An Improved Model for Legal Case Text Document Classification

May Tamara Stow, Chidiebere Ugwu, and Laeticia Onyejegbu

***Abstract* — Misjudgments in court cases are inevitable in any judicial system irrespective of how civilized the country in which the judicial system is. The economic effects of failed court judgments cannot be overemphasized. The passing of wrong judgments can be a result of a lack of evidence due to poor research by counsels. Preparing for a court case is not an easy fit as a lot of research must be done on the part of the attorneys in charge. This paper presents an improved Hybrid model for legal case document classification. The system starts by collecting legal case documents from an online domain. The collected documents were converted to texts using a pdf miner library in python. The converted texts were used in creating tables using the pandas library. After the creation of the dataset table, the dataset was pre-processed by removing noise, and non-alphanumeric values, and performing tokenization. The tokenized data was then passed into principal component analysis for the selection of important features. The selected features were used in training an LSTM model for the classification of the legal case documents. The system was designed with Object-Oriented Analysis and Design method and implemented using python programming language. The result of the LSTM is outstanding, having an accuracy of 99% when evaluated with unseen legal case documents. The model was deployed in building a web application for the classification of legal documents. Upon testing the application with emerging documents, it sufficiently classified them and reduced tremendously the conflicting judgments experienced before the application of the improved model for legal case classification.**

***Keywords* — Legal Case, Long Short-Term Memory, Principal Component Analysis, Python Flask.**

I. Introduction

Text categorization or classification, according to Sebastiani [12], is defined as assigning new documents to a set of pre-defined categories based on the classification patterns. Classification of text documents is the process of assigning class labels to unseen documents based on the model generated in the training phase. It can also be defined as the task of automatically categorizing collections of electronic documents into their annotated classes, based on their contents [7]. Text document classification plays an important role in providing intuitive navigation and browsing mechanisms by organizing large amounts of information into a small number of meaningful classes [6]. Text Document Classification has remained a reliable and conventional procedure to differentiate texts based on their subjects among scientific texts, web pages, and digital libraries [8].

This is because we are in an era where we deal with a huge amount of data in our daily life. These data contain relevant information that will be retrieved for use at one point or another. Data can be in different forms such as text, image, spatial form, etc. The most common form of data that we come across every day is text data. The news stories we read daily, posts, and messages on social media are mostly in the form of text [14]. With the advancement of technology and reduced storage costs, individuals and organizations are tending towards the usage of electronic media for storing textual information and documents.

Misjudgments are inevitable in any judicial system irrespective of how civilized the country in which the judicial system is. The economic effects of failed court judgments cannot be overemphasized: In a criminal case, an innocent person can be pronounced guilty, and vice versa. In the case of the former, the defendant's image and name will be tarnished, his family members can be ostracized, and if there is a death penalty, the defendant will lose his life. In the case of the latter, a guilty person will walk free and will go into society to probably commit the same crime again.

Failed court judgments can be a result of a lack of evidence due to poor research by either counsel. Preparing for a court case is not an easy feat as a lot of research has to be done on the part of the attorneys in charge. This research often includes delving into cases similar to the case or cases that they are preparing for at the moment. This will help them with ideas and the direction to take to put forth a convincing argument when they are defending their client(s). In the case where court case documents of previous cases are in the form of hard copies and are handled manually, there is the possibility of these methods failing in the case of natural disasters such as fire disasters or flood disasters. Such documents would be destroyed if that were the case. Now, imagine a situation where these lawyers and attorneys can easily and efficiently access case documents that are similar to the cases they are working on. That there is an intelligent system in which they can input certain keywords and all the documents with those keywords will automatically be returned as output. This will surely facilitate the process of preparing for court cases, thereby making lawyers and attorneys better prepared and in the long run, reduce to some extent, the misjudgments and delays that occur in clearing some cases.

The objective of this paper is to address the problem of misclassification in text classification especially in legal case

Submitted on March 02, 2023.
Published on April 26, 2023.

M. T. Stow, Department of Computer Science, University of Port Harcourt, Nigeria.
(corresponding e-mail: maystow@gmail.com)

C. Ugwu, Department of Computer Science, University of Port Harcourt, Nigeria.
(e-mail: chidiebere.ugwu@uniport.edu.ng)

L. Onyejegbu, Department of Computer Science, University of Port Harcourt, Nigeria.
(e-mail: laeticia.onyejegbu@uniport.edu.ng)

DOI: http://dx.doi.org/10.24018/ejece.2023.7.2.509

documents using the improved hybrid model developed in this paper for text document classification. Extensive text pre-processing, dimensionality reduction using Principal Component Analysis (PCA), and Long Short-Term Memory (LSTM) an architecture of Deep Learning was used in the building of the model.

II. Related Works

The reviewed literature gave us insight into what others have done in the area of text document classification and the gaps that these studies were unable to address. In a paper, Wei *et al*. [16] attempted towards integrating WordNet with lexical chains to alleviate these problems. The proposed approach exploits ontology hierarchical structure and relations to provide a more accurate assessment of the similarity between terms for word sense disambiguation. Furthermore, they introduced lexical chains to extract a set of semantically related words from texts, which can represent the semantic content of the texts. Their integrated way can identify the theme of documents based on the disambiguated core features extracted, and in parallel downsize the dimensions of feature space. The experimental results using their proposed framework on Reuters-21578 show that clustering performance improves significantly compared to several classical methods. However, there were some limitations to their research. Some important words which were not included in the WordNet lexicon were not considered concepts for similarity evaluation.

A legal case document classification system was developed using a hybrid approach [15]. The architecture of the system integrates stages that include document collection, document pre-processing, dimensionality reduction, vectorization, training, and classification. The system employed preprocessing techniques to prepare the document features. The probabilistic nature of the Naïve Bayes algorithm was integrated to generate vectorized data from the document features for the classifier and the most important features were selected by feature ranking using the Chi-square method. For the final classification, the Support Vector Machine was used. The performance of the system was evaluated using precision and recall and accuracy metrics. Precision means the percentage of your relevant results [13]. Recall refers to the percentage of total relevant results correctly classified by your algorithm [13]. During the evaluation performance against the legal case documents, when the support vector machine polynomial kernel was used, a recall of 76.92% was obtained in the housing category. However, in the evaluation performance against the legal case documents when the support vector machine sigmoid kernel was used, some categories but not all, had a precision of 100%. The remaining categories had a precision below 95%. The 100% means the percentage of their results which were relevant. The 76.92% indicates the percentage of the total relevant results correctly classified by their algorithm. From the result of the recall accuracy, there is a need to improve the performance of their legal case document classification system. Also, even though it was not stated in their paper, a limitation of support vector machines is speed and size, both in training and testing [2]. For this classification system, the system was trained and tested with very few documents (less than 100). What then happens in a situation where there is a plethora of documents to train, test and classify? This is usually the challenge of most machine learning classification systems: only a small amount of corpus is used for training and testing.

Pinto and Melgar [10] proposed a classification model for Portuguese documents in the juridical domain. The paper was motivated by the huge number of notifications received daily by the attorney's office in Brazil. Their model is developed to understand the meaning of each notification and indicate what kind of response should be prepared for any situation. Machine learning algorithms were used. The problem was modeled as a classification problem using free text documents. However, their system was built using a specific collection of documents. This makes the system inflexible as the system cannot be applied to another domain of interest. Also, the performance of their model was restricted to the use of ROC graph.

Al-Khurayji and Sameh [1] proposed an approach for use in Arabic text classification using a classifier known as Kernel Naive Bayes (KNB). The following techniques for pre-processing were used. They are: Arabic words light Stemmer, Stop word removal, and word tokenization. TF-IDF technique was applied as well for Arabic word feature extraction so that they will be converted to vector space for normalized classification. The results showed that the proposed classifier got good results considering the accuracy and time which is contrary to other classifiers used in the previous studies. It is hereby concluded that the Arabic text classification of electronic documents using the KNB that was proposed indicated better performance than others.

Categorization of Supreme Court Cases using Multiple Horizontal Thesauri is an interesting paper on document classification by Pudaruth *et al.* [11]. In their work, several lexicons were created for some predefined areas of law through an automated approach. The lexicons were used to categorize cases from the Supreme court into eight distinct areas of law. The performance of these lexicons was compared with each other. Their results showed that lexicons with a mixture of single words and short phrases performed slightly better than those consisting simply of single words. Their best model achieved a global accuracy of 78.9%. However, this result faired less than those results gotten from machine learning approaches which are usually above 80%.

Fusheng *et al.* [4] worked on an empirical study of deep learning for text classification in legal documents. They conducted experiments to compare deep learning results with results obtained using an SVM algorithm on the four datasets of real legal matters. Their results showed that CNN performed better with a larger volume of training datasets and should be a fit method for text classification in the legal industry.

Due to the challenge of multi-label text classification in Arabic language [3] introduced a new rich and unbiased dataset for both the single-label (SANAD) as well as the multi-label (NADiA) Arabic text categorization tasks. They presented an extensive comparison of several deep learning (DL) models for Arabic text categorization to evaluate the effectiveness of such models on SANAD and NADiA.

A unique aspect of their work was that it did not require a pre-processing phase but was able to produce a classification accuracy of 96.94%.

Mohammed and Kora [9], proposed a new meta-learning ensemble method that fuses baseline deep learning models using 2-tiers of meta-classifiers. They conducted several experiments on six public benchmark datasets to evaluate the performance of the proposed ensemble. For each benchmark dataset, committees of different deep baseline classifiers are trained, and their best performance is compared with the performance of the proposed ensemble.

In a bid to extract important information from discharge medical notes written by Physicians [5], explored the model performance of various deep learning models in text classification tasks on medical notes for different disease class imbalance scenarios. Their results showed that for classification tasks on medical notes, Transformer encoders are the best choice if the computation resource is not an issue. Otherwise, when the classes are relatively balanced, CNNs are a leading candidate because of their competitive performance and computational efficiency.

The new model handles the issue of misclassification and improves classification performance by capturing and maintaining the context of words in the sentences of each document and uses that information to make more accurate predictions.

III. Materials and Methods

Fig. 1 shows an extension of the Text Document Classification architecture adopted by Ugwu and Obasi [15].

The PCA component was incorporated to reduce the dimensionality of the text representation to make the classifier algorithm faster and more scalable. The LSTM component was inculcated in building the model to capture the context of words in a sentence and use that information to make more accurate predictions. The architecture used legal case data which comprises good, quality legal case documents for training and testing the proposed system for the first classification task. All data collected were in pdf format but were converted to text format by the Pdf-to-Text converter component. The total number of documents collected was 1500 legal case documents.

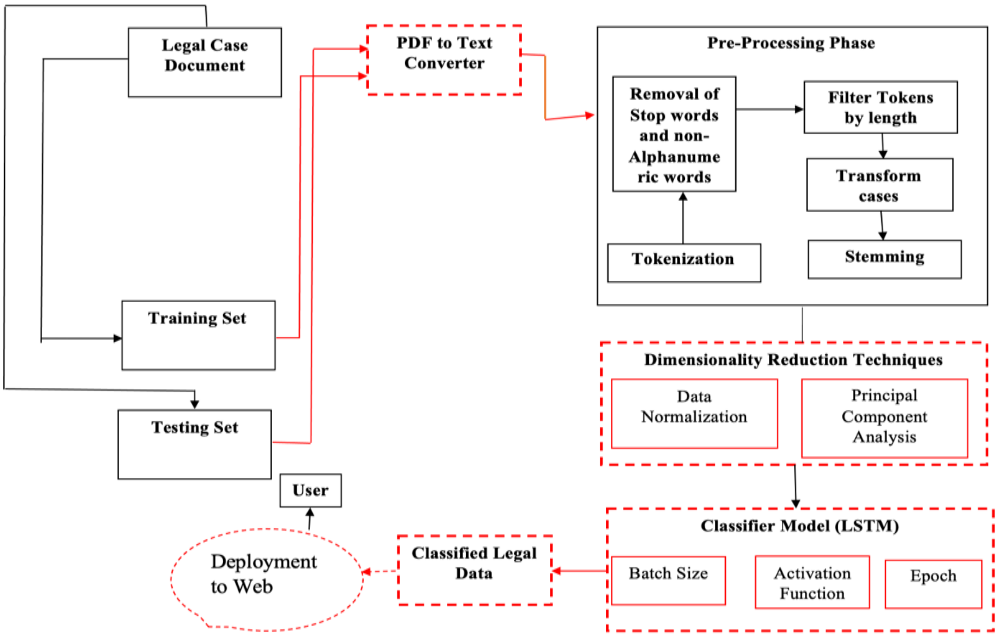


Fig. 1. Architectural design of the proposed system.

TABLE I: Sample of Pre-Processed Data

S/N	Text_Data	text_clean	text_tokens
1	Case Title: ABEKE v. ODUNSI & ANOR (2013) LP...	case title abeke v odunsi anor lpelrsc abe...	[case, title, abeke, v, odunsi, anor, lpelrsc,...
2	Before Our Lordships Muhammad Saifullahi Munta...	before our lordships muhammad saifullahi munta...	[before, our, lordships, muhammad, saifullahi,...
3	"However, where a tenant for a fixed term refu...	however where a tenant for a fixed term refuse...	[however, where, a, tenant, for, a, fixed, ter...
4	&amp; Anor vs. B'asil O. Ezegbu &amp; Anor (19...	amp anor vs basil o ezegbu amp anor nwlr pt ...	[amp, anor, vs, basil, o, ezegbu, amp, anor, n...
5	INTRODUCTION: THIS APPEAL BORDERS ON LANDLORD ...	introduction this appeal borders on landlord a...	[introduction, this, appeal, borders, on, land...
6	DECISION/HELD: IN THE FINAL ANALYSIS, THE SUPR...	decisionheld in the final analysis the supreme...	[decisionheld, in, the, final, analysis, the, ...
7	1 ESTATE OF MICHAEL ABIODUN JOSEPH FOR THE S...	estate of michael abiodun joseph for the su...	[estate, of, michael, abiodun, joseph, for, th...
8	2 TESTIFIED THAT HE WAS NOT A TENANT IN THE...	testified that he was not a tenant in the ...	[testified, that, he, was, not, a, tenant, in,...
9	COURT, PARTIES FILED AND EXCHANGED THEIR BRIEF...	court parties filed and exchanged their briefs...	[court, parties, filed, and, exchanged, their,...

These 1500 legal case documents were classified into six categories namely, Civil, Criminal, Housing, Political/Government, Finance, and Land. The 1500 legal case text documents were used to synthetically generate more legal case documents to serve as input for the proposed system. The data generated from the 1500 legal case text documents was a little above 12,000 legal case documents.

Fig. 2 shows a histogram distribution of the total number of documents present in the dataset.

The pre-processing was handled in five stages. The tokenization step involved the breaking down of the legal case contents into tokens. The stop-words and non-alphanumeric words were removed from the corpus when passed through the list of stop-words and in the second stage. Filtering by length (which has to do with removing unwanted words) was done in the third stage after which transforming all characters to lowercase was done in the fourth stage. Stemming and lemmatization was done in the final stage of pre-processing. The sample output of the pre-processed data can be seen in Table I. The pre-processed data was used for the PCA as input for feature selection. The result of the PCA can be seen in Fig. 3. The result of the PCA was used as input for the LSTM model. The architecture that shows these processes can be seen in Fig. 4.

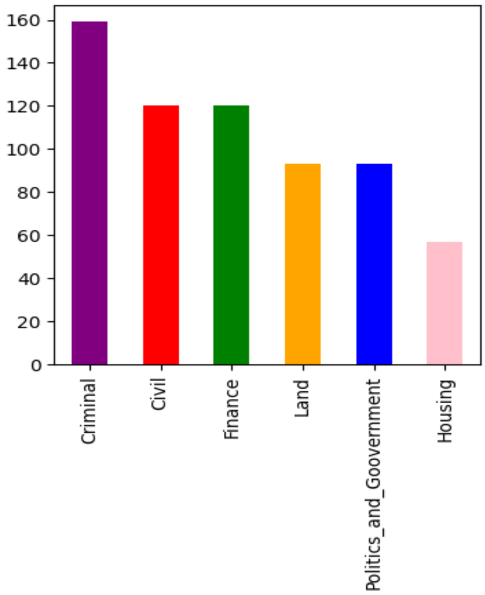


Fig. 2. Histogram distribution of the training data.

The histogram distribution in Fig. 2 shows the total number of cases that are in the dataset for the six categories of legal case document classifications. It shows that civil court cases appear more in the dataset than any other court case. This was achieved using the pandas library in python. The histogram representation was used to tell how many civil cases, criminal cases, finance cases, housing cases, land cases, and politics and government cases were present in our dataset. The representation shows that civil cases had over 2000 reports, the criminal case category has about 1600 reports, politics, and government, 1450, land, 1400, housing 700, and finance 250.

The algorithm for Principal Component Analysis used is as follows:

- Step 1: Standardize the legal case data.
- Step 2: Calculate the covariance matrix for the input features in the Botnet IoT data.
- Step 3: Calculate the eigenvalues and eigenvectors for the covariance matrix.
- Step 4: Sort eigenvalues and their corresponding eigenvectors.
- Step 5: Select k eigenvalues and form a matrix of eigenvectors.
- Step 6: Perform a transformation of the original matrix.

The output from PCA can be seen in Fig. 3, which represents the most important features that were gotten from the legal case document. These data are represented in an array format. The digits here represent the characters that were broken down into various tokens. These tokens represent various court case statements that have been tokenized and converted to an array. These selected features in an array format were used as input data in training the LSTM model for legal case classification.

The LSTM model in Fig. 4 consists of the cell state, hidden layer, input layer, activation functions (sigmoid and tanh), and the output layer.

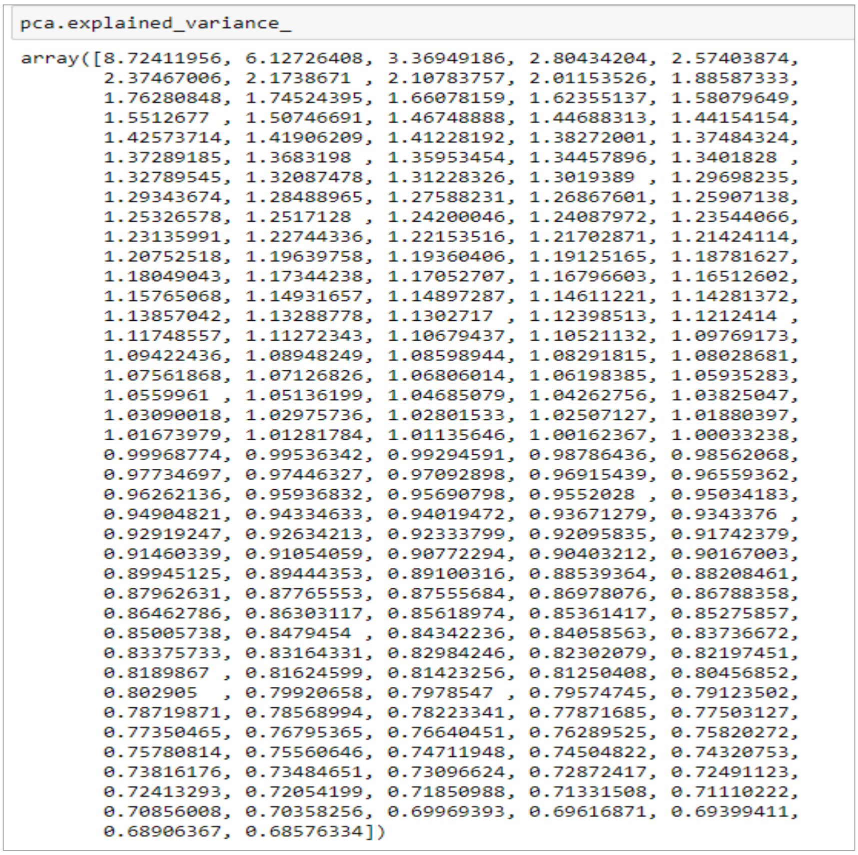
The cell state (long-term memory) is responsible for taking information (the result of the PCA), processing them, and keeping the information in the hidden state, and the output layer is the tensor of all the hidden layers. It displays the final result of the LSTM model.

A sequential LSTM model was used. The term sequential simply means that the model processes sequences of integers, embedding its integer into a 64- dimensional vector then processes these sequences of vectors using the LSTM layer. An embedding layer and spatial dropout are used in reducing overfitting in the training of the Long Term-Short. An optimizer was used to increase the learning rate of the model and the metrics variable is used to hold the loss values and accuracy gotten during the training process. The LSTM model was trained using the result of the PCA as input on a training step of 20, and a batch size of 32. The batch size determines the time taken for the LSTM model to complete a training step. At each training step, the result of the PCA served as input to the LSTM model, the LSTM model processed the data using the cell state and stored the result in the hidden layer. The LSTM model repeats this process twenty times and saves the result in the hidden state. With this, the LSTM model was able to tell if a legal case document belongs to either civil, land, housing, finance, politics, or government cases. The output layer was responsible for showing the six categories of legal case documents.

IV. Experiments and Results

Below, we describe in detail the setup used for the experiments and report on some of the significant results we obtained.

For the system implementation, Bootstrap framework, Flask framework, Python programming language, and MySql Database were used. The development tools and environment used was Jupyter Notebook, Sypder, and Anaconda (Python Distribution). The software requirements are a web browser and Microsoft Windows.



```
pca.explained_variance_
```

```
array([8.72411956, 6.12726408, 3.36949186, 2.80434204, 2.57403874,
       2.37467006, 2.1738671 , 2.10783757, 2.01153526, 1.88587333,
       1.76280848, 1.74524395, 1.66078159, 1.62355137, 1.58079649,
       1.5512677 , 1.50746691, 1.46748888, 1.44688313, 1.44154154,
       1.42573714, 1.41906209, 1.41228192, 1.38272001, 1.37484324,
       1.37289185, 1.3683198 , 1.35953454, 1.34457896, 1.3401828 ,
       1.32789545, 1.32087478, 1.31228326, 1.3019389 , 1.29698235,
       1.29343674, 1.28488965, 1.27588231, 1.26867601, 1.25907138,
       1.25326578, 1.2517128 , 1.24200046, 1.24087972, 1.23544066,
       1.23135991, 1.22744336, 1.22153516, 1.21702871, 1.21424114,
       1.20752518, 1.19639758, 1.19360406, 1.19125165, 1.18781627,
       1.18049043, 1.17344238, 1.17052707, 1.16796603, 1.16512602,
       1.15765068, 1.14931657, 1.14897287, 1.14611221, 1.14281372,
       1.13857042, 1.13288778, 1.1302717 , 1.12398513, 1.1212414 ,
       1.11748557, 1.11272343, 1.10679437, 1.10521132, 1.09769173,
       1.09422436, 1.08948249, 1.08598944, 1.08291815, 1.08028681,
       1.07561868, 1.07126826, 1.06806014, 1.06198385, 1.05935283,
       1.0559961 , 1.05136199, 1.04685079, 1.04262756, 1.03825047,
       1.03090018, 1.02975736, 1.02801533, 1.02507127, 1.01880397,
       1.01673979, 1.01281784, 1.01135646, 1.00162367, 1.00033238,
       0.99968774, 0.99536342, 0.99294591, 0.98786436, 0.98562068,
       0.97734697, 0.97446327, 0.97092898, 0.96915439, 0.96559362,
       0.96262136, 0.95936832, 0.95690798, 0.9552028 , 0.95034183,
       0.94904821, 0.94334633, 0.94019472, 0.93671279, 0.9343376 ,
       0.92919247, 0.92634213, 0.92333799, 0.92095835, 0.91742379,
       0.91460339, 0.91054059, 0.90772294, 0.90403212, 0.90167003,
       0.89945125, 0.89444353, 0.89100316, 0.88539364, 0.88208461,
       0.87962631, 0.87765553, 0.87555684, 0.86978076, 0.86788358,
       0.86462786, 0.86303117, 0.85618974, 0.85361417, 0.85275857,
       0.85005738, 0.8479454 , 0.84342236, 0.84058563, 0.83736672,
       0.83375733, 0.83164331, 0.82984246, 0.82302079, 0.82197451,
       0.8189867 , 0.81624599, 0.81423256, 0.81250408, 0.80456852,
       0.802905  , 0.79920658, 0.7978547 , 0.79574745, 0.79123502,
       0.78719871, 0.78568994, 0.78223341, 0.77871685, 0.77503127,
       0.77350465, 0.76795365, 0.76640451, 0.76289525, 0.75820272,
       0.75780814, 0.75560646, 0.74711948, 0.74504822, 0.74320753,
       0.73816176, 0.73484651, 0.73096624, 0.72872417, 0.72491123,
       0.72413293, 0.72054199, 0.71850988, 0.71331508, 0.71110222,
       0.70856008, 0.70358256, 0.69969393, 0.69616871, 0.69399411,
       0.68906367, 0.68576334])
```

Fig. 3. Result of Principal Component Analysis.

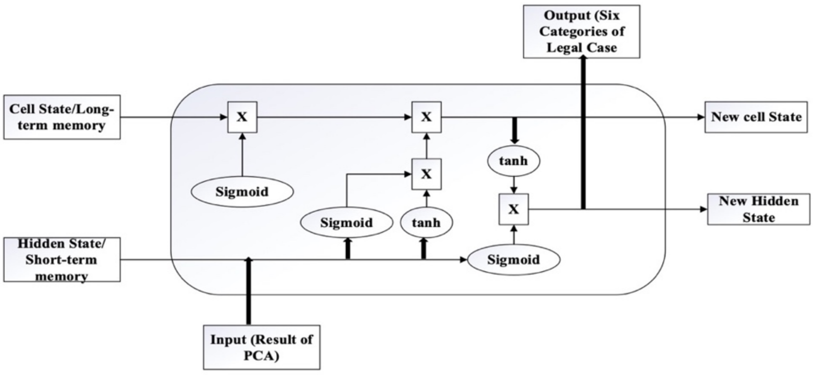

Output (Six Categories of Legal Case
Cell State/Long-term memory
X
X
New cell State
tanh
X
Sigmoid
Sigmoid
tanh
X
New Hidden State
Hidden State/ Short-term memory
Sigmoid
Input (Result of PCA)

Fig. 4. Architecture of LSTM.

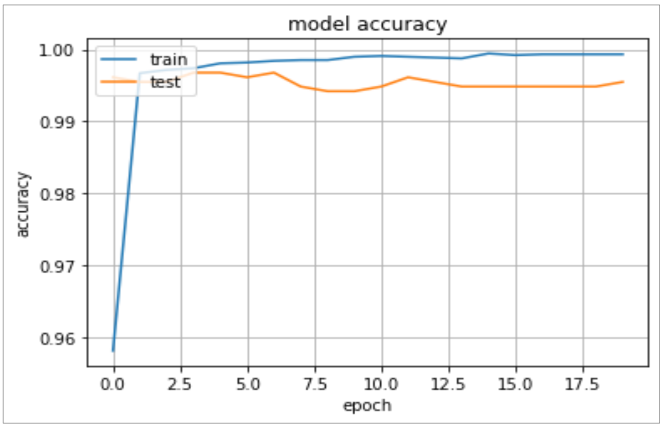


Fig. 5. Accuracy of the Trained Model.

By accuracy, we mean how well the model performed during training. Fig. 5 shows the model achieved an accuracy of 99.91% for the training data and about 99.61% for the validation or testing data. The blue line represents the model training accuracy, whereas the orange line represents the validation test accuracy. By validation test, we simply mean the evaluation of the model performance by using testing data. The line graph shows the performance of the model at each training step. The line graph shows the performance of the model at each training step. In step 1, the training performance of the model was 95.81% and the validation score was 99.61%, in step 5 the training performance of the model was 99.82% and the validation data was 99.61%, in step 10, the model had a training performance of 99.91% and the validation score was 99.59%.

The losses acquired by the model during training and testing are shown in Fig. 6. The green line indicates the loss acquired by the model during training, and the orange line indicates the loss acquired by the model during testing. The loss values are acquired at each training step, starting from step 1 to step 20. By loss values, we mean the losses the model had during training.

DOI: http://dx.doi.org/10.24018/ejece.2023.7.2.509

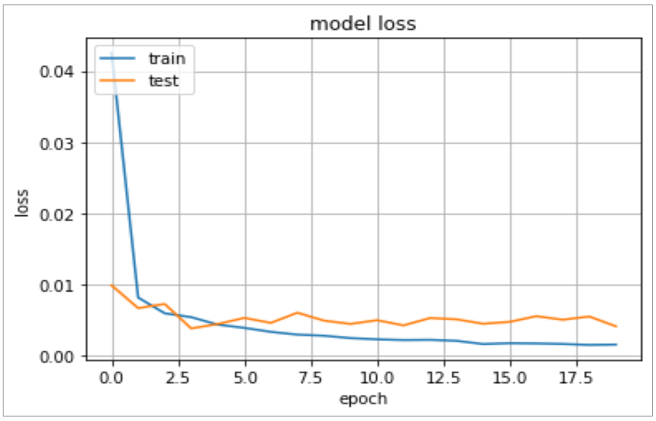


Fig. 6. Loss value of the trained Model.

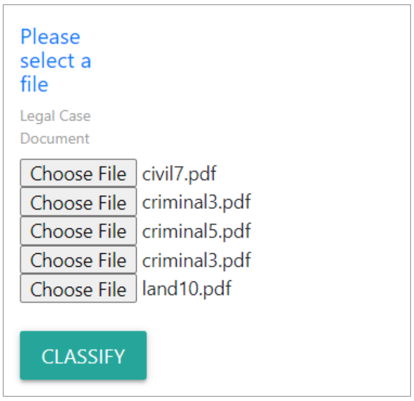
This shows that model achieved a loss value of about 0.0023% for the training data and 0.005% for the validation or testing data.

TABLE II: CLASSIFICATION REPORT

	precision	recall	fl-score	support
0	1.00	1.00	1.00	1810
1	1.00	1.00	1.00	1275
2	1.00	0.97	0.98	221
3	1.00	1.00	1.00	470
4	1.00	1.00	1.00	1061
5	1.00	1.00	1.00	1151
micro avg	1.00	1.00	1.00	5988
macro avg	1.00	0.99	1.00	5988
weighted avg	1.00	1.00	1.00	5988
samples avg	1.00	1.00	1.00	5988

Table II shows a detailed report of the model performance in terms of accuracy, precision, recall, f1-score, and support. Precision refers to the number of true positives divided by the total number of positive predictions. Recall quantifies the number of positive class predictions, made out of all positive examples, and support means the total prediction made by the model on each of the categories. The precision score shows 100% for civil cases, 100% for criminal cases, 97% for finance cases, 100% for housing cases, 100% for land cases, and 100% for politics and government cases, and finally, the evaluation of the model's performance on the never-before-seen dataset is 99.58% which the system approximated to 100%. The classification report above shows that the model is in good shape. Further testing will of the model will be carried out on real-time documents (legal cases documents).

The user testing phase involved testing fresh legal case documents on our model to see how it performs. The user selects files from their computer that they want to classify.


Please select a file

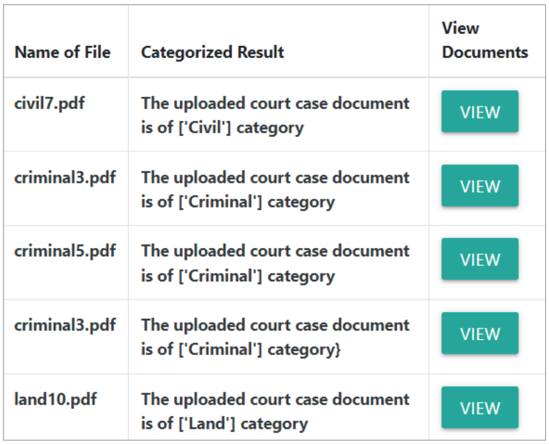
Legal Case Document

Choose File civil7.pdf
Choose File criminal3.pdf
Choose File criminal5.pdf
Choose File criminal3.pdf
Choose File land10.pdf

CLASSIFY

Fig. 7. Sample of the Selected Files.

Fig. 7 shows us the files the user selected for classification. After the selection process, the user clicks on the classify button.

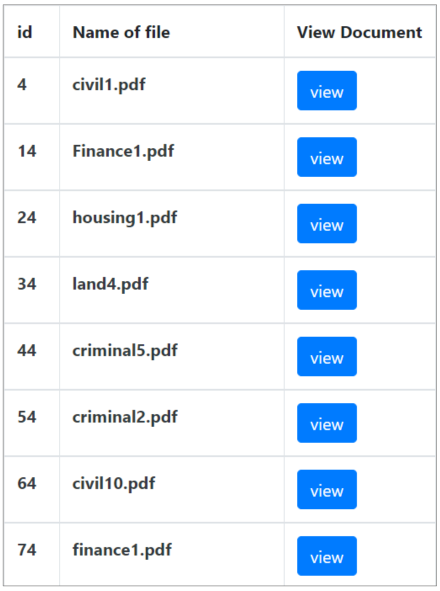


Name of File	Categorized Result	View Documents
civil7.pdf	The uploaded court case document is of ['Civil'] category	VIEW
criminal3.pdf	The uploaded court case document is of ['Criminal'] category	VIEW
criminal5.pdf	The uploaded court case document is of ['Criminal'] category	VIEW
criminal3.pdf	The uploaded court case document is of ['Criminal'] category}	VIEW
land10.pdf	The uploaded court case document is of ['Land'] category	VIEW

Fig. 8. Results of the Classified Documents.

Fig. 8 shows us the results of the classification of the documents selected by the user. The documents are classified accurately. The "VIEW" option allows the user to view the contents of the document.

The user also has the option to add new documents which will be saved for future use.



id	Name of file	View Document
4	civil1.pdf	view
14	Finance1.pdf	view
24	housing1.pdf	view
34	land4.pdf	view
44	criminal5.pdf	view
54	criminal2.pdf	view
64	civil10.pdf	view
74	finance1.pdf	view

Fig. 9. Results of the Added Documents.

Fig. 9 shows the interface where documents can be added, and the documents that were added, which can be viewed by other users aside from the user that added them.

V. RESULTS AND DISCUSSION

From the experiment conducted, Fig. 7 shows the selected legal case documents that the user needs to be classified. Each legal case document shown is named as such, that is, named to reflect the category it falls under so that when the classification task is done, you can see for sure if each file was classified into the accurate category. This was done just for illustration purposes. Any other legal case document, that has any file name can and will be classified into the accurate category by our classification system.

Fig. 8 shows the classified result of the selected legal case documents. Here, each category under which the selected documents fall has been displayed. The user can also view the classified document by clicking the view button. Any other user that logs into the application online can also view these documents that have been classified.

Fig. 9 shows the interface by which the administrator can add new legal case documents that can be viewed later by various users around the world. You can also see the documents that have been added by the administrator.

VI. Conclusion

This paper presents an improved hybrid model for legal case document classification. The predictive ability and accuracy of the model have significantly improved. The result of the improved model is outstanding, having an accuracy of 99.58% when evaluated with never-before-seen legal case documents. The improved classification accuracy is due to the improved model's ability to capture and maintain the context of words in a sentence and use the information to make more accurate predictions. The memory of previous input was maintained, and this enabled the model to better understand the context of the current input. This is particularly useful because the meaning of a word can change based on the words that come before and after it in a sentence. Extensive data pre-processing helped to improve the accuracy of the model by removing irrelevant and noisy data, and by standardizing the data to uniform data for easy processing and analysis. The Principal Component Analysis algorithm improved the computational efficiency of the model, allowing it to train faster and make predictions more quickly.

The model was deployed to the web for easy execution, testing, and assessment.

References

[1] Al-Khurayji R, Sameh A. An Effective Arabic Text Classification Approach Based on Kernel Naïve Bayes Classifier. *International Journal of Artificial Intelligence & Applications*, 2016; 8(6): 1-10.

[2] Burgess CJC. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 1998; 2(1): 955-974.

[3] Elnagar A, Al-Debsi R, Einea O. Arabic Text Classification using Deep Learning Models. *Information Processing and Management*, 2020.

[4] Fusheng W, Han Q, Shi Y, Haozhen Z. Empirical Study of Deep Learning for Text Classification in Legal Document Review. *IEEE International Conference on Big Data (Big Data)*, 2018.

[5] Hongxia L, Ehwerhemuepha L, Rokovski C. A Comparative Study on Deep Learning Models for Text Classification of Unstructured Medical Notes with various levels of Class Imbalance. *BMC Medical Research Methodology*, 2022; 22(1).

[6] Hotho A, Staab S, Stumme G. WordNet Improves Text Document Clustering. *International ACM SIGIR Conference on Research and Development in Information Retrieval*; 2003.

[7] Isa D, Lee LH, Kallimani VP, Rajikuma R. Text Document Preprocessing using the Bayes Formula for Classification Based on the Vector Space Model. *Computer and Information Science Journal*, 2008; 1(4).

[8] Madjid K, Shiva H. *Document Classification Methods*. [Internet]. 2019. Retrieved from: https://www.researchgate.net/publication/335880715_Document_classification_methods.

[9] Mohammed A, Kora R. An Effective Ensemble Deep Learning Framework for Text Classification. *Journal of King Saud University-Computer and Information Sciences*, 2022; 34(10): 8825-8837.

[10] Pinto L, Melgar A. A Classification Model for Portuguese Documents in the Juridical Domain. *11th Iberian Conference on Information Systems and Technologies (CISTI)*; 2016.

[11] Pudaruth S, Soydaudah KMS, Gunputh RP. Categorisation of Supreme Court Cases Using Multiple Horizontal Thesauri. *Intelligent Systems Technologies and Applications, Advances in Intelligent Systems and Computing*, 2016; 385.

[12] Sebastiani F. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 2002; 34(1): 1-47.

[13] Shruti S. *Precision vs Recall*. [Internet]. 2018. Retrieved from https://towardsdatascience.com/precision-vs-recall-386cf9f89488.

[14] Thomas AM, Resmipriya MG. An efficient Text Classification Scheme Using Clustering. *International Conference on Emerging Trends in Engineering, Science, and Technology*, 2016; 24(1): 1220-1225.

[15] Ugwu C, Obasi K. Legal Case Document Classification Application based on an Improved Hybrid Approach. *International Journal of Engineering Research and Technology (IJERT)*, 2015; 4(4): 517-525.

[16] Wei T, Lu Y, Chang H, Zhou Q, Bao X. A semantic approach for text clustering using WordNet and lexical chains: Expert Systems with Applications. *Journal of Expert Systems with Applications,* 2015; 2(42): 2264-2275.