Sentiment Analysis of Hate Speech on Twitter Public Figures with AdaBoost and XGBoost Methods

Daffa Ulayya Suhendra, Jondri, Indiwarti*
Fakultas Informatika, Universitas Telkom, Bandung, Indonesia

Abstract—Public figures are often scrutinized by social media users, either because of what they say or even because of their role in a television series. Generally, public figures upload something on their social media accounts to help shape their image. But not everyone who sees it is happy. Some even dislike the upload. This study aims to determine public sentiment towards public figure Anya Geraldine conveyed on Twitter in Indonesia. The classification process in this study uses the Adaptive Boosting (AdaBoost) and Extreme Gradient Boosting (XGBoost) classification methods with text preprocessing using cleaning, case folding, tokenization, and filtering. The data used are tweets in Indonesian with the keyword "@anyaselalubenar", with a total dataset of 7,475 tweets divided into 6,887 positive and 588 negative tweets. From the label results using oversampling to avoid excessive overfitting problems. The feature used is TF-IDF weighting. Four experimental scenarios were carried out to validate the effectiveness of the model used: first model performance without oversampling, second model performance with oversampling, third model performance with undersampling, and fourth model performance with Hyperparameter tune. The experimental results show that XGBoost+SMOTE+Hyperparameter achieved 95% compared to AdaBoost+SMOTE+Hyperparameter of 87%. The application of SMOTE and Hyperparameter tune is proven to overcome the problem of data imbalance and get better classification results.

Keywords: Twitter; Hate Speech; Sentiment; Analysis; AdaBoost; XGBoost

1. INTRODUCTION

Hate Speech is a word or utterance of hatred. Hate speech in the time before the internet was spoken directly to the hated person, but along with the times and Information Technology, hate speech can be expressed in many media [1]. The presence of the internet should be used to obtain information, build relationships with other users and expand relationships between users [2]. However, not all of these conveniences tremendously influence all internet users. The notion of "this is my social media, whatever I want to say!” often triggers conflicts between internet users. The use of profane language is daily in posts intended to attack certain parties or just for fun. If blasphemous words are easy to find on the internet, it can affect how users, especially teenagers, think that using profane words in everyday life is not a problem [3]. Abusive words are expressions that include harsh or dirty words, both spoken and written. The widespread use of offensive words on the internet and social networks is that there are no practical tools to filter the use of abusive words. There is a lack of empathy among internet users and parental supervision [3].

Twitter media is a popular online media in Indonesia. Many public figures become famous from social media. Every controversy caused by public figures raises the pros and cons of netizens on social media such as Twitter [4]. Indonesian model and artist Anya Geraldine was recently sought after by netizens and became more popular when she played the third person in the mini-series "Layangan Putus". From this information, some netizens on Twitter triggered words containing hate speech toward Anya. The number of tweets scattered on the timeline can be classified based on the sentiment from the classification of hate speech sentiment can provide information about whether the sentence is hate speech or not.

To overcome these problems, the solution that can be given is to use sentiment classification, which is the process of understanding, extracting, and processing text data automatically to obtain sentiment information contained in opinion statements. In this research, sentiment analysis is carried out to see whether a person's opinion is included in hate speech or not [1]. To perform the analysis stage of a classification system, a data preprocessing step is required [5]. Data preprocessing aims to convert data into an easier and more effective format for users. Some of the data preprocessing methods are cleaning data that has missing values, adjusting the amount of data, and separating data into two groups, namely train and test [6]. The methods used for this sentiment analysis problem are AdaBoost and XGBoost. Adaboost (Adaptive Boosting) is one of the classification algorithms discovered by Yoav Freund and Robert Schapire [7]. It builds a robust classifier by combining several weak classifiers. This algorithm is adaptive because it can adapt to data and other classifier algorithms [8]. The data owned by researchers cannot always be processed directly. There is a possibility that the data has problems, such as an imbalance of class data. Datasets are classified as imbalanced class data if the data between classes are not balanced [9]. This can cause the prediction results to be accurate only for the class with the most responses. Researchers use the SMOTE algorithm to generate synthetic data from minor classes to overcome this problem. In comparison, XGBoost (Extreme Gradient Boosting) is one of the machine learning techniques to overcome regression and classification problems based on the Gradient Boosting Decision Tree (GBDT) [10].

Previous researchers have conducted many studies on text classification. Abusive language categorized based on Twitter social media tweets using the Indonesian language dictionary has also been done. Research by...
Hidayatullah et al. classification of coarse language with Indonesian language dictionary tweets using NBC and SVM algorithms. The performance of the two algorithms obtained in the results of the SVM algorithm is higher than the NBC algorithm. The accuracy results obtained by researchers are 98% for NBC and 99% for SVM [11]. Kristiawan also conducted research using the Indonesian dictionary. Using cyberbullying datasets on Twitter by researchers Luqyana et al. And classifying cyberbullying words with ANN and Adaboost methods [12]. The comparison results of the two methods get 99.8% accuracy with ANN and 99.5% with Adaboost [13].

Abusive language detection research was also conducted with Thai language datasets by researchers Tuarob et al. Using Facebook datasets with several machine learning-based classifiers such as kNN, DMNB, SVM, and RFDT. Researchers use n-gram word features (unigram words and bigram words) and TF-IDF as feature extraction. The results obtained an F1-Score value of 86.01% using Discriminative Multinomial Naive Bayes (DMNB) as a classifier. This shows that DMNB is better than KNN, SVM, and RFDT [3]. Suwarno et al. Researchers also researched rough language classification using machine learning algorithms and analyzed the accuracy, precision, recall, and F1-score of 3 algorithms, namely SVM, XGBoost, and ANN. Using the Indonesian Twitter dataset. The analysis results that researchers have done get SVM results higher than XGBoost and ANN. The results are 83.2% with SVM, 76.6% with XGBoost, and 82.9 with ANN [14].

Research by Sahrul, Rahman et al. Implemented Artificial Neural networks (ANN) and Recurrent Neural networks (RNN) as a comparison for offensive coarse language detection. From the research results, the RNN model gets better results with 83.8% accuracy (label 1) and 84.4% (label 2), while the ANN model with 82.2% accuracy (label 1) and 80.6% (label 2). Researchers also classify Twitter datasets into two labels: offensive (label 1) and not offensive (label 2). Researchers perform the synthetic minority oversampling technique (SMOTE) because the amount of data between labels one and two is not balanced [15].

Social media applications, especially Twitter, are a personal means to express opinions, their hearts, and so on. Twitter is also used by various organizations, agencies, and individuals. The internet celebrity who is the subject of this research is Anya Geraldine. On social media, Twitter Anya has an account with the name (@anyaselalubenar) and has more than 4.3 million followers with more than 2,100 tweets. The selection of Anya Geraldine as the subject of this research is due to the many pros and cons of Twitter users in Indonesia to Anya Geraldine. Therefore, the author is interested in researching sentiment analysis of hate speech types on public figures based on tweet data using the AdaBoost and XGBoost methods. The output of this research is to analyze the performance results of the classification that has been carried out by the system using the AdaBoost and XGBoost methods.

2. RESEARCH METHODOLOGY

2.1. Research Design

The system built is a system that can detect the use of harsh sentences in Indonesian text. Figure 1 is an overview of the flow of the system created in this study.

![Figure 1. Research Process Flow](image)

Based on Figure 1, the system creation flow will be explained.

2.2. Dataset

This study uses tweet data from Twitter using the Twitter API and only tweets in Indonesian. The data taken results from a Twitter search with the keyword "@anyaselalubenar" in January - March 2022. The initial data obtained was 20,995 tweets containing created-at, from-user, to-user, text, and id columns. After text preprocessing, the dataset became 7,475 tweets. The data consists of 2 labels with the following numbers.

<table>
<thead>
<tr>
<th>Label</th>
<th>Caption</th>
<th>Total Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Positive</td>
<td>6887</td>
</tr>
<tr>
<td>1</td>
<td>Negative</td>
<td>588</td>
</tr>
</tbody>
</table>

Daffa Ulayya Suhendra, Copyright © 2022, MIB, Page 1485
Submitted: 29/06/2022; Accepted: 11/07/2022; Published: 25/07/2022
2.3. Cleaning and Preprocessing Data

Text preprocessing is an important step in classification. Text preprocessing is meant to remove noise, standardize word formats, and reduce word count [16]. The following preprocessing stage consists of cleaning, case folding, tokenizing, and filtering processes.

a. Cleaning is the process of removing unnecessary words to reduce noise from the classification process. An example of cleaning results can be seen in Table 2.

Table 2. Cleaning

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>@anyaselalubenar Sama saja jadi orang buruk jg</td>
<td>Sama saja jadi orang buruk jg cobaan nya lebih</td>
</tr>
<tr>
<td>cobaan nya lebih banyak ?? paling enak yg jadi</td>
<td>banyak paling enak yg jadi orang normal</td>
</tr>
<tr>
<td>orang normal campuran baik dan buruk ??????</td>
<td>campuran baik dan buruk</td>
</tr>
</tbody>
</table>

b. Case folding is converting all letters in the document to lowercase, only letters ‘a’ to ‘z’ are accepted. An example of Case Folding results can be seen in Table 3.

Table 3. Case Folding

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>COBAAN MAH EMNG BANYAK, KALO</td>
<td>cobaan mah emng banyak, kalo dikit mah cobain</td>
</tr>
<tr>
<td>DIKIT MAH COBAIN</td>
<td></td>
</tr>
</tbody>
</table>

c. Tokenizing or parsing is the truncation of the input string based on each word that comprises the input string. An example of Tokenizing results can be seen in Table 4.

Table 4. Tokenizing

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mungkin kamu ngerasa km udah jd orang baik</td>
<td>[mungkin, kamu, ngerasa, km, udah, jd, orang,</td>
</tr>
<tr>
<td>menurut dirimu sendiri tapi dimata orang lain</td>
<td>baik, menurut, dirimu, sendiri, tapi, dimata,</td>
</tr>
<tr>
<td>enggak</td>
<td>orang, lain, enggak]</td>
</tr>
</tbody>
</table>

d. Filtering is the stage where important words are extracted from the previous step’s results. The process can use a stoplist algorithm (removing less important words) or a wordlist (keeping important words). Examples of stopwords in the bag-of-word approach are “which”, “and”, “in”, “of”. An example of Filtering results can be seen in Table 5.

Table 5. Filtering

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>bener ga sih kalo orang dibaikin terus jadi</td>
<td>bener ga sih kalo orang dibaikin ngelunjak</td>
</tr>
<tr>
<td>ngelunjak, sekalinya kita jahat malah orangnya jd</td>
<td>sekali jahat orang jd bgt kampret musti</td>
</tr>
<tr>
<td>baik bgt. kampret musti…</td>
<td></td>
</tr>
</tbody>
</table>

e. The following are example sentences in the dataset based on labels that have gone through the text preprocessing stage. An example of text preprocessing results can be seen in Table 6.

Table 6. Example Sentences in the Dataset

<table>
<thead>
<tr>
<th>Label</th>
<th>Tweet</th>
<th>Caption</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>kak cantik banget</td>
<td>Positif</td>
</tr>
<tr>
<td>1</td>
<td>jangan kak acting kakak belom bagus kalo udah bagus baru deh pensiun</td>
<td>Negative</td>
</tr>
</tbody>
</table>

2.4. Word Breaker

The word weighting process used is Term Frequency-Inverse Document Frequency (TF-IDF) which is a simple feature selection method with low computation and focuses on the occurrence of terms (words) throughout the document. The following weight calculation used is [16].

\[ n_t(d) = d_f_i \times \log \frac{D}{df_i} \]  

(1)

Caption:

\[ d_f_i \] = The number of documents in the dataset contains the searched feature i (word).

\[ D \] = Total number of documents.
2.5. Implementation

2.5.1. AdaBoost

Adaptive Boosting is one of several variations of boosting algorithms [17]. AdaBoost includes ensemble learning which is commonly used in boosting algorithms. Boosting can be combined with other classifier algorithms to improve classification performance intuitively. Combining multiple models can help when the models are different from each other [17]. The steps of the AdaBoost algorithm are as follows.

a. Input: Data with labels \((x_i, y_i), \ldots, (x_N, y_N)\), to component learn algorithm turnover amount \(T\).

b. Initialize: weight of a training sample \(w_i^1 = 1/N\), for all \(i = 1, \ldots, N\)

c. Do for \(t = 1, \ldots, T\)

d. Use component learn to train a classification component \(h_t\), on sample training weights.

e. Compute the training error at

\[
h_t : \epsilon_t = \sum_{i=1}^{N} w_i^t y_i \neq h_t(x_i)\]

f. Assign weight to component classifier

\[
h_t = \alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)\]

g. Update training sample weights

\[
w_i^{t+1} = \frac{w_i^t \exp\left(-\alpha_t y_i h_t(x_i)\right)}{C_t}, i = 1, \ldots, N\]

\(C_t = \) a normalization constant.

h. Output:

\[
f(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)\]

2.5.2. XGBoost

XGBoost (Extreme Gradient Boosting) is an implementation of a gradient Decision Tree designed for speed and performance [18]. In use, XGBoost is used for supervised learning problems, which use training data where \(x_i\) predict target variable \(y_i\). The steps of the XGBoost algorithm in calculating the predicted value of step \((t)\) at \(\hat{y}_i^{(t)}\) is as follows [19],

\[
\hat{y}_i^{(t)} = \sum_{k=1}^{T} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)\]

XGBoost has objective functions for training loss and regularization. The objective function is

\[
\text{obj}(\theta) = L(\theta) + \Omega(\theta)\]

Where \(L\) is the training loss function and \(\Omega\) is the regularization. Training loss measures how predictive the model is in the training data. A frequently used choice is the mean squared error which is

\[
L(\theta) = \sum_{i}(y_i - \hat{y}_i)^2\]

Furthermore, to define regulatory complexity is as follows,

\[
\Omega(f) = y^T + \frac{1}{2} \lambda \sum^T w_j^2\]

2.6. Smote

Synthetic Minority Oversampling Technique (SMOTE) is an oversampling technique reproduces minority class data using synthetic data derived from minority class data replication. Oversampling in SMOTE takes instances of the minority class, finds the KNN of each instance, then creates synthetic instances instead of replicating instances of the minority class. Therefore, it can avoid the problem of excessive overfitting [9].

The algorithm that works with SMOTE takes the difference between the vector of minority class features and the nearest neighbor value and multiplies the value by a random number from 0 and 1. Then the result of the calculation is added to the feature vector so that the result of the vector value is obtained [20].

\[
x_{\text{new}} = x_i + (\hat{x}_i - x_i) \times \delta\]

2.7. Performance Analysis

Confusion Matrix is a measurement tool that can be used to calculate the performance or correctness of the classification process. Using Confusion Matrix analyzes how well a classifier can recognize records from different classes. The following is the Confusion Matrix Table used in this study [21].
Table 7. Confusion Matrix

<table>
<thead>
<tr>
<th>Actual label</th>
<th>TP</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>

Positive Negative Prediction label

a. Accuracy is a comparison of correct predictions compared to the overall positive predicted outcomes.

\[ \text{Accuracy} = \frac{TP + FN}{(TP + FP + FN + TN)} \] (11)

b. Precision is a comparison of correctly classified data and all correctly classified data.

\[ \text{Precision} = \frac{TP}{(TP + FP)} \] (12)

c. Recall is the ratio of the number of correctly classified data to the number of data in that class. The calculation formula is as follows.

\[ \text{Recall} = \frac{TP}{(TP + FN)} \] (13)

d. F1-score is the average ratio of precision value and recall value.

\[ F1 - \text{Score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \] (14)

3. RESULTS AND DISCUSSIONS

3.1. Test Results

The dataset used was 7,475 tweets divided into two labels: Positive 6,887 tweets and Negative 588 tweets. Three experts do the labeling manually, and the majority determine a label. In this study, the dataset that has been labeled will be tested. The data will be split with 70% training data weight of 5,232 rows and 30% test data of 2,243 rows by testing each method, namely AdaBoost and XGBoost. Of course, the data has been preprocessed first.

![Figure 2. Number of Sentences for Each Label Before Oversampling](image2)

Figure 2 shows imbalanced data. When imbalanced data, the classification tends to ignore the minority class. So that many test data that should be in the minority class are mispredicted by the classification model [9].

![Figure 3. Number of Sentences for Each Label After Oversampling](image3)
Figure 3 shows the dataset used in this study is balanced (balanced dataset) by performing oversampling techniques (SMOTE). The accuracy results obtained from a balanced dataset will be better than those using an unbalanced dataset [9].

Figure 4. Number of Sentences for Each Label After Undersampling

For comparison in Figure 4, testing is done with Random Under Sampling, which balances the majority class dataset with the minority class so that the accuracy results obtained by the model are not as good as using the oversampling technique (SMOTE).

After balancing the data, parameter tuning with the best parameters and simultaneous tuning with grid search. The following are the parameters that were tried.

Table 8. Model Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>n_estimators</td>
<td>50, 100, 200</td>
</tr>
<tr>
<td>learning rate</td>
<td>0.005, 0.01, 0.05, 0.1, 0.3</td>
</tr>
</tbody>
</table>

Table 8 show parameter tuning uses 5-fold cross validation and the best parameters are n_estimators = 50 and learning_rate = 0.005.

3.2. Analysis of Test Results

In this section, we compare the performance of two algorithms, AdaBoost and XGBoost. After the data is vectorized using TF-IDF, researchers conducted several trials on each machine learning model. The model performance can be seen in Table 9, Table 10, Table 11, and Table 12. The model performance results from the performance that has been tested using the predetermined model.

Table 9. Model Performance before Oversampling

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>0.9358</td>
<td>0.7082</td>
<td>0.532</td>
<td>0.5443</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.9349</td>
<td>0.6693</td>
<td>0.5188</td>
<td>0.5211</td>
</tr>
</tbody>
</table>

Table 9 shows the model results from AdaBoost and XGBoost before oversampling. The results obtained for Accuracy, Precision, Recall, and F1-Score are 0.9358, 0.7082, 0.532, 0.5443 with Adaboost and 0.9349, 0.6693, 0.5188, 0.5211 with XGBoost.

Table 10. Model Performance after Oversampling

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>0.8269</td>
<td>0.8342</td>
<td>0.8274</td>
<td>0.8261</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.9314</td>
<td>0.9322</td>
<td>0.9315</td>
<td>0.9314</td>
</tr>
</tbody>
</table>

Table 10 shows the model results from AdaBoost and XGBoost before oversampling. The results obtained for Accuracy, Precision, Recall, and F1-Score are 0.8269, 0.8342, 0.8274, 0.8261 with Adaboost and 0.9314, 0.9322, 0.9315, 0.9314 with XGBoost.

Table 11. Model Performance after Undersampling

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>0.5583</td>
<td>0.5708</td>
<td>0.5527</td>
<td>0.5253</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.6222</td>
<td>0.6289</td>
<td>0.6244</td>
<td>0.6196</td>
</tr>
</tbody>
</table>

Table 11 shows the model results from AdaBoost and XGBoost after undersampling. The results obtained for Accuracy, Precision, Recall, and F1-Score are 0.5583, 0.5708, 0.5527, 0.5253 with Adaboost and 0.6222, 0.6289, 0.6244, 0.6196 with XGBoost.
Table 12. Model Performance with Hyperparameter Tune

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>0.87</td>
<td>0.88</td>
<td>0.87</td>
<td>0.87</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 12 shows the model results from AdaBoost and XGBoost with hyperparameter tune. The results obtained for Accuracy, Precision, Recall, and F1-Score are 0.87, 0.88, 0.87, 0.87 with Adaboost and 0.95, 0.95, 0.95, 0.95 with XGBoost.

From Table 9 and Table 10, it can be seen that there is a considerable increase after going through the OverSampling stage. This proves that balanced data will affect better results for the model used. Conversely, if you use the Under Sampling technique as in Table 11, the results of the model decrease because the source of the dataset label is reduced. So it requires the Over Sampling technique to increase the performance of the model. The test results on SMOTE AdaBoost and SMOTE XGBoost models have a higher value than SMOTE AdaBoost. Table 12 shows that XGBoost has the best Accuracy, Precision, Recall, and F1-Score results. From this study, it can be concluded that the XGBoost algorithm performs classification tasks for Indonesian sentiment datasets better.

3.3. Model Prediction

Confusion Matrix in Table 13 resulting from predictions for AdaBoost and XGBoost models against 7,475 test or validation data. 6,887 positive tweets were correctly classified as 2,000 and ranked negative 74. A total of 588 negative tweets were correctly classified as 130 and classified as positive 1,900. So it can be seen that the accuracy of the classification of positive sentiment is more significant in the labeling result data.

Table 13. Confusion Matrix Result of Model

<table>
<thead>
<tr>
<th>Actual label</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2,000</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>130</td>
<td>1,900</td>
</tr>
</tbody>
</table>

4. CONCLUSION

In this research, we develop the performance of AdaBoost and XGBoost to perform sentiment analysis of Hate Speech words on public figure Anya Geraldine. Four scenarios were tested to see the interpretation of these methods in conducting sentiment analysis with Indonesian Twitter datasets. Based on the results, the best method for research in all scenarios is the same, namely XGBoost accuracy. XGBoost performed well in all scenarios with 93.4% in the design before oversampling, 93.1% in the design after oversampling, 62.2% in the design after undersampling, and 95% in the scenario with hyperparameter tune. However, the accuracy performance of Adaboost for sentiment analysis in all plans is relatively similar to the accuracy performance of XGBoost, with 93.5% in the scenario before oversampling, 82.6% in the design after oversampling, 55.8% in the design after undersampling and 87% in the design with hyperparameter tune. Although XGBoost performs better than Adaboost, it takes more time than Adaboost to train the model. From the results of this study, it is found that the algorithm of XGBoost is higher than the Adaboost algorithm, XGBoost, which uses the best hyperparameter tune method in performing sentiment analysis. The accuracy, precision, recall, and f1-score rate reach 95%.

REFERENCES


