iee

by USER DOSEN

Multilabel Classification of Student Feedback Data Using BERT and Machine Learning Methods

2023 14th International Conference on Information & Communication Technology and System (ICTS) | 979-8-3503-1216-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICTS58770.2023.10330849

Hamzah Setiawan
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
7025211025@mhs.its.ac.id

Chastine Fatichah*
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
chastine@if.its.ac.id

Ahmad Saikhu
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
saikhu@if.its.ac.id

***Abstract*— Studying student feedback is essential for educational institutions to provide good services to their students. The main purpose of evaluation is to improve the services offered to students through the interest monitoring information system. The guardian of each learner should provide feedback on infrastructure and learning services so that the educational institution can improve its services. In addition, the purpose of evaluation is to investigate student inquiries and receive responses from the appropriate departments to ensure the successful delivery of student services. Automatic classification of student feedback is necessary to improve response time and service quality. Student feedback should immediately follow the service to the relevant department, therefore the automated system classifies the feedback according to the unit handling it, prioritizing the most rapid development of the system. Each student feedback can be handled by more than two units, so the problem includes multilabel classification.**

This study aims at multi-label classification of student feedback data. This study uses a Bidirectional Encoder Representation from Transformers (BERT) to derive word vectors from student feedback data. In this study, several machine learning methods such as Support Vector Machines(SVM), K-Nearest Neighbors(KNN), Random Forests(RF), and Decision Trees(DT) are used to classify multi-label student feedback and compare their performances.

This dataset consists of an assessment of the guardianship information system for 3323 students with the composition of the experiment using a comparison of 80% training data and 20% testing data. The SVM method with linear kernel has the best performance as evidenced by the accuracy of 82% and F1 value of 90%.

Keywords—Pre-trained Word Embedding, BERT, Machine Learning, Multilabel classification, Student feedback.

I. Introduction

Higher education has an important role in human resource development. Society demands more advanced and satisfying services from universities, especially with the development of science and technology. In addition, students also expect pleasant administrative services so that they can study well and get satisfaction. Services provided by universities must be of high quality to gain public trust. To achieve this, universities must have service criteria and standards or fulfill customer service measures to improve the quality of education [1].

Customer complaints are feedback from user about the quality of the service. The more complaints, the more important it is for the company to pay extra attention and improve the product or service. Student complaints against educational services are also important because they can affect student turnover. Important factors in handling student complaints are physical appearance, credibility, responsiveness, assurance, and empathy. The trust and commitment of students as users of educational services is very important to maintain the image of the university. Educational services are an integral part of creating a good academic environment. Providing excellent service to learners is an important factor in the progress of their studies [2].

A previous study compared single machine learning, ensemble learning, and deep learning approaches for complaint classification using hybrid features. The evaluation results show that the proposed ensemble classifiers and deep learning models provide better accuracy compared to single classifiers. The Random Forest algorithm performs best among the classifiers. Using hybrid features for classification also improves the performance compared to traditional frequency-based features [3]. The problem domain of Hotel reviews domain is sentiment analysis and opinion mining in online reviews and social media data. This research focuses on identifying extreme opinions in terms of praise and complaints from customer reviews to understand their true opinion about a product or service.

The previous research on customer complaints using deep learning techniques on the official COMS CRIS (Centre for Railway Information Systems) application dataset [4]. This research proposed RailNeural that used a Bidirectional long-short term memory (Bi-LSTM) to analyze the user complaints, capture the underlying character-level features, and classify them into appropriate field units to ensure fast and accurate complaint response. This research outperformed several baseline models, achieving 93.25% accuracy and 93% F1 [4].

Previous research focused on analyzing customer complaints, most of which used multiclass data. In this context, multi-class data refers to a data set that contains more than two different categories or classes related to customer complaints. This study uses Student Learning Feedback Service to categorize by relevant units. Each claim classified into two or more units should be included in the multi-label classification problem.

To address such issues, we propose a multi-label classification of student feedback using Bidirectional Encoder Representations of Transformers (BERT) and machine learning methods. This research used BERT, a pre-training language model, to extract significant features from student feedback text. The extracted features are then used as input for various machine learning models, such as KNN, SVM and (RF), to perform multi-label classification of the feedback data. The purpose of this research is to develop an accurate system to assist educational institutions in providing a better response to student complaints and needs by providing responses that can be followed up by the relevant units that get responses regarding student problems.

II. Methods and Materials

This system diagram describes the use of BERT and machine learning methods for multi-label classification of student feedback data. It includes data collection, preprocessing, BERT coding, feature extraction, model training, evaluation, deployment, and iterative improvement to build a robust and accurate system for analyzing and understanding student feedback.

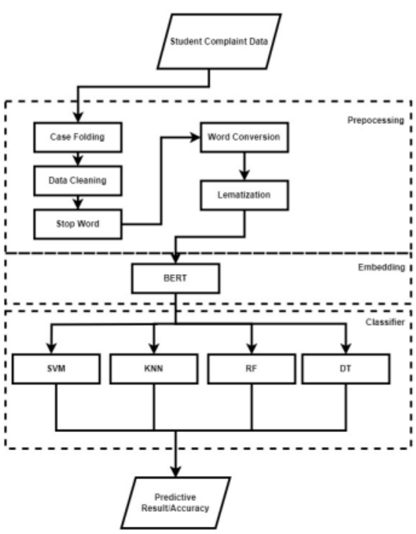

Fig 1. Process Diagram

A. Datasets collection

In this study, we use the indexing method to retrieve data from the database of the student information system. The data covers the period from February 11, 2021 to December 31, 2021 and includes student feedback collected during tutoring meetings with their tutors. Student feedback is used to evaluate the services offered by University work units and to provide feedback and advice on what students experience across all University work units.

The dataset used in this study contains 3,323 records filtered from a larger dataset that originally contained 9,380 records. The material includes reactions from various university work units, such as the University Secretariat (SU), Property and Environment Maintenance (DPAL), Department of Information Systems and Technology (DSTI), Department of Academic Affairs (DA), PMB, Student Affairs and Alumni Department (DPMBKA), Finance Section (DK) and Library.

Each work item is appropriately labeled to facilitate analysis of the dataset. This is described in Table 1 in Dataset Labeling Characteristics. The dataset was processed using crawling techniques, and the resulting dataset was used to run SVM, KNN, RD and (DT)[5].

The aim of this investigation is to analyze the feedback provided by students and elicit a response from relevant stakeholders to ensure the smooth operation of services provided to students. By leveraging the power of machine learning algorithms and preprocessing techniques, this study aims to provide insights into the challenges faced by students and identify areas where improvements can be made to enhance the quality of services provided by the university[6].

Once the data is cleaned, pre-processing is done to make it more manageable and consistent. One of the data processing steps is case sensitivity, converting all letters to lowercase using Python's "lower" function. Punctuation marks, symbols and numbers were also removed from the data to eliminate sources of noise that could interfere with the classification process. The data is then tokenized. This means that the text is divided into individual words or tokens. Next, stop words, commonly used words that do not provide meaningful information for the classification process, are removed from the data. Finally, we performed stemming to convert each word into a stem form to make the data analysis process easier. All these steps were performed using the Python programming language and existing library resources. The resulting data is stored in CSV format and can be used in the classification process. The labels assigned to the records are listed in Table 1, using passive voice to describe the characteristics of each label.[7].

B. Preprocessing

Preprocessing is an important step in natural language processing such as text classification and analysis. Transform raw text data into a format suitable for further analysis and modeling. Various pretreatment techniques are described.[8]:

1. Case Folding:

Case folding, also known as text normalization, involves converting all text to lowercase or uppercase to ensure consistency and remove any potential discrepancies caused by different letter cases. This step helps to treat words with the same spelling but different cases as identical. For example, "Hello" and "hello" would be transformed to "hello" during case folding.

2. Data Cleaning:

Data cleaning involves removing irrelevant or noisy information from the text. This can include removing special characters, punctuation marks, URLs, and any other symbols that do not contribute to the overall meaning of the text. Data cleaning helps reduce the dimensionality of the data and focus on the essential content.

3. Stop Words:

Stop words are common words that occur frequently in a language and often do not carry significant meaning. Examples of stop words include "a," "an," "the," "and," "is," "are," etc. In many text analysis tasks, stop words are removed from the text data to reduce noise and improve computational efficiency. Removing stop words can be beneficial in cases where the frequency of occurrence is not essential, such as sentiment analysis or topic modeling.

4. Word Conversion:

Word conversion techniques involve transforming words into a standard or canonical form. This step helps to treat different variations of the same word as identical. Common word conversion techniques include stemming and lemmatization. Stemming: Stemming reduces words to their root or base form by removing suffixes or prefixes. It aims to remove inflections and variations from words while preserving their core meaning. For example, stemming the words "running," "runs," and "run" would result in the common stem "run."[9].

5. Lemmatization

Lemmatization: Lemmatization, similar to stemming, reduces words to their base form, but it takes into account the context and applies morphological analysis. It converts words to their lemma, which is the dictionary or canonical form of a word. For example, lemmatizing the words "running," "runs," and "run" would result in the lemma "run." Lemmatization typically produces more accurate results compared to stemming, as it considers the part of speech of the word and ensures that the resulting lemma is a valid word in the language. However, lemmatization is computationally more expensive than stemming.

Preprocessing techniques, including case folding, data cleaning, stop word removal, and word conversion (stemming or lemmatization), help to standardize and clean the text data, making it more suitable for subsequent analysis and modeling tasks. These techniques contribute to reducing noise, improving computational efficiency, and ensuring that the focus is on the essential content and semantic meaning of the text[7].

C. *Word Embedding*

Word embedding is a powerful technique used in natural language processing (NLP) and machine learning to convert words into numeric vectors that can be easily processed and analyzed. It involves representing words as high-dimensional vectors in a continuous vector space, where words with similar meanings are located close to each other. This technique captures the context and meaning of words in a way that traditional bag-of-words methods cannot. Word embeddings have been pre-trained on large corpora of text, such as Wikipedia and news articles, and are often used as a starting point for further fine-tuning on specific tasks, such as sentiment analysis or text classification. Bidirectional Encoder Representations from Transformers (BERT) is one such pre-trained language model that has gained popularity in recent years for its state-of-the-art performance on a range of NLP tasks[10].

The BERT model is a pre-trained contextual word representation model that makes use of two-way transformers. It is based on the Masked Language Models (MLMs). The BERT model employs a multi-layer, bidirectional transformer encoder-decoder system for its architectural underpinnings. The architecture described here is adhered to by transformers, which operate utilizing stacked self-attention and point-wise and are fully coupled to encoders and decoders. The performance of the BERT framework consists of pre-training and fine-tuning, both of which are distinct phases. Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) are utilized for the pre-training data in BERT rather than the more conventional left-to-right or right-to-left method [8]. MLM is used to fill in the gaps, where the model makes use of the words around the token mask to make a prediction about the proper word, whereas NSP uses the two models given to make a prediction about the subsequent sentence. BERT will then perform fine-tuning on the data after it has completed pre-training, with the fine-tuning process being initialized with the pre-trained parameters. Every parameter gets fine-tuned with the use of labeled data from subsequent processes[10].

D. *Classifier*

Machine learning is a branch of artificial intelligence (AI) that involves developing algorithms and statistical models that allow computers to learn from experience and improve themselves without being explicitly programmed. Machine learning models such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Random Forests (RF), and Decision Trees (DT) are commonly used in classification tasks as discussed in this context[5].

Support Vector Machine (SVM) is a popular machine learning algorithm that belongs to the family of supervised learning algorithms. SVMs are used for classification and regression analysis and are known for their high accuracy and ability to handle large datasets. The basic principle behind SVM is to find the best hyperplane that separates the data into different classes with the maximum margin. The hyperplane is calculated based on the support vectors, which are the data points that are closest to the decision boundary[5].

The Gaussian kernel is a commonly used kernel function in SVM that transforms the data into a higher-dimensional space in order to better separate the classes. The penalty parameter C in SVM controls the trade-off between maximizing the margin and minimizing the classification error. The choice of C value affects the bias-variance trade-off, with smaller C values leading to a wider margin and potentially more errors and larger C values leading to a narrower margin and potentially overfitting.

The random forest method is often used in situations where there is a large dataset and a complex decision boundary. It works by creating multiple decision trees and then combining their predictions to form a final decision.

Each tree is trained on a subset of the data and uses a random subset of the features to make decisions at each node. The final decision is made by aggregating the decisions of all the trees in the forest. Random Forest is also able to identify the most important features in the data, which can be useful for feature selection. Overall, Random Forest is a powerful tool for classification and regression tasks and has been successfully applied in various domains such as healthcare, finance, and social media analysis[11].

Decision trees are often used in machine learning as a way to classify data by splitting it into smaller and smaller subsets based on specific attributes. This process of dividing the data into smaller groups continues until a specific criterion is met, such as all members of a subset having the same classification. At each step in the process, the algorithm decides which attribute to split on based on a measure of impurity, such as entropy or the Gini index. The resulting tree is made up of decision nodes, which represent the attribute used to split the data, and leaf nodes, which represent the final classification of the data. Decision trees are easy to interpret and can handle both numerical and categorical data, but they can be prone to overfitting, especially with noisy data or datasets with a large number of attributes. To overcome this issue, ensemble methods such as Random Forest can be used, as mentioned earlier[12].

E. Experimental Scenarios

This research will undertake model training utilizing one pre-trained word embedding and four different classifiers. We use BERT as word embedding in the classifier models used, notably SVM, KNN, RF, and DT. The data used comprises feedback data, which is given a suggestion label and marked for each unit. Next, the data goes through conventional preprocessing stages: case folding, tokenization, stop word, word conversion, and lemmatization. Finally, the preprocessing results are utilized to build word vectors, which we then train machine learning algorithms on. In each experiment, 80 percent of the dataset was utilized as training data and 20 percent as test data. This data sharing is meant to use more data for the training procedure[13].

F. Evaluation

In order to evaluate the model in this study, several metrics will be used, including the F1 score, precision, recall, and accuracy. The true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values will be calculated using equations 3 through 6:

$$Precision = \frac{TP}{(TP+FP)} \quad (3)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (4)$$

$$F1\ Score = \frac{(2\ x\ Precision\ x\ Recall)}{(Precision+Recall)} \quad (5)$$

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (6)$$

III. Results and Discussion

The model built in this study uses machine learning testing. Based on the results of the first scenario, classification with SVM, KNN, RF, and DT without pre-training word embedding results in better performance, as shown in Table II.

Based on the experimental SVM method using the kernel, degree, and C parameters, the best accuracy was obtained on the linear kernel with an accuracy of 80%, a precision of 90%, a recall of 86%, and an f1-score of 88%. This indicates that the linear kernel is the most suitable for this dataset and can effectively classify the student feedback. Additionally, the precision and recall values are also high, indicating that the model is able to accurately identify both positive and negative feedback, which are described in Table II.

TABLE II. Classification Accuracies of Machine Learning Algorithms Support Vector Machine (SVM)

Kernels	Degree	C	Precision	Recall	F1 Score	Accuracy
Poly	2	2	**93%**	78%	85%	78%
Poly	3	2	93%	64%	75%	68%
linear	-	2	90%	**86%**	**88%**	**80%**

Based on the experimental KNN method using the "n_neighbors" parameter, the best accuracy was obtained with 3 neighbors, achieving an accuracy of 63%, a precision of 86%, a recall of 61%, and an f1-score of 71%, which is described in Table III.

TABLE III. Classification Accuracies of Machine Learning Algorithms K-Nearest Neighbor (K-NN),

n_neighbors	Precision	Recall	F1 Score	Accuracy
1	86%	61%	71%	**63%**
2	92%	46%	62%	53%
3	91%	53%	67%	58%
4	94%	44%	60%	52%
5	92%	47%	62%	55%

Based on the Random Forest method experiment, the best accuracy was obtained using the n_estimators and random_state parameters. The optimal values were n_estimators = 30 and random_state = 40 or 60, resulting in an accuracy of 69%, precision of 93%, recall of 80%, and f1-score of 86%. Random forest is a popular ensemble learning method that utilizes multiple decision trees to generate predictions. The use of multiple decision trees reduces the risk of overfitting and improves the accuracy of the model. In this study, the Random Forest algorithm was able to achieve good results for the classification of student feedback, demonstrating the effectiveness of this approach in natural language processing tasks, which are described in Table V.

TABLE IV. CLASSIFICATION ACCURACIES OF MACHINE LEARNING ALGORITHMS RANDOM FOREST (RF),

n_estimators	Precision	Recalls	F1 Score	Accuracy
10	93%	79%	85%	78%
10	92%	78%	85%	77%
20	93%	79%	85%	78%
30	93%	79%	86%	78%
30	**93%**	**80%**	**86%**	**79%**

Based on the experiment with the Decision Tree method, the best accuracy results are obtained with the criterion parameter set to "gini" and the max_depth parameter set to 20, resulting in an accuracy of 74% and a precision of 85%.

The experimental results of the four methods produced processed data showing that the SVM method has the highest F1 score of 88% and accuracy of 80% compared to the other methods listed in Table VI.

The experimental results of the four methods produce processed data using 70% training data and 30% testing data which shows that the SVM method has the highest F1 value of 88% and accuracy of 80% compared to the other methods listed in Table VI.

TABLE V. RESULT DATA PROCESS USING 70% TRAINING AND 30% TESTING

No	Method	Precision	Recall	F1 Score	Accuracy
1	**SVM**	90%	**86%**	**88%**	**80%**
2	KNN	86%	61%	71%	63%
3	RF	**93%**	80%	86%	79%
4	DT	85%	81%	83%	74

TABLE VI. RESULT DATA PROCESS USING 80% TRAINING AND 20% TESTING

No	Method	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)
1	**SVM**	92%	**88%**	**90%**	**82%**
2	KNN	88%	63%	73%	65%
3	RF	**95%**	82%	88%	81%
4	DT	87%	83%	85%	76%

Other experimental results using the 80% training data composition and 20% testing data show that the SVM method has better value with the highest F1 value of 90% and accuracy of 82% compared to other methods listed in Table VII.

IV. CONCLUSIONS

In this study, we aim to classify student feedback as positive or negative by using four machine learning algorithms: SVM, k-NN, RF, and DT. We evaluated the performance of these algorithms on a dataset of 3323 feedbacks using metrics such as accuracy, precision, recall, and f1-score. The SVM method with a linear kernel had the best performance, evidenced by an accuracy of 82% and an F1 score of 90% using 80% training data and 20% testing data. We found that SVM with pre-trained BERT word embedding achieved the highest accuracy of 82%, outperforming other algorithms. Our results show that machine learning approaches can be effectively applied to analyze and classify student feedback, which can provide valuable insights to improve the quality of education. future work we will do to get the best results by using ensemble learning to improve accuracy.

REFERENCES

[1] S. Indriyani and S. Mardiana, “Pengaruh Penanganan Keluhan (Complaint Handling) Terhadap Kepercayaan Dan Komitmen Mahasiswa,” *J. Bus.*, vol. 2, no. 01, pp. 10–11, 2016, [Online]. Available: https://jurnal.darmajaya.ac.id/index.php/JurnalBisnis/article/view/615/406%0Ahttps://www.neliti.com/id/publications/75725/pengaruh-penanganan-keluhan-complaint-handling-terhadap-kepercayaan-dan-komitmen

[2] S. H. Lee, D. Levin, P. D. Finley, and C. M. Heilig, “Chief complaint classification with recurrent neural networks,” *J. Biomed. Inform.*, vol. 93, no. March, p. 103158, 2019, doi: 10.1016/j.jbi.2019.103158.

[3] S. Khedkar and S. Shinde, “Deep Learning and Ensemble Approach for Praise or Complaint Classification,” *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 449–458, 2020, doi: 10.1016/j.procs.2020.03.254.

[4] M. Gupta, A. Singh, R. Jain, A. Saxena, and S. Ahmed, “Multi-class railway complaints categorization using Neural Networks: RailNeural,” *J. Rail Transp. Plan. Manag.*, vol. 20, no. March, p. 100265, 2021, doi: 10.1016/j.jrtpm.2021.100265.

[5] D. Mou, Z. Wang, X. Tan, and S. Shi, “A variational inequality approach with SVM optimization algorithm for identifying mineral lithology,” *J. Appl. Geophys.*, vol. 204, no. December 2021, p. 104747, 2022, doi: 10.1016/j.jappgeo.2022.104747.

[6] A. Zingg, T. Singh, and S. Myneni, “Analysis of Online Peripartum Depression Communities: Application of Multilabel Text Classification Techniques to Inform Digitally-Mediated Prevention and Management,” *Front. Digit. Heal.*, vol. 3, no. May, 2021, doi: 10.3389/fdgth.2021.653769.

[7] Y. Guan and X. Li, “Multilabel Text Classification with Incomplete Labels: A Safe Generative Model with Label Manifold Regularization and Confidence Constraint,” *IEEE Multimed.*, vol. 27, no. 4, pp. 38–47, 2020, doi: 10.1109/MMUL.2020.3022068.

[8] T. H. J. Hidayat, Y. Ruldeviyani, A. R. Aditama, G. R. Madya, A. W. Nugraha, and M. W. Adisaputra, “Sentiment analysis of twitter data related to Rinca Island development using Doc2Vec and SVM and logistic regression as classifier,” *Procedia Comput. Sci.*, vol. 197, no. 2021, pp. 660–667, 2021, doi: 10.1016/j.procs.2021.12.187.

[9] W. K. Sari, D. P. Rini, R. F. Malik, and I. S. B. Azhar, “Klasifikasi Teks Multilabel pada Artikel Berita Menggunakan Long Short- Term Memory dengan Word2Vec,” *Resti*, vol. 1, no. 10, pp. 276–285, 2017.

[10] N. Quoc, K. Le, Q. Ho, V. Nguyen, and J. Chang,

"BERT-Promoter : An improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection," *Comput. Biol. Chem.*, vol. 99, no. July, p. 107732, 2022, doi: 10.1016/j.compbiolchem.2022.107732.

[11] L. Zhang and X. Chen, "Feature Selection Methods Based on Symmetric Uncertainty Coefficients and Independent Classification Information," *IEEE Access*, vol. 9, pp. 13845–13856, 2021, doi: 10.1109/ACCESS.2021.3049815.

[12] A. Almomany, W. R. Ayyad, and A. Jarrah, "Optimized implementation of an improved KNN classification algorithm using Intel FPGA platform: Covid-19 case study," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3815–3827, 2022, doi: 10.1016/j.jksuci.2022.04.006.

[13] M. Qiu, Y. Zhang, T. Ma, Q. Wu, and F. Jin, "Convolutional-neural-network-based multilabel text classification for automatic discrimination of legal documents," *Sensors Mater.*, vol. 32, no. 8p2, pp. 2659–2672, 2020, doi: 10.18494/SAM.2020.2794.

iee

ORIGINALITY REPORT

14%	10%	7%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Kaohsiung Medical University Student Paper	2%
2	Sujata Khedkar, Subhash Shinde. "Deep Learning and Ensemble Approach for Praise or Complaint Classification", Procedia Computer Science, 2020 Publication	1%
3	Submitted to Universitas Atma Jaya Yogyakarta Student Paper	1%
4	www.irjmets.com Internet Source	1%
5	Submitted to Coventry University Student Paper	1%
6	Mohd Fadly Hisham Ismail, Zazilah May, Vijanth Sagayan Asirvadam, Nazrul Anuar Nayan. "Machine-Learning-Based Classification for Pipeline Corrosion with Monte Carlo Probabilistic Analysis", Energies, 2023 Publication	1%

7	Submitted to rdia Student Paper	1%
8	Submitted to Kirkuk University Student Paper	1%
9	Submitted to Ngee Ann Polytechnic Student Paper	1%
10	fr.darktrace.com Internet Source	1%
11	journal.universitasbumigora.ac.id Internet Source	1%
12	hk.aconf.org Internet Source	1%
13	riunet.upv.es Internet Source	1%
14	ijritcc.org Internet Source	1%
15	Ji Hye Hwang, Gwang Ho Ahn, Soo Chang Chae, Ki Jin Kim. "A Design of High power 65nm CMOS 79GHz Power amplifier", 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), 2023 Publication	1%
16	Submitted to University of Greenwich Student Paper	1%

17	catalog.libraries.psu.edu Internet Source	1 %
18	9pdf.net Internet Source	1 %
19	acikbilim.yok.gov.tr Internet Source	1 %

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		