Journal of Integrated and Advanced Engineering (JIAE)
Vol. 1, No. 2, September 2021: 123-134
http://asasijournal.id/index.php/jiae
http://doi.org/10.51662/jiae.v1i2.24

Theme Identification using Machine Learning Techniques

Siti Hajar Jayady*, Hasmawati Antong
Department of Mechatronics Engineering, Faculty of Engineering, International Islamic University Malaysia, Malaysia

Abstract
With the abundance of online research platforms, much information presented in PDF files, such as articles and journals, can be obtained easily. In this case, students completing research projects would have many downloaded PDF articles on their laptops. However, identifying the target articles manually within the collection can be tiring as most articles consist of several pages that need to be analyzed. Reading each article to determine if the article relates theme and organizing the articles based on themes is time and energy-consuming. Referring to this problem, a PDF files organizer that implemented a theme identifier is necessary. Thus, work will focus on automatic text classification using the machine learning methods to build a theme identifier employed in the PDF files organizer to classify articles into augmented reality and machine learning. A total of 1000 text documents for both themes were used to build the classification model. Moreover, the pre-preprocessing step for data cleaning and TF-IDF feature extraction for text vectorization and to reduce sparse vectors were performed. 80% of the dataset were used for training, and the remaining were used to validate the trained models. The classification models proposed in this work are Linear SVM and Multinomial Naïve Bayes. The accuracy of the models was evaluated using a confusion matrix. For the Linear SVM model, grid-search optimization was performed to determine the optimal value of the Cost parameter.

Keywords:
Multinomial Naïve Bayes;
Portable Document File (PDF);
Pre-processing;
Term Frequency-Inverse Document Frequency (TF-IDF);

Article History:
Received: November 2, 2021
Revised: November 22, 2021
Accepted: November 25, 2021
Published: November 30, 2021

Corresponding Author:
Siti Hajar Jayady,
Electrical Department of Mechatronics Engineering, Faculty of Engineering, International Islamic University Malaysia, Malaysia
Email:
sitihajarjayady30@gmail.com

INTRODUCTION

With evolving technology, lots of information is available in electronic documents that can be retrieved from different resources in this modern era. This information exists in various forms, such as texts and pictures [1][2]. Along with the fast increment of such information, various studies have been made to process and analyze them. For instance, opinions or reviews from customers on certain products can be obtained from social media and certain websites. In this case, users tend to express and exchange opinions through these platforms. Therefore, the information will become a reference for the buyers before they decide on buying the product. Besides, many research articles are made available on the internet and become the main valuable resources for research in many areas, including innovation, education, development, and medicine. Thus, it becomes crucial to analyze and process the articles such that these articles are very useful and informative for everyone in society. However, due to many text documents, it is hard to retrieve, classify and organize these data manually because it is time and energy-consuming [3, 4, 5]. As a result, humans tend to get tired and easily distracted, making mistakes. All these limitations bring to the development of classifiers that can do classification tasks to make tasks well-organized and easier [6, 7, 8, 9].

The classification of Portable Document Format (PDF) articles is the key to processing the articles' information, which can help identify the theme of an article effectively and fast [10, 11, 12, 13]. Text classification is defined as a process of automatically categorizing a document of texts into its desired class, and it is a method used for organizing a large number of textual electronic documents [14].

PDF articles might consist of hundreds to thousands of words to read and identify. If the topic is related to the reader's target topic and manually organizing a large number of PDF articles into their respective themes to ease the process of article selection later is inconvenient as it takes longer time and energy-consuming. In addition, most of the current classifiers focus on sentiment analysis and spam detection. In contrast, the text is far shorter than text on PDF files, for instance, journals and articles. Thus, a classification algorithm suitable for longer text is still not being identified. Moreover, the PDF articles contain many words, including punctuation marks and special and numerical characters. In contrast, these characters do not contribute to the article's content and affect the machine learning model's performance.

Last but not least, the resulting vector from the text vectorization of a document that consists of an abundance of words may consist of lots of zero values as not all PDF documents contain the respective words. This vector is called the sparse vector. The increases of the sparse vector will increase space and time complexity.

This project aims to construct a theme identifier using a suitable machine learning model to identify the theme of an article, whether the article is related to augmented reality or machine learning. The theme identifier will be employed in the GUI that organize PDF articles based on themes.

MATERIAL AND METHOD

Text Classification

Text classification is defined as automatically categorizing a document of texts into its desired class. According to Coi et al., terms automatically are explained as the classifier's ability to decide to classify the texts based on its past observation, which can be accomplished by using machine learning [15]. Text classification has been used to categorize text documents into one or more categories based on the words in the document. As mentioned in [16], there are two types of classification: binary and multi-classification. Binary classification is classifying a text into one of two categories. For instance, it is implemented in sentiment analysis and spam detection. The multiclass classification is further divided into single label multiclass and multi-label classification.

Binary Classification

Qader et al. [17] proposed a sentiment analyzer for movie reviews to determine what kind of movies users prefer. The Linear Support Vector Machine (SVM) and Naïve Bayes machine learning algorithms were tested on four different movie genre datasets: action, adventure, drama and romance genre. The result shows that Linear SVM outperformed the Naïve Bayes in all four datasets. The algorithm also includes data pre-processing for words reduction. Another sentiment analysis had proposed by [17]. This work compared the performance of Linear SVM and Multinomial Naïve Bayes on the dataset of airline reviews obtained from Twitter. Several words removal steps, stemming, and data splitting were done on the dataset before classification. They used 67% of the dataset as the training set and the remaining dataset as the test set. From the result, the SVM obtained higher accuracy and precision than Naïve Bayes. A research study by Prastyo et al. [18] work on sentiment analysis on the dataset of Indonesia tweets regarding the omnibus law issue and compares Linear SVM performance to different kernel functions such as Polynomial, Radial Basis Function (RBF) and Sigmoid kernel. The Term Frequency-Inverse Document Frequency (TF-IDF) extraction and pre-processing steps were performed on the dataset. Based on the result, the classification model using Linear SVM works excellent on highly dimensional data as it achieved the highest accuracy on both datasets with 4000 features and dataset with all features with 96.53% and 96.50% of accuracy.

Moreover, Olatunji had proposed a spam detector using SVM with RBF kernel. It is mentioned that the kernel function works by transforming the non-linearly separable data into higher dimensional space such that the data will be linearly separable [14]. According to Olatunji, spam detection can be performed using a content-based filtering technique implemented with classification models such as Decision Tree and SVM. The filtering technique will create rules to identify spam and non-spam emails based on the occurrences of certain keywords [14]. In this case, when the rate of the keywords exceeds the threshold value, the email is considered spam email.

Multiclass Classification

Multiclass classification is a classification problem that involves more than two classes. Wang et al. [16] stated that Multiclass classification could be divided into single-label multiclass and multi-label classification. Single-label multiclass is used to classify a sample into only one class out of the classes, which can be accomplished using several binary classifiers. Differ from single-label multiclass classification, multi-label classification is used for classifying a sample into several classes simultaneously. Based on [19], single-label multiclass classification using the Naïve Bayes approach can be performed directly using the same method as a binary classification problem. The posterior probability of the sample belonging to each class will be calculated and compared. Thus, the predicted sample will be assigned to the class with the highest posterior probability among the other classes. On the other hand, single-label multiclass classification using SVM cannot be performed directly. However, this can be accomplished by breaking down the classification problem into a series of binary classification problems [19], which can be accomplished based on n One-vs-One (OVO) and One-vs-Rest (OVR) [19].

Machine Learning

Machine learning is a subset of Artificial Intelligence (AI) that allows a system to learn and improve from the observed data by training the machine learning model with a set of data. In-text classification, the labelled dataset will be fed into the machine learning model to recognize the pattern and correctly classify the unlabeled dataset. There are two main machine learning categories: supervised and unsupervised machine learning [19]. Supervised machine learning is suitable for classification and regression tasks. On the other hand, unsupervised machine learning is suitable for clustering and association problems. In supervised machine learning, the input dataset provided to the algorithm during training is already being labelled with its expected output. With an adequate dataset during the training process, the model will classify the unlabelled dataset [20]. However, the input dataset is not being labelled for an unsupervised machine learning model. Thus, the model will have to infer the patterns within the dataset and group the dataset based on the similarity of its content.

Feature Extraction

Feature extraction is a method used to transform the text into a numerical representation which is the language that machine learning can understand. Besides, it also can reduce the text dimension by only extracting the selected features. Olatunji [14] stated that reducing the text dimension and retaining only significant words can improve classifier accuracy in classifying text. Bag of Words (BOW) and TF-IDF is commonly used for feature extraction.

The vector form of a document represents the frequency of each word occurrence. As for the TF-IDF method, the vectorization takes into consideration both term frequency in that document and other documents. Thus, each word's TF- IDF score signifies the word important to the document. The lower value of the TF-IDF score means the words are not as crucial to the document as the word in most documents, such as the stop words. Meanwhile, Higher TF-IDF signifies that the word is important as it rarely occurs within the dataset and frequently occurs in the given document. However, vectorization using both methods does not preserve the sequence of the words in a document. Olatunji [14] had compared the performance of sentiment analyzer with Linear SVM and Multinomial Naïve Bayes that implemented TF-IDF and BOW. The accuracy of both models using TF-IDF is higher than the accuracy when using BOW.

Method

In this study, the text classification on augmented reality and machine learning related articles to build a theme identifier has been done using two machine learning methods: Linear SVM and Multinomial Naïve Bayes. Several stages were conducted to build the theme identifier, including data acquisition, data pre-processing. It needs splitting data into training and testing data for cross-validation purposes, feature extraction using TF-IDF, classification by the machine learning models and evaluation of the models on test set using a confusion matrix. Furthermore, for the Linear SVM, a grid-search method is performed to find the optimal value of the Cost parameter. As a result, the SVM model will achieve its best performance on the dataset. Finally, a GUI system that implemented the theme identifier also had developed. Figure 1 shows the framework of the proposed method.

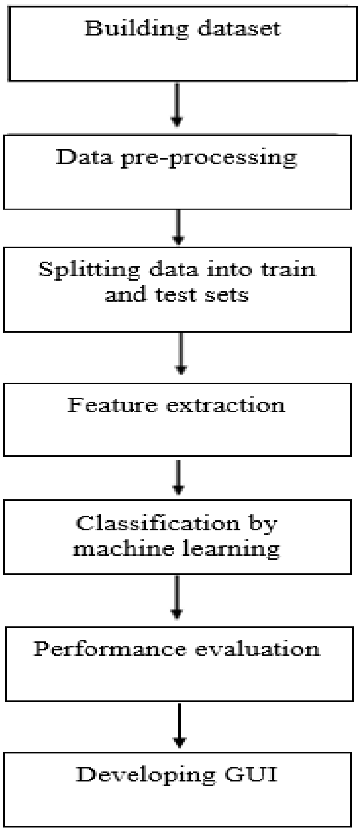

Figure 1. Framework design

Dataset Construction

The first step in building the theme identifier is to construct a dataset by downloading many PDF articles related to machine learning and augmented reality from available online resources such as IEEE Xplore and Google Scholar. Next, the articles for machine learning were searched using relevant keywords such as text classification, machine learning, sentiment analysis, supervised machine learning and feature extraction. On the other hand, the relevant keywords, augmented reality, are being used to search for augmented reality-related articles. Texts from these downloaded PDF files then were extracted into text documents using the PDFMiner library in Python. A total of 1000 text documents for both themes are used in this project, whereas 500 text documents are machine learning related and the remaining 500 text documents are augmented reality related.

Data Pre-processing

The text classification task is a widely used natural language processing task implemented in different applications. Several phases have been done in this step to prepare structured data, such as punctuations, numeric, stop words removal, conversion to lowercase, lemmatization, and removal of short words. Figure 2 shows the steps for data pre-processing. First, punctuation marks and digits are being removed, leaving only words in the text. Then, all uppercase letters are converted into lowercase letters as the feature extractor recognizes the same words with different letter cases as different features. Next, words with less than four letters were removed as most existing words have more than three letters; thus, removing words with less than four letters will reduce the sparse vector as words that rarely occur within the dataset were removed. Then, the stop words are unimportant and do not contribute to the theme. Then, tokenization of texts was done such that the texts in each document were separated into smaller parts called tokens in preparation for lemmatization. During the lemmatization, plural words were converted into their root words, called a lemma. For example, in lemmatization, the word "algorithms" will be reduced into its root word "algorithm".

Splitting Data

The next step is to divide the dataset randomly using train- test split approach in Cross-Validation to validate the trained model as the trained model might not work well even after the training session. In this project, 80% of the dataset were used to train the machine learning models. The remaining data were used to validate the trained model during the validation session. Table 1 shows the dataset for classification.

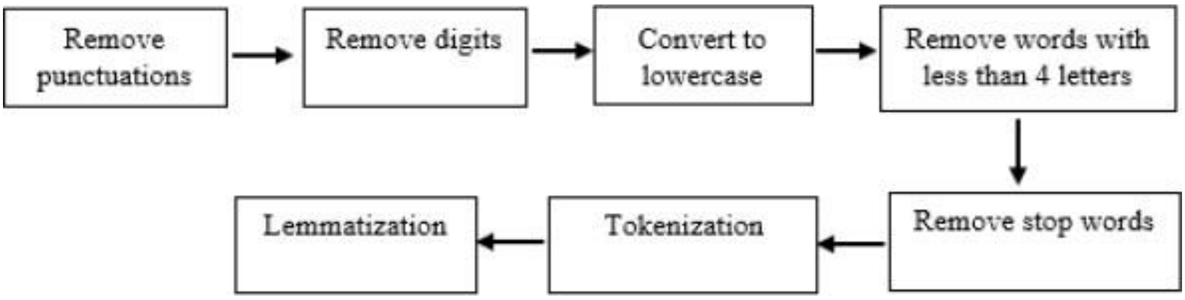

Figure 2. Data pre-processing steps

Table 1. Dataset for Theme Classification

Themes	Number of Documents	Number of Documents	Number of Documents
Augmented Realty	500	404	96
Machine Learning	500	396	104
Total	1000	800	200

TF-IDF Feature Extraction

The proposed method for feature extraction in this project is TF- IDF approach. It is being employed to transform the text into vector representation that the machine learning algorithm can understand. This method considers the relevance between words within the text corpus to calculate the TF-IDF scores for each word at such Term Frequency (TF) and Inverse Document Frequency (IDF) are being calculated. TF represents the word frequency, *t* occurs in each document while IDF is the inverse of the number of documents consisting of the term, t. The TF-IDF function for text vectorization can be written in (1). To improve the classifier performance, further text reduction has been done by setting the parameter of TF-IDF vectorizer, max_df as 0.95 to remove the words that occurred in more than 95% of PDF documents as most of the articles contain words such as "introduction", "conclusion", "abstract" and "methodology". Moreover, to reduce the sparse vector, min_df was set at 10 to remove words that occurred in less than 11 documents.

$$F\ IDF(w) = \frac{n.count(w)}{ln\frac{n}{m+1} \cdot \sum_{j=1}^{m} count(w^{t_j})} \tag{1}$$

Data Classification

In this project, supervised machine learning techniques, Linear SVM and Multinomial Naïve Bayes were employed for the classification process and evaluated to determine which classification model has better performance. As a result, the classification model with the highest accuracy is implemented in the GUI system.

Support Vector Machine

A Linear SVM with a soft margin has been used for the classification task in this project. Classification using an SVM classifier is being done by finding the optimal hyperplane that maximizes the margin between augmented reality and machine learning themes on the vector space. The function for the optimal hyperplane is expressed in (2) and functions that defined the marginal hyperplanes on both sides of classes to develop the margin are expressed in (3) and (4).

$$Wx_j + b = 0 \tag{2}$$

$$Wx_j + b = -1 + \xi_i \tag{3}$$

$$Wx_j + b = 1 + \xi_i \tag{4}$$

ξ_i in the equation is a slack variable that signifies the amount of error allowed for the data points to be inside the margin. Thus, the other data points of class 1 can be described by (5) and data points of class -1 as (6).

$$Wx_j + b \leq -1 + \xi_i \tag{5}$$

$$Wx_j + b \geq 1 + \xi_i \tag{6}$$

The hyperplane construction with a soft margin technique is influenced by the Cost parameter, C. The cost parameter is referred to as the cost of misclassifying data points. The parameter affects the margin and the classification error. In this case, a low value of C will generate a hyperplane with a large margin, thus allowing more misclassifications of training data points to obtain the large margin. Meanwhile, a high value of C will generate a hyperplane with a small margin which leads to fewer or perfectly classified training data points if it is possible. Therefore, this parameter can prevent the machine learning model from overfitting and under-fitting. To avoid the model from being under-fitting and overfitted, the optimal value of the Cost parameter was determined using the grid-search method. Overfitting the machine learning model happens when the model constructs a hyperplane that tends to classify perfectly. The training dataset, if possible, up to the point that when the hyperplane is generalized into the testing dataset, many misclassifications will occur. Meanwhile, underfitting is the phenomenon when the misclassification rate on the training dataset is very high. Thus, even when the hyperplane is generalized into the testing set, the misclassification error will also be high. The appropriate values for initialization of the parameter in the grid-search method were set at such $C(2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 1, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, 2^{6}, 2^{7})$.

Multinomial Naïve Bayes

Another classification has been done using Multinomial Naïve Bayes algorithm at such the classification are done by finding the probability of the document belongs to the theme class. Classification using the Multinomial Naïve Bayes approach are being done based on the Naïve Bayes theorem, whereas the probability of class p given the document will be calculated. The probability of the document belonging to each theme is calculated and compared at such theme of the document determined by the highest probability obtained. The probability of a test document n belonging to class p is being calculated using (7).

$$P(p|n) = P(p) \times \Pi 1 < z < n_z P(t_z|p) \tag{7}$$

$P(p)$ is the prior probability of article class p which is found using (8) and $P(t_z|p)$ is the conditional probability of each word found using t_Z in the given test sample calculated using (9).

$$P(p) = \frac{Number\ of\ documents\ under\ class\ p}{Total\ number\ of\ documents\ in\ text\ corpus} \tag{8}$$

$$P(t_z|p) = \frac{count(t_z|p) + 1}{count\left(t_p\right) + |V|} \tag{9}$$

In (9), $count(t_z|p)$ is the summation of TF-IDF scores of the word tz that presents in all documents underclass p and $count\left(t_p\right)$ refers to the summation of TF-IDF scores of words in all documents underclass p. Last but not least, $|V|$ is the total words that consist in all text documents of train set, not considering how many times the word occurs.

Performance Evaluation

The accuracy classifying the test set are being calculated based on the confusion matrix formula as described in (10) to evaluate the performance of both classification models. TP and TN are the numbers of correctly classified articles, while FP and FN are the numbers of falsely classified documents into another theme. Table 2 shows the confusion matrix for evaluation.

Table 2. Confusion Matrix to Evaluate Binary Classifiers

		Actual	
		Augmented Reality	**Machine Learning**
Prediction	Augmented Reality	TP	FP
	Machine Learning	FN	TN

$$Accuracy(A) = \frac{TP + TN}{(TP + TN + FP + FN)} \qquad (10)$$

GUI Development

The GUI application has been developed using the Tkinter library in Python with the function to select a folder, identify the theme of each article inside the selected folder and organize the articles into the newly created folders based on their themes. For this project, the articles inside the folder can be classified into either machine learning or augmented reality related articles where the best classification model performs theme identification. Figure 3 shows the working principle of the developed GUI system.

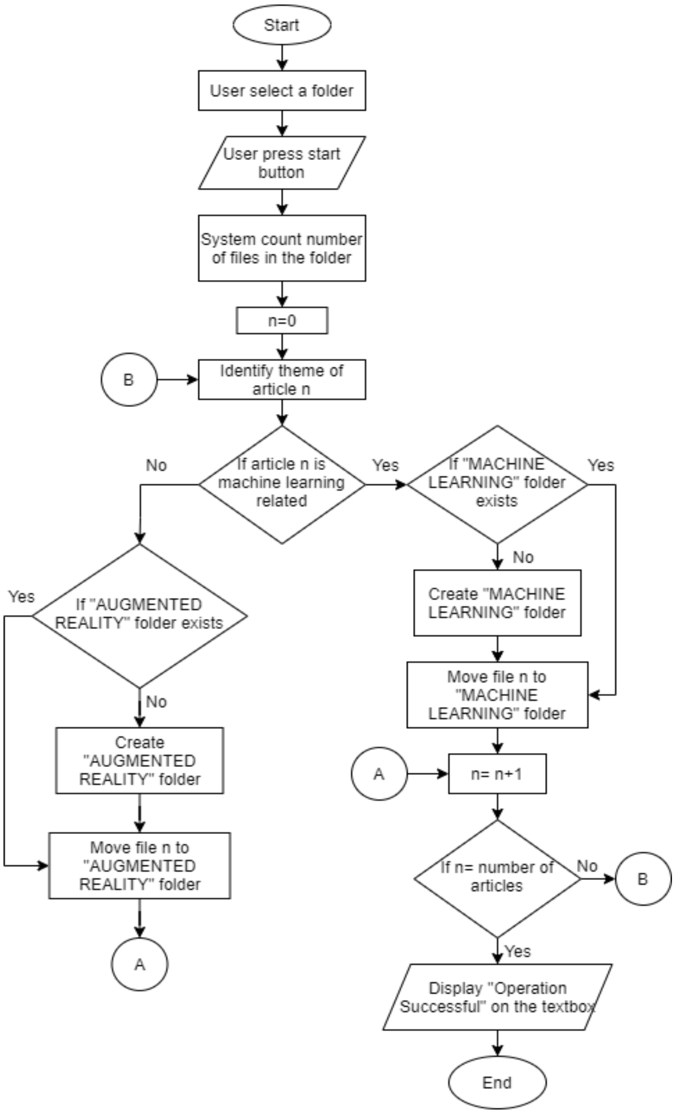
Start
User select a folder
User press start button
System count number of files in the folder
n=0
B
Identify theme of article n
If article n is machine learning related
No
Yes
If "MACHINE LEARNING" folder exists
Yes
No
Create "MACHINE LEARNING" folder
Move file n to "MACHINE LEARNING" folder
Yes
If "AUGMENTED REALITY" folder exists
No
Create "AUGMENTED REALITY" folder
Move file n to "AUGMENTED REALITY" folder
A
A
n= n+1
If n= number of articles
No
B
Yes
Display "Operation Successful" on the textbox
End

Figure 3. Flowchart of the GUI System

RESULT AND ANALYSIS

Data Construction and Pre-processing

From the generated text document using the constructed code, all information contained inside all three PDF document pages except the figure and table has been successfully extracted into the text document. Moreover, the data pre-processing step successfully reduced the number of words in the dataset. Table 3 shows the words reduction after each pre-processing stage; moreover, the 20 most occurring words after this step reflect both themes.

Table 3. Number of Words After Each Pre-processing Stage

Steps	Number of Words Remained
Punctuation Removal	193,925
Digit Removal	142,285
Short Words Removal	113,458
Stop Words Removal	113,384
Lemmatization	108,846

Feature Extraction

During feature extraction, TF-IDF also has proven to reduce the text dimension by eliminating words that occurred in less than 11 documents and more than 95% of text documents. In this case, the number of words has been reduced from 108 846 words before TF-IDF to 7996 words after the feature extraction. Table 4 shows the division of the dataset after splitting and the number of features after TF-IDF.

Table 4. Dataset for Theme Classification

Shape	Documents	Features
Data		
Training Set	800	7,996
Training Set	200	7,996

Grid-Search Method for Cost Parameter Optimization of Linear SVM

Next, from the grid-search method conducted on the training dataset, it is determined that the optimal value of the Cost parameter is 2^{-4}. It is found that the mean classification accuracy during cross-validation is the highest, with 98.75% accuracy compared to classification accuracy when using other initialized Cost parameter values. Thus, the Cost parameter was initialized with 2^{-4} in the Linear SVM model. Table 5 shows the mean classification accuracy based on 5-fold cross-validation in the grid-search method.

Influence of Cost parameter Values on the SVM Model Performance

The accuracy of Linear SVM on the training and testing dataset obtained based on the confusion matrix is recorded and analyzed. From Table 6, it can be seen that the lowest classification accuracy is obtained on both training and testing sets when the Cost parameter is initialized with 2^{-7}. In this case, the classification model is under-fitting as it has poor performance during training and validation sessions. In addition, it can be observed that the number of misclassified data points on the training set is higher compared to the testing set when the Cost parameter is being initialized with 2^{6}, 2^{7} and 2^{8}. This represents that the classification model is likely to be overfitted. The model tried to perfectly classify the training data as such the generalized hyperplane on the testing data causing many misclassifications of data points.

Table 5. Mean Accuracy of Each Initialized Cost Parameter

Cost Parameter, C	Mean Accuracy (%)
2^{-7}	50.50
2^{-6}	90.63
2^{-5}	98.50
2^{-4}	98.75
2^{-3}	98.75
2^{-2}	98.75
2^{-1}	98.63
2^{0}	98.63
2^{1}	98.38
2^{2}	98.13
2^{3}	97.75
2^{4}	97.50
2^{5}	97.13
2^{6}	97.13
2^{7}	97.00
2^{8}	97.00

Table 6. Performance of Linear SVM with each initialized parameter value on train and test set

Cost Parameter, C	Training Set					Training Set				
	Augmented Reality (TP)					Augmented Reality (TP)				
	TP	FP	TN	FN	Acc (%)	TP	FP	TN	FN	Acc (%)
2^{-7}	404	0	2	394	50.75	96	0	0	104	48.00
2^{-6}	404	0	385	11	98.63	96	0	104	0	100.00
2^{-5}	404	0	385	11	98.63	96	0	104	0	100.00
2^{-4}	404	0	388	8	99.00	96	0	104	0	100.00
2^{-3}	404	0	388	8	99.00	96	0	104	0	100.00
2^{-2}	404	0	385	7	99.13	96	0	104	0	100.00
2^{-1}	404	0	389	5	99.38	96	0	104	0	100.00
2^{0}	404	0	391	5	99.50	95	1	104	0	99.50
2^{1}	404	0	392	4	99.63	95	1	104	0	99.50
2^{2}	404	0	393	3	99.63	95	1	104	0	99.50
2^{3}	404	0	393	3	99.63	95	1	103	1	99.00
2^{4}	404	0	393	3	99.63	95	1	103	1	99.00
2^{5}	404	0	393	3	99.63	95	1	103	1	99.00
2^{6}	404	0	394	2	99.75	93	3	101	3	97.00
2^{7}	404	0	394	2	99.75	93	3	101	3	97.00
2^{8}	404	0	394	2	50.75	93	3	101	3	48.00

Evaluation of the Machine Learning Models

The accuracy of each model on the test set is obtained based on the confusion matrix. Figure 4 shows the accuracy comparison between the two models. In this case, the Linear SVM achieved 100% in classifying the validation set compared to Multinomial Naïve Bayes, which achieved 98.50% accuracy. Thus, the Linear SVM classification model is implemented in the GUI.

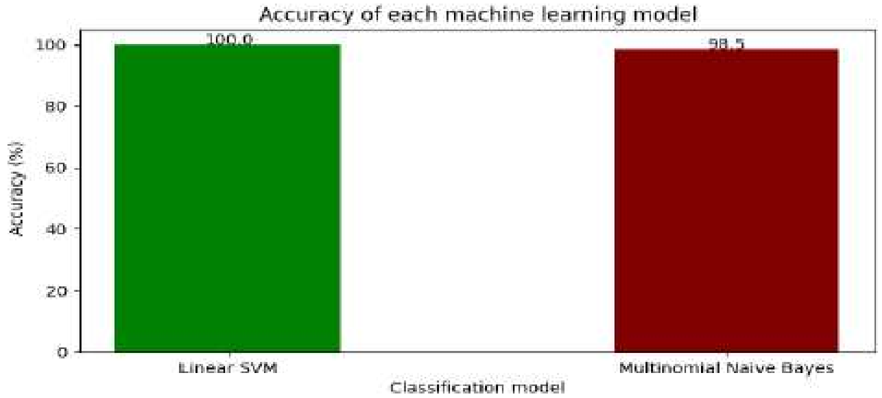

Figure 4. Accuracy of models on the test set

CONCLUSION

Manually organizing many PDF articles is inconvenient as it is energy and time-consuming. Therefore, to organize these PDF articles automatically and in a shorter time, this project introduced methods included in the development of the algorithms for theme identifiers. As a result, the file organizer can be developed by employing the theme identifier. From the end of the project development, it can be concluded that the first objective has been achieved where the suitable machine learning model for text classification on the dataset, which is highly dimensional data, has been determined, which is the Linear SVM model. The next objective is to develop a GUI system that can organize the articles inside the selective folder into machine learning and augmented reality folders based on the identified theme. It has been perfectly achieved since it employed the best classification model to identify the article's theme.

The third and fourth objectives of this project also have been achieved from the literature review where the data pre-processing step had proven to reduce the number of words such that 20 most occurred words within the text corpus after the process reflected both themes. Moreover, the TF-IDF feature extraction reduces the sparse vector, and converting the dataset into a form suitable to be fed into the machine learning model has also been successfully performed. The last objective is to construct the dataset for classification tasks of models on machine learning and augmented reality articles. It has been achieved as such the dataset contributes to building the theme identifier that can differentiate between machine learning and augmented reality related articles. In conclusion, all the objectives have been achieved. Thus, all the research objectives provide the direction in building the system.

REFERENCES

[1] M. K. I. Kassab, S. S. Abu Naser & M. J. Al Shobaki, “An Analytical Study of the Reality of Electronic Documents and Electronic Archiving in the Management of Electronic Documents in the Palestinian Pension Agency (PPA),” *European Academic Research,* vol. 4, no. 12, pp. 10052-10102, 2017.

[2] S. Li, F. Jiao, Y. Zhang, and X. Xu, “Problems and Changes in Digital Libraries in the Age of Big Data from the Perspective of User Services,” *The Journal of Academic Librarianship*, vol. 45, no. 1, pp. 22-30, 2019, doi: 10.1016/j.acalib.2018.11.012

[3] D. Dosso and G. Silvello, "Search Text to Retrieve Graphs: A Scalable RDF Keyword-Based Search System," in *IEEE Access*, vol. 8, pp. 14089-14111, 2020, doi: 10.1109/ACCESS.2020.2966823.

[4] N. Hossain, M. Ghazvininejad, and L. Zettlemoyer, “Simple and Effective Retrieve-Edit-Rerank Text Generation,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 2020, pp. 2532-2538, doi: 10.18653/v1/2020.acl-main.228.

[5] D. Sahoo and R. C. Balabantaray, “A Hybrid Approach to Retrieve Knowledge from a Document,” *International Journal of Knowledge Management (IJKM*), vol. 16, no. 1, pp. 18 pages, 2020, doi: 10.4018/IJKM.2020010104

[6] Z. Saleem, A. Alhudhaif, K. N. Qureshi, and G. Jeon, “Context-aware text classification system to improve

the quality of text: A detailed investigation and techniques," *Concurrency and Computation: Practice and Experience*, Special Issue 2021, doi: 10.1002/cpe.6489

[7] K. Shah, H. Patel, D. Sanghvi & M. Shah, "A Comparative Analysis of Logistics Regression, Random Forest, and KNN Models for the Text Classification*," Augmented Human Research,* vol. 5, no 12, 2020, doi: 10.1007/s41133-020-00032-0

[8] F. A. Bohani, S. R. Yahya, S. N. H. S. Abdullah, "Microgrid Communication and Security: State-Of-The-Art and Future Directions," *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 1, no. 1, pp. 37-52, 2021, doi: 10.51662/jiae.v1i1.10

[9] O. Wisesa, A. Adriansyah, and O. I. Khalaf, "Prediction analysis sales for corporate services telecommunications company using gradient boost algorithm," *2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering (BCWSP)*, 2020, pp. 101-106, doi: 10.1109/BCWSP50066.2020.9249397

[10] Z. P. Putera, M. D. Anasanti, and B. Priambodo, "Designing Translation Tool: Between Sign Language to Spoken Text on Kinect Time Series Data using Dynamic Time Warping," *SINERGI,* vol. 22, no. 2, pp. 91-100, 2018, doi: 10.22441/sinergi.2018.2.004

[11] F. M. Hanis and M. Teimouri, "Dataset for file fragment classification of textual file formats," *BMC Research Notes*, vol. 12, no. 801, 2019, doi: 10.1186/s13104-019-4837-4

[12] V. Karthikeyani and S. Nagarajan, "Machine Learning Classification Algorithms to Recognize Chart Types in Portable Document Format (PDF) Files," *International Journal of Computer Applications*, vol. 39, no. 2, pp. 1-5, 2012

[13] X. Luo, "Efficient English text classification using selected Machine Learning Techniques," *Alexandria Engineering Journal*, vol. 60, no. 3, pp. 3401-3409, 2021, doi: 10.1016/j.aej.2021.02.009

[14] S. O. Olatunji, "Improved email spam detection model based on support vector machines," *Neural Computing and Applications*, vol. 31, no. 3, pp. 691–699, 2019, doi: 10.1007/s00521-017-3100-y.

[15] C. Choi, J. Kim, J. Kim, D. Kim, Y. Bae, and H. S. Kim, "Development of Heavy Rain Damage Prediction Model Using Machine Learning Based on Big Data," *Advance in Meteorology*, vol. 2018, 5024930, 2018, doi: 10.1155/2018/5024930.

[16] X. Wang *et al.*, "Research and Implementation of a Multi-label Learning Algorithm for Chinese Text Classification," *Proc. - 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, 2017, pp. 68–76, doi: 10.1109/BIGCOM.2017.34.

[17] W. A. Qader, M. M. Ameen, and B. I. Ahmed, "An Overview of Bag of Words; Importance, Implementation, Applications, and Challenges," *Proc. 2019 International Engineering Conference (IEC)*, 2019, pp. 1–4, doi: 10.1109/IEC47844.2019.8950616.

[18] P. H. Prastyo, I. Ardiyanto, and R. Hidayat, "Indonesian Sentiment Analysis: An Experimental Study of Four Kernel Functions on SVM Algorithm with TF-IDF," *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, 2020, pp. 1-6, doi: 10.1109/ICDABI51230.2020.9325685.

[19] D. M. J. Tax and R. P. W. Duin, "Using two-class classifiers for multiclass classification," *2002 International Conference on Pattern Recognition*, 2002, vol. 2, pp. 124-127, doi: 10.1109/ICPR.2002.1048253.

[20] R. Saravanan and P. Sujatha, "Algorithms: A Perspective of Supervised Learning Approaches in Data Classification," *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, pp. 1–5, doi: 10.1007/978-981-13- 7403-6_11.