learning methods, such as the Naïve Bayes (NB) [14], Maximum Entropy (ME) [15], K-Nearest Neighbors (KNN) [16], Random Forest (RF) [17–18], or Support Vector Machines (SVM) [19] classifiers. However, most single classifiers have their own inherent defects [20], and their performance is also domain-dependent [21].

An ensemble of classifiers is a set of base classifiers whose individual outputs are combined in some way to classify new documents in order to enhance classification accuracy [22]. Several experiments using ensemble techniques have been conducted on topical text classifications. The results have shown that ensembles can improve classification accuracy (e.g., [12], [23]–[24]).

In this study, an automatic complaint classification system was built for Sambat Online. The categories were each of the Malang city government's SKPDs. This work was motivated by the desire to automate the system administrator's process for classifying and forwarding complaints to the appropriate SKPDs. The ensemble methods of several popular classifiers, including the NB, KNN, ME, RF, and SVM, were used. Because only a small amount of training data was available, Neural Network method was not used even though it is the most popular approach to document classification. This small amount of training data was insufficient for the proper training of a

Paper received February 1, 2018; revised July 24, 2018; accepted August 29, 2018. Date of publication December 25, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vlado Delić.

M. Ali Fauzi is with the Faculty of Computer Science, Brawijaya University, Malang, Indonesia (telephone: +6285746749125; e-mail: moch.ali.fauzi@ub.ac.id).

neural network. The goal was to conduct an intensive study of the effectiveness of ensemble techniques for classifying documents. However, this study is only the initial step in developing a high-accuracy document classification system specifically for Sambat Online documents. Several additional steps will need to be taken to achieve that goal.

II. TEXT CLASSIFICATION

Text classification is the task of assigning free texts or documents to one or more categories created from a predefined set. Text classification is currently a major research area with applications in many fields. It is also known as text categorization, document classification, and even topic spotting. The most common approach to text classification is machine learning methods. First, the machine learning was trained with training data. It was then used for predicting the category for each tested data point. The following sections provide a brief description of the general text classification process.

A. Document Preprocessing

Document preprocessing involves tokenization, filtering, and stemming. Tokenization is the transformation of documents into smaller units: tokens or terms. This process includes case folding and the removal of punctuation, numbers, and non-alphabetic characters. The next process is filtering: stopword removal. Filtering removes the uninformative words related to the existing stopword dictionary. The final process is stemming: converting every word or term to its root. Affixes, such as prefixes, infixes, and suffixes, are removed. For example, the words “membeli,” “dibeli,” “pembeli,” and “pembelian” are converted to the same word: “beli.”

B. Document Representation

After the documents have been preprocessed, each document is then modelled. A commonly used model for document classification is Vector Space Model (VSM), which represents each document as a feature vector in space terms [25]. The space has a dimension, the number of unique terms in the document collection, because the terms in a VSM are taken from all of the unique terms in the documents. The vector of each document is represented through a term weighting method, the most popular being term frequency–inverse document frequency (TF-IDF) [26]. The TF-IDF is the multiplication of the term frequency (TF) by the inverse document frequency (IDF). The TF gives each term a weight proportional to the number of its occurrences in the document, and the IDF considers the distribution of the term throughout all the documents. With the IDF, the value of each term is assumed to be inversely proportional to the total number of documents containing the term. The weight combination for term t in document d can be calculated as follows:

$$TF.IDF(t, d) = (1 + \log(f_{t,d})) \times \left(1 + \log \frac{N_d}{df_t}\right) \quad (1)$$

Where $f_{t,d}$ is the number of the term t occurrence in document d , N_d is the number of documents in the corpus, and df_t is the number of documents containing term t in the corpus.

C. Machine Learning

Machine learning is a field of computer science that enables computers to learn automatically from previous data or experiences without being explicitly programmed. Traditional machine-learning techniques, such as the NB, KNN, ME, RF, and SVM classifiers, have been widely used, and they have been shown to perform well in text classification.

D. NB Classifier

The NB classifier is widely used for document classification. This method is preferred because of its speed, simplicity, and effectiveness [27]–[28]. The NB classifier, which is based on the Bayes theorem, was created by using training data to estimate the posterior probability of each category of the test data. The posterior probability of a class, for example class j , given the test document d , can be simplified as follows:

$$P(c_j|d) \approx P(c_j) \times P(d|c_j) \quad (2)$$

where $P(c_j|d)$ is the posterior probability of class j , $P(c_j)$ is the prior probability of class j , and $P(d|c_j)$ is the conditional probability. Using the BOW framework, document d with a number of words, n , is represented by $[w_1, w_2, \dots, w_n]$ where w_k denotes the k th word in the document. Because the NB classifier assumes that the words in a document are mutually independent, the conditional probabilities can be computed as follows:

$$P(d|c_j) = P([w_1, w_2, \dots, w_n]|c_j) \approx \prod_{k=1}^n P(w_k|c_j) \quad (3)$$

The NB category decision can therefore be represented as:

$$c^* = \arg \max_{j=1, \dots, c} P(c_j) \prod_{k=1}^n P(w_k|c_j) \quad (4)$$

There are two NB probabilistic models that are commonly used for text classification: the multinomial model and the multivariate Bernoulli model. McCallum and Nigam [29] found that the multinomial logistic regression model yielded better results than the multivariate Bernoulli; therefore, the multinomial model was used in this study.

E. KNN Classifier

The KNN classifier is a case-based learning algorithm. The test document is classified according to the category that accounts for a majority of its k nearest neighbors. A document’s neighbors are the most similar training documents. The similarities between two documents can be computed using the inverse of the Euclidian distance of their feature vectors.

The method for making the category decisions regarding the test document is generally majority voting between the k most similar training documents. In majority voting, all neighbors can be treated equally. However, there are modification methods that use other types of category decisions. The fuzzy-KNN classifier, which uses fuzzy methods to determine the probability of each category, is an example. The KNN classifier has often been applied because of its effectiveness and usability. However, the computation time is lengthy, and finding the optimal value of K is not easy.

F. ME Classifier

The ME, a probabilistic classifier, belongs to the family of exponential models. Unlike the NB classifier, it does not assume that the features or words in a document are conditionally independent of one another. The ME classifier is based on the principle of selecting the probability distribution that produces the largest entropy consistent with the task's constraints, which are derived from the training process. ME classifiers have been widely used to solve many types of text classification problems.

With the ME classifier, the conditional distribution of the class c , given a document d , can be computed as follows [30]:

$$P_{\lambda}(c|d) = \frac{1}{Z(d)} \exp\{\sum_i \lambda_i f_i(d, c)\} \quad (5)$$

where $Z(d)$ is a normalization function and λ_i is the weight coefficient of constraint derived from the feature function $f_i(d, c)$. The feature function can be described as follows:

$$f_i(d, c) = \begin{cases} 1, & d = d_i \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

The model with ME is the one in the parametric family $P_{\lambda}(c|d)$ that maximizes the likelihood of class and document.

G. RF Classifier

The RF classifier is an ensemble learning method based on the decision tree model [31]. The RF classifier grows a forest with many classification trees. Each tree is trained on a bootstrapped sample of training data, and at each candidate split in the learning process, a random subset of the features is selected. To classify an input vector in the RFs, the vector has to be entered as an input to each tree in the forest. Each tree classifies the input vector. The classification class is the vote of each tree, and the forest chooses the class that has the most votes.

H. SVM Classifier

SVM are a set of supervised learning methods that can be used for classification or regression. Because of their good performance, they are among the most popular algorithms used in classification [32]. The only disadvantage of these methods is their high computational complexity [33].

Given a training dataset with each data point labelled as belonging to one or the other of two categories, an SVM classifier finds the widest possible gap in the data for the separate categories. In other words, the SVM classifier constructs a hyperplane in a high-dimensional space with the maximum margin that separates a set of two-category instances. These instances establishing a hyperplane are support vectors.

SVM classifiers are suitable for binary classifications; however, they can be applied to multiclass classifications using one-against-one or one-vs-the-rest approaches. The classification of a new document for the one-versus-the-rest approach is done by a winner-takes-all strategy in which the classifier with the highest output function assigns the category. For the one-versus-one approach, classification is done by a max-wins voting strategy in which every classifier sets the document to one of the two categories. The vote for the set category is then increased by one vote, and last, the category with the most votes decides the document classification.

I. Classifier Ensemble

Classifier ensemble methods train multiple classifiers and combine them to solve the same problem [34]–[35]. They offer some advantages especially because the risk of selecting a particularly poorly performing classifier can be reduced by combining a number of different classifiers [36]. It is important to note that using an ensemble of some of the classifiers rather than the best individual classifier cannot guarantee higher accuracy. Combining classifiers may not necessarily overcome the accuracy of the best classifier in the ensemble; however, it would undoubtedly lower the risk of selecting a poor classifier for classifying new data [37].

III. THE APPROACH FOR THIS STUDY

The classification of Sambat online in this study consisted of three main stages: preprocessing, training the classifiers, and combining the classifiers (see Figure 1). The first stage involved several steps, including tokenization, filtering, and stemming. This stage produced BOW features. In the second stage, some classifiers, including the NB, ME, KNN, and RF classifiers, and the SVM, were trained with training data using the BOW as their features. For the NB classifier, three kinds of distributions were used: the Bernoulli NB, multinomial NB, and Gaussian NB. For the SVMs, the linear, polynomial, sigmoid, and radial basis function (RBF) kernels were used. Last, the classifiers were combined for classifying the testing data.

In the combination stage, two kinds of ensemble methods, hard and soft voting, were used. With the hard voting method, each individual classifier had one vote. As can be seen in Figure 2, the category for a document is the majority category that was predicted by each individual classifier. With the soft voting method, the average predicted probabilities were used. The predicted category probabilities for each classifier were collected and averaged, and the final category for a document was then derived from the category with the highest average probability. In the current study, machine learning was achieved with Scikit-learn [38].

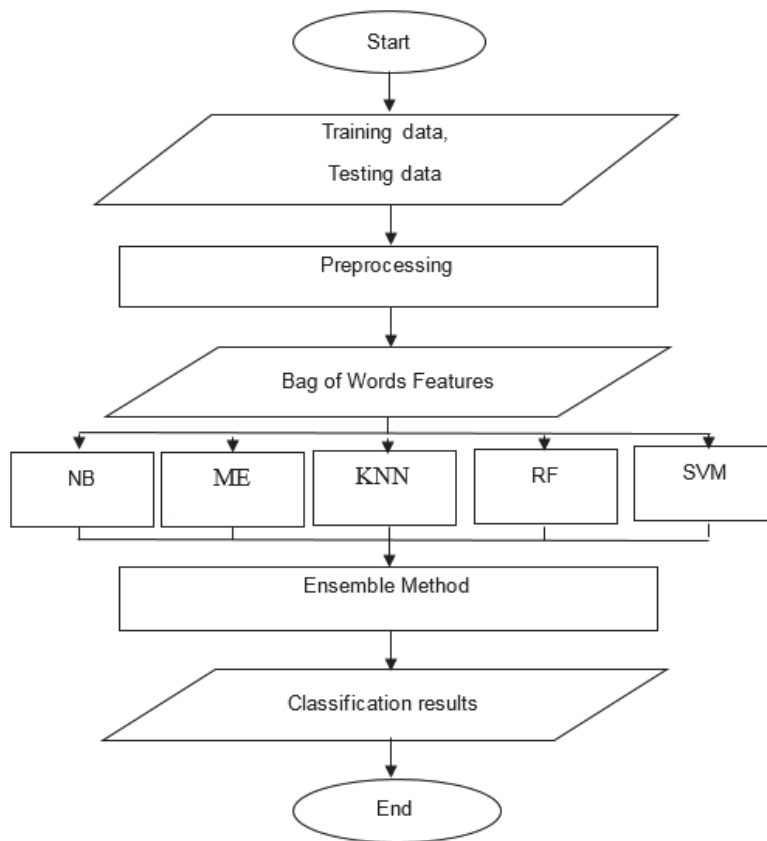


Fig. 1. Main System Flowchart.

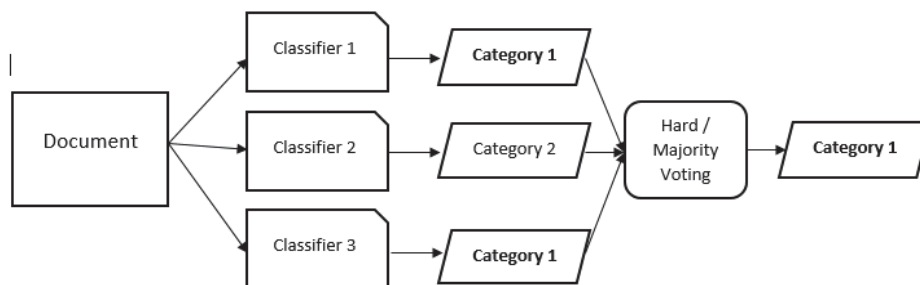


Fig. 2. Hard Voting Method.

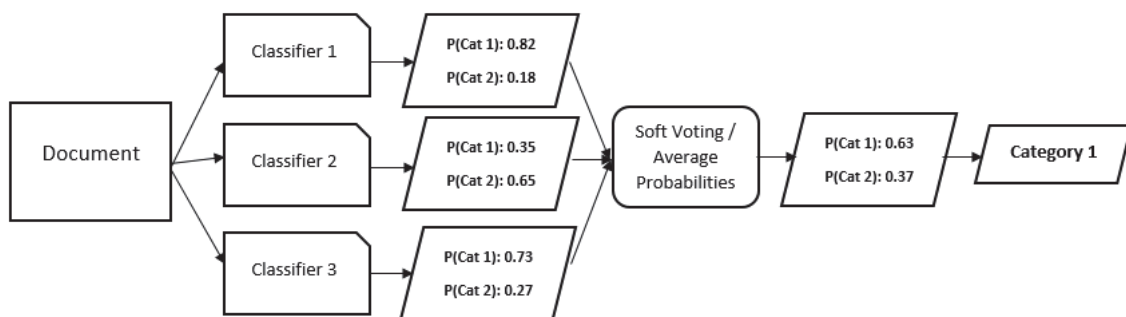


Fig. 3. Soft Voting Method.

IV. RESULTS AND DISCUSSION

The dataset used in this experiment was complaint documents from the Sambat Online system. There were 204 documents from three SKPDs, the categories. The dataset was imbalanced because it was composed of 99 documents from the first category, 40 from the second, and 65 from the third. This experiment was conducted to determine the effect of an ensemble method on classification performance. Because 5-fold cross validation was used, the data were divided equally into five folds. In each iteration of the cross validation, documents from four of the folds were used as training data, and the documents in the one remaining fold were used as testing data. The evaluation method used in this experiment was the average classification accuracy. The results are presented in Table 1.

Of the NB variations, the multinomial NB classifier yielded the highest accuracy: 80.7%. The Gaussian NB classifier had the lowest accuracy, 65.5%, similar to the accuracy, 67.7%, of the Bernoulli NB. These results are in line with those reported by McCallum and Nigam [29]. Of the SVM kernel variations, the linear kernel-based SVM had the highest accuracy: 78.2%. In contrast, the SVM classifiers with polynomial, sigmoid, and RBF kernels yielded similarly poor results: 48.5% accuracy. This is also in accordance with the results reported by Yang et al [39].

TABLE 1: EXPERIMENT RESULTS.

Classification Method	Accuracy (%)
NB	
Bernoulli NB	67.7
Multinomial NB	80.7
Gaussian NB	65.5
ME	74.2
KNN	74.2
RF	75.8
SVM	
Linear Kernel	78.2
Polynomial Kernel	48.5
Sigmoid Kernel	48.5
RBF Kernel	48.5
Ensemble Method Using 5 Classifiers	
Hard Voting	78.7
Soft Voting	81.2
Ensemble Method Using 3 Best Classifiers	
Hard Voting	81.2
Soft Voting	81.2

Of the individual classifiers, the multinomial NB had the best performance: 80.7% accuracy. The performance of the linear SVM classifiers was slightly inferior to that of the NB classifier: 78.2%. It is clear that the KNN and ME were the worst performing classifiers, with accuracy values of only 74.2%. The RF classifier performed slightly better than the lowest two classifiers: 75.8% accuracy.

The ensemble methods used in this study were the hard and soft voting strategies. Almost all of the ensemble

methods performed slightly better than the individual classifiers although on the hard voting strategy with five classifiers (the multinomial NB, ME, KNN, RF, and linear SVM), the performance of the ensemble methods was still surpassed by that of the multinomial NB classifier. The decision in hard voting was affected equally by all of the individual classifiers. Because the difference in accuracy between the best classifier (80.7%) and the two worst classifiers (74.2%) was too great, the hard voting accuracy tended to vary, and this contributed to the poorer performance compared to that of the best classifier. This did not occur when only the three best classifiers (the multinomial NB, RF, and linear SVM) were used for the hard voting strategy or all five classifiers were used with the soft voting strategy. The hard voting strategy with the three best classifiers and the soft voting strategy with all five classifiers yielded similar results: 81.2% accuracy. This performance surpassed that of the best individual classifiers even though the difference was not remarkable. However, it did lower the risk of selecting a poor classifier to be used for classifying new data.

In this experiment, the use of the two alternative ensemble strategies, hard voting and soft voting, yielded different results only when all five classifiers were used. When only the three best classifiers were used, the soft voting and the hard voting strategies produced similar results: 81.2% accuracy.

V. CONCLUSION

This study evaluated the application of automatic document classification to a complaint management system to replace the manual process used by the Malang Diskominfo to classify complaints by the SKPDs. Five individual classification algorithms and two ensemble strategies were applied to the Sambat Online dataset. The experiment results show that the multinomial NB classifier offered the best performance, 80.7% accuracy, of the five individual classifiers, including the ME, KNN, RF, and SVM. The results also indicate that an ensemble technique is effective for combining different classification algorithms to achieve better classification performance. Almost all of the ensemble methods offered an improvement (81.2%) over the individual classifiers. Even though the improvement was not remarkable, the risk of selecting a poor classifier for classifying new data was lowered. The alternative ensemble strategy used in this study, hard and soft voting, produced different results only when all five classifiers were used. When only the three best classifiers were used, the soft voting and the hard voting strategies yielded similar results: 81.2% accuracy.

Future studies should use the ensembles of the feature set instead of relying on just the BOW features. This could be a promising direction for achieving improvements; however, it could create a new problem: specifically, high dimensionality. Therefore, feature selection is an important issue worthy of additional research.

ACKNOWLEDGMENT

We thank Dinas Komunikasi dan Informasi Pemerintah Kota Malang for their cooperation.

REFERENCES

- [1] M. A. Hearst, "Untangling text data mining," *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pp. 3-10, 1999.
- [2] M. A. M. García, R. P. Rodríguez, M. V. Ferro, and L. A. Rifón, "Wikipedia-Based Hybrid Document Representation for Textual News Classification," *Proceedings of the 3rd International Conference on Soft Computing & Machine Intelligence (ISCM)*, pp. 148-153, 2016.
- [3] K. Watanabe, "Newsmap: A semi-supervised approach to geographical news classification," *Digital Journalism*, pp. 1-16, 2017.
- [4] M. A. Fauzi, A. Z. Arifin, and A. Yuniarti, "Arabic Book Retrieval Using Class and Book Index Based Term Weighting," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 6, pp.3705-3710, 2017.
- [5] M. A. Fauzi, A. Arifin, and A. Yuniarti, "Term Weighting Berbasis Indeks Buku dan Kelas untuk Peringkatan Dokumen Berbahasa Arab," *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, vol 5, no. 2, pp. 435-442, 2014.
- [6] E. S. Pramukantoro and M. A. Fauzi, "Comparative analysis of string similarity and corpus-based similarity for automatic essay scoring system on e-learning gamification," *Proceedings of the 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 149-155, 2016.
- [7] M. A. Fauzi, D. C. Utomo, E. S. Pramukantoro, and B. D. Setiawan, "Automatic Essay Scoring System Using N-Gram and Cosine Similarity for Gamification Based E-Learning," *Proceedings of the International Conference on Advances in Image Processing (ICAIP)*, pp. 151-155, 2017.
- [8] N. Shelke, S. Deshpande, and V. Thakare, "Domain independent approach for aspect oriented sentiment analysis for product reviews," *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*, pp. 651-659, 2017.
- [9] J. Wehrmann, W. Becker, H. E. Cagnini, and R. C. Barros, "A character-based convolutional neural network for language-agnostic Twitter sentiment analysis," *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2384-2391, 2017.
- [10] M. A. Fauzi, R. F. N. Firmansyah, T. Afrianto, "Improving sentiment analysis of short informal Indonesian product reviews using synonym based feature expansion," *Telkonnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 3, pp. 1345-1350, 2018.
- [11] A. N. Nguyen, M. J. Lawley, D. P. Hansen et al., "Symbolic rule-based classification of lung cancer stages from free-text pathology reports," *Journal of the American Medical Informatics Association*, vol. 17, no. 4, pp. 440-445, 2010.
- [12] J. J. G. Adeva, J. M. P. Atxa, M. U. Carrillo, and E. A. Zengotitabengoa, "Automatic text classification to support systematic reviews in medicine," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1498-1508, 2014.
- [13] O. D. Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for author identification forensics," *ACM Sigmod Record*, vol. 30, no. 4, pp.55-64, 2001.
- [14] I. Pop, "An approach of the Naive Bayes classifier for the document classification," *General Mathematics*, vol. 14, no. 4, pp. 135-138, 2016.
- [15] A. M. El-Halees, "Arabic text classification using maximum entropy," *IUG Journal of Natural Studies*, vol. 15, no. 1, pp. 157-167, 2015.
- [16] A. K. Nikhath, K. Subrahmanyam, and R. Vasavi, "Building a K-Nearest Neighbor Classifier for Text Categorization," *International Journal of Computer Science and Information Technologies* vol. 7, no. 1 pp. 254-256, 2016.
- [17] Q. Wu, Y. Ye, H. Zhang, M. K. Ng, and S. Ho, "ForesTexter: an efficient random forest algorithm for imbalanced text categorization," *Knowledge-Based Systems*, vol. 67, pp. 105-116, 2014.
- [18] M. A. Fauzi, "Random Forest Approach for Sentiment Analysis in Indonesian Language," *Indonesian Journal of Electrical Engineering and Computer Science* vol. 12, no.1. 2018
- [19] B. S. Kumar and V. Ravi, "Text Document Classification with PCA and One-Class SVM," *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*, pp. 107-115, 2017.
- [20] L. Li, Y. Zhang, L. Zou et al., "An ensemble classifier for eukaryotic protein subcellular location prediction using gene ontology categories and amino acid hydrophobicity," *PLoS One* vol. 7, no. 1, Article ID e31057.
- [21] B. S. Kumar and V. Ravi, "Text Document Classification with PCA and One-Class SVM," *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*, pp. 107-115, 2017.
- [22] J. Kittler, "Multiple classifier systems," *Soft Computing Approach to Pattern Recognition and Image Processing*, pp. 3-22, 2002.
- [23] L. S. Larkey, and W. B. Croft, "Combining classifiers in text categorization," *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 289-297, 1996.
- [24] Y. Dong, and K. Han, "A comparison of several ensemble methods for text categorization," *Proceedings of the 2004 IEEE International Conference on Services Computing (SCC 2004)*, pp. 419-422, 2004.
- [25] K. F. H. Holle, A. Z. Arifin, and D. Purwitasari, "Preference based term weighting for arabic fiqh document ranking," *Jurnal Ilmu Komputer dan Informasi*, vol. 8, no. 1, pp. 45-52, 2015.
- [26] G. Salton, and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513-523, 1988.
- [27] A. Goel, J. Gautam, and S. Kumar, "Real time sentiment analysis of tweets using Naive Bayes," *Proceedings of the 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pp. 257-261, 2016.
- [28] M. A. Fauzi, A. Z. Arifin, and S. C. Gosaria, "Indonesian News Classification Using Naive Bayes and Two-Phase Feature Selection Model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 8, no. 3, pp. 610-615, 2017.
- [29] A. McCallum, and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI-98 workshop on learning for text categorization*, vol. 752, pp. 41-48, 1998.
- [30] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Computational linguistics*, vol. 22, no. 1, pp. 39-71, 1996.
- [31] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [32] B. Liu, "Web data mining: exploring hyperlinks, contents, and usage data," *Springer Science & Business Media*, 2007.
- [33] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121-167, 1998.
- [34] Z. Zhou, "Ensemble methods: foundations and algorithms," *CRC press*, 2012.
- [35] M. A. Fauzi, A. Yuniarti, "Ensemble method for indonesian twitter hate speech detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 1, pp. 294-299, 2018.
- [36] T. G. Dietterich, "Ensemble methods in machine learning," *International workshop on multiple classifier systems*, pp. 1-15, 2000.
- [37] N. F. F. Da Silva, E. R. Hruschka, and E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems* vol. 66, pp. 170-179, 2014.
- [38] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
- [39] Y. Yang, and X. Liu, "A re-examination of text categorization methods," *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 42-49, 1999.