Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Effective Document Classification using Novel Enhanced Long Short Term Memory-based Namib Beetle Optimization Algorithm

T. Elavarasi[1], Dr. R. Nagarajan[2]

[1]Research scholar, Department of computer and information science, Annamalai University, Email: arasithirugnanam@gmail.com
[2]Assistant Professor/Programmer, Department of computer and information science, Annamalai University, Email: rathinanagarajan@gmail.com

The process of classifying documents involves grouping them into predetermined groups according to their content. Automating document processing, increasing efficiency, and enhancing decision-making in a variety of enterprises as well as organizations are all made possible by document categorization. This technique has been widely used for text categorization applications such as sentiment analysis, topic modelling, and spam filtering. Further research is required to develop more complex techniques as well as algorithms that will increase the accuracy as well as efficacy of intelligent document categorization. Therefore, this research paper accomplishes the document classification using the novel intelligent deep learning methodology. Initially, the data is gathered from the online sources that includes a group of documents as well as their respective categories. The collected data is subjected to the pre-processing that is done with the help of tokenization, normalization, removal of header/footer, removal of stop words, and stemming of words. The output from the pre-processing is given to the feature extraction phase for extracting the features using the Chi-square approach. The extracted features enter the final classification phase, which is accomplished using the novel Enhanced Long Short Term Memory (ELSTM), in which the parameter tweaking of LSTM is performed by the nature inspired optimization algorithm called Namib Beetle Optimization (NBO) algorithm. The main objective function behind the entire novel document classification methodology is the maximization of accuracy. Finally, this innovative ELSTM-NBO classifies the output into various categories such as image processing, deep learning, data mining, sports, networks, and machine learning respectively. All things

considered, the innovative ELSTM-NBO technique for document classification represents a practical as well as effective way to manage and organize massive amounts of textual data. The proposed ELSTM-NBO for the document classification model in terms of accuracy is 11.39%, 22.17%, 6.25%, and 3.03% better than SSW-based SVM, DoCA, MHGAT, and HCMBO respectively. Similarly, the proposed ELSTM-NBO for the document classification model in terms of F1 Score is 24.99%, 16.43%, 2.98%, and 2.86% advanced than SSW-based SVM, DoCA, MHGAT, and HCMBO respectively.
Keywords: Document Classification; Enhanced Long Short Term Memory; Namib Beetle Optimization Algorithm; Chi-square approach.

1. Introduction

Enhanced NN architectures as well as optimization approaches are used in document classification employing deep learning-oriented optimization to automatically classify and arrange text documents into predetermined categories or groupings [1] [2]. Owing to their capacity to recognize intricate patterns as well as relationships in textual information, deep learning methods such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Transformer methods such as GPT and BERT have shown impressive success in document classification tasks [3].

Training these NN methods on sizable labelled datasets is the optimization component of deep learning-oriented document classification, which aims to reduce a loss function and boost prediction accuracy [4] [5]. To improve model effectiveness and avoid overfitting, batch normalization, regularization strategies, and hyper parameter tweaking are also used [6] [7]. Deep learning methods can capture both global and local relationships in texts by utilizing the hierarchical descriptions learnt via several layers of NNs, which produces more accurate classification outcomes [8].

Deep learning-oriented optimization for document classification is not without its difficulties, though [9]. All things considered, deep learning-oriented optimization for document classification provides a strong as well as adaptable method for automatically organizing and classifying textual documents, facilitating effective information retrieval and content management across a range of fields [10].

The paper contribution is as below.

- To accomplish the document classification using the novel intelligent deep learning methodology.
- To perform pre-processing with the help of tokenization, normalization, removal of header/footer, removal of stop words, and stemming of words.
- To do the classification using the novel ELSTM, in which the parameter tweaking of LSTM is performed by the nature inspired optimization algorithm called NBO algorithm with the main objective function as the maximization of accuracy.

- To classify the final output into various categories such as image processing, deep learning, data mining, sports, networks, and machine learning respectively by the innovative ELSTM-NBO.

The paper organization is as follows. Section 1 is the introduction regarding document classification model. Section 2 is literature survey. Section 3 is proposed methodology with proposed model, data collection, pre-processing, feature extraction by Chi-square approach, classification by ELSTM, and NBO algorithm. Section 4 is results. Section 5 is conclusion.

1.1 Motivation

The driving force is the growing amount as well as complexity of textual information being produced in a wide range of fields as well as industries. Conventional statistical as well as rule-oriented approaches to document categorization sometimes find it difficult to capture the context, subtleties, and unpredictability found in unstructured text data. In order to overcome these obstacles, deep learning-oriented optimization offers a viable solution that makes hierarchical representation, automated feature learning, and pattern detection in texts possible. Researchers want to create more scalable, precise, and adaptable document classification frameworks that can efficiently as well as accurately classify, organize, and retrieve textual data by utilizing Neural Network (NN) topologies and optimization approaches. To fully realize the feasibility of textual data analysis in a data-driven world, it is imperative to advance research in this area. Additionally, the feasible applications of deep learning-oriented document classification span domains like sentiment analysis, information retrieval, fraud detection, content recommendation, and risk assessment.

2. Related Work

This work used a metric called Sentiment Similarity Weight (SSW) to present a word-embedding-oriented traffic document classification method for emerging risk detection [11]. By taking into account and categorizing the significance as well as polarity of terms in traffic documents, the suggested technique found new hazards. Semantically important keywords were not used by traditional sentiment analysis techniques unless they were present in a sentiment dictionary. In this work, the suggested technique solved the drawback of emotion dictionaries by expanding the restricted utilization range and employing word imputation utilizing an established similarity dictionary. Three tests were used to assess the suggested approach. In the first, model accuracy was assessed by measuring the similarity among terms. Three classifiers for the categorization of emerging risks were compared in the second test.

From the ACM-prepared SANTOS dataset, four meta-data-oriented features—abstract, keyword, title, and generic terms—were selected [12]. Rather than using the more popular count-oriented methods, a semantic-oriented method named BERT was produced to quantitatively express these attributes. Every record that BERT created was a 768-dimensional vector, adding a great deal of computational time complexity. Furthermore, a genetic algorithm was used to optimize the characteristics in the suggested method. In this area, choosing the most relevant features from this large-dimensional space took less time and improved the overall accuracy related to the document classification framework. This was where optimal feature selection played a critical role. SVM and GNB and GNB classifiers were used for classification.

The purpose of this work was to investigate how Arabic DC was affected by stemming approaches, such as ARLStem, Tashaphyne, and Information Science Research Institute (ISRI) [13]. This research employed three classification algorithms: SVM, Naïve Bayesian (NB), and K-Nearest Neighbors (KNN). The results of this study showed that the ARLStem accomplished better than the ISRI stemmers and Tashaphyne.

The Document Classification and Analysis (DoCA) methodology was put into place to streamline and automate various kinds of processes for various file formats, including multimedia files (audio and video), scanned documents (PDFs and images), and office documents (spreadsheets, text, and presentations) [14]. The accuracy of several tasks demonstrated the DoCA's potential as a platform for document analysis as well as categorization.

An effective method for document representation was given employing clustering algorithms to split a document container into many subcontainers and determine the relationships among the subcontainers [15]. The findings demonstrated that the methods surpassed the traditional methods in producing high-quality document representations for document-level categorization associated tasks when compared to basic linear methods.

In order to accurately classify documents, a unique multi-element hypergraph gated attention network was presented [16]. This network was capable of capturing multi-element data as well as word location. To be more precise, a brand-new multi-element hypergraph was initially suggested to represent the sentence structure, word location, and overall content of the document. Lastly, to combine learnt element descriptions into whole document descriptions for classification, a novel block-wise read-out module was created.

The suggested strategy's objective or fitness function was made to reduce the rate of term/feature selection as well as the rate of classification error [17]. The effectiveness related to the proposed method was tested using four text corpora: AG News Corpus, 10 Newsgroups, Reuters-R8, and WOS-5736. The comparative analysis report showed that the recommended technique outperformed the remaining competing strategies. The recommended strategy's effectiveness differed significantly from remaining competing strategies, as indicated by the Nemenyi post-test above the significance threshold of 0.05.

It was suggested to use contextualized document representations of pretrained language methods for ML-DST+ and ML-DST [18]. Developed were a newly constructed weighted loss function for fine tuning and a multi-label classifier on the basis of BERT. In order to enhance multi-label prediction, two label propagation-oriented algorithms, SemLPA+ and SemLPA, were also developed. Seman-space fine tuning was an iterative process that involved fine-tuning the semantic space made up of document representations in order to better describe learnt label correlations. Classifier fine tuning and Semantic-space both employed the high-confidence label predictions that were identified by looking at the prediction score for every category independently. The findings of the experiment demonstrated the superiority of the suggested technique, with the technique's effectiveness consistently surpassing the representative baselines beneath varying label rates.

An explainable transfer-learning-oriented approach to document categorization for multidisciplinary texts was proposed in this work [19]. Utilizing previously published labelled single-discipline papers, a single-discipline classification method was trained initially. Next, in order to solve the lack of labelled interdisciplinary data, the information gained from single-discipline classification was applied to interdisciplinary classification. The suggested model was expanded by including discipline co-occurrence data. Using transdisciplinary data to train the transferred method, the final method was acquired. Furthermore, layer-wise relevance propagation was used to generate keyword-oriented explanations for text classification. Tests conducted on actual NSFC data demonstrated the efficacy related to the suggested approach, which could foster interdisciplinary growth by building a just as well as effective categorization framework for interdisciplinary review methodologies.

In order to address issues with localization and type definition of documents with a known format, the study focused on techniques and approaches for document image analysis [20]. It explained the design ideas that have evolved as performance standards have tightened and input data has grown increasingly complicated. From dealing with scanned images to video stream frames and photos, and from the major general classes of documents related to text structure to the tightest ones on the basis of their visual properties, the approaches described in the article illustrated the scientific route of the school.

2.1 Research Gaps

Large labelled datasets are necessary for training deep learning methods, but effectively labelling them may be costly as well as time-consuming, especially when it comes to domain-specific or specialized document categories. Large computing resources, such as high-performance TPUs or GPUs, are required for training deep learning methods for document categorization. This can result in longer training durations and possible scaling problems. When trained on little amounts of data, deep learning methods are susceptible to overfitting, which reduces the resilience of the method in real-world applications and leads to poor generalization effectiveness on unseen documents. The adaptability as well as transferability of pre-trained deep learning methods across distinct datasets may be limited if extra retraining and fine-tuning are needed to adapt them to particular domain changes or document categorization tasks.

3. Proposed Methodology

3.1 Proposed Model

The proposed document classification model includes various phases such as data collection, pre-processing, feature extraction, and classification. The first step involves gathering the data from web sources, which comprise a collection of documents and their corresponding categories. Pre-processing is applied to the gathered data, and it includes tokenization, normalization, stop word removal, header and footer removal, and word stemming. The preprocessing result is sent into the feature extraction step, which uses the Chi-square method to extract the features. The collected features move on to the last stage of classification, which is completed using the innovative ELSTM. The NBO algorithm, an optimization technique inspired by nature, is used to adjust the LSTM's parameters. Maximizing accuracy is the primary goal of the entire innovative document categorization approach. Ultimately, the output is categorized into many categories by this novel ELSTM-NBO, including image processing, deep learning, data mining, sports, networks, and machine learning. The overall proposed methodology diagram for the document classification model is shown in Figure 1.

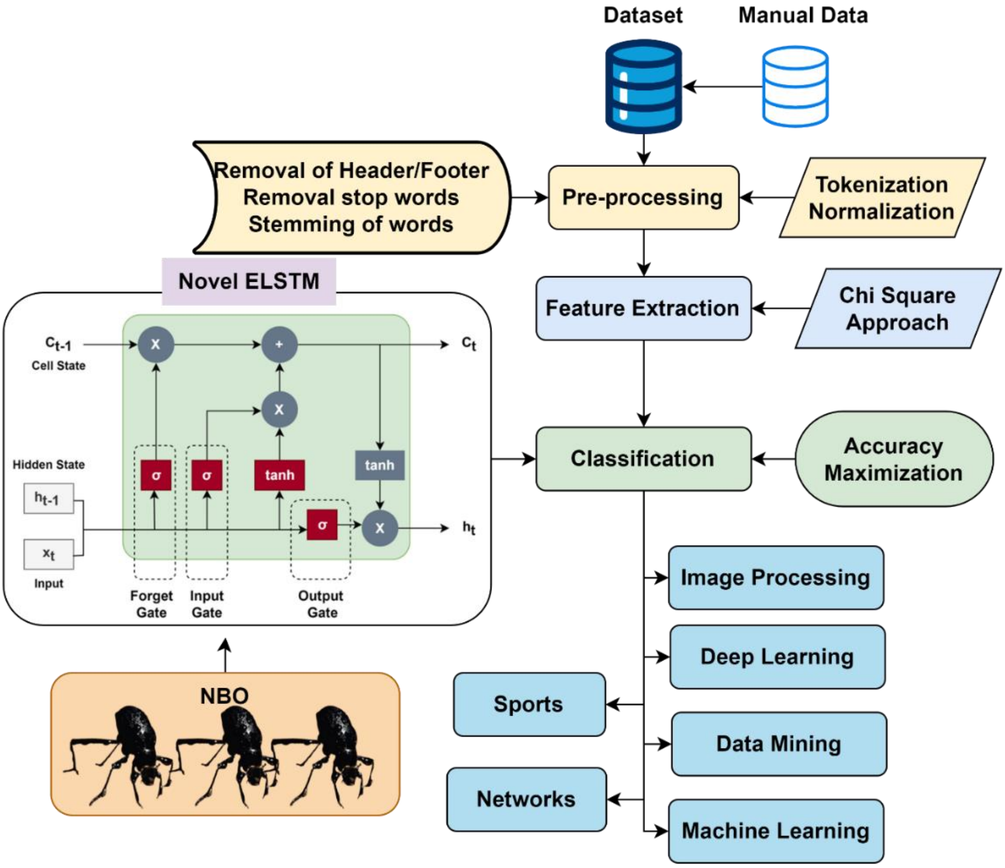


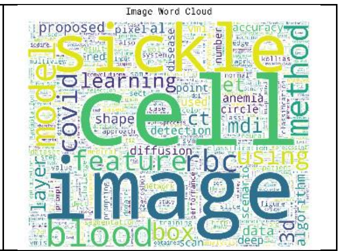
Figure 1. Overall proposed methodology diagram for document classification

3.2 Data collection

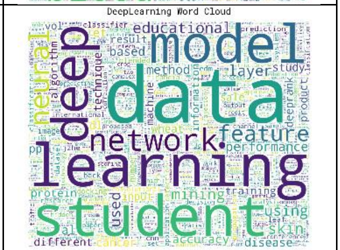
Image processing, deep learning, data mining, sports, networks, and machine learning are the six manual sources that provide the dataset for the suggested published document classification methodology. Every source consists of ten scholarly publications. As seen in Figure 2, these sources can be diagrammatically represented as a word cloud.

Dataset	Symbolic representation

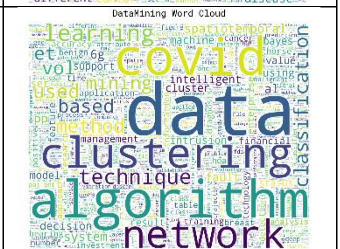
Image processing



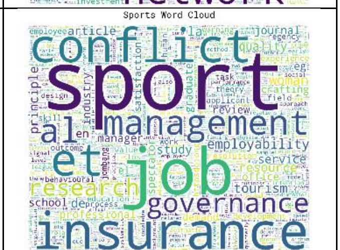
Deep learning



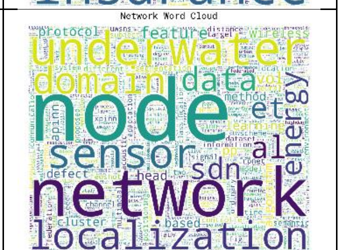
Data mining



Sports



Network



Machine learning

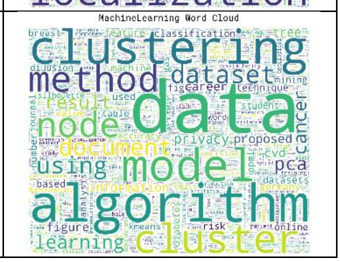


Figure 2. Diagrammatic description of word cloud for the collected data sources

3.3 Pre-processing

The procedure that transforms the documents into the appropriate classification format is known as document preprocessing. Preprocessing techniques improve performance and classification accuracy by reducing the feature space and computing process. The pre-processing of the collected data for the proposed data classification model is accomplished using the tokenization, normalization, removal of header/footer, removal of stop words, and stemming of words. Each of these techniques is briefly described as below.

Tokenization: The process that divided the text into tokens is called tokenization. These "tokens" might be phrases, individual words, or paragraphs. Frequently, punctuation marks like commas, white space, semicolons, periods, and quotations are used to divide words from one another. It describes a crucial stage in the pre-processing of documents, when text is separated into smaller pieces known as tokens. Usually, numerals, words, or punctuation are used as these tokens. It is the procedure of dividing the input text into discrete tokens according to pre-established guidelines. It techniques that are often used involve dividing text according to certain punctuation, delimiters, or spaces.

Normalization: The procedure of transforming a text letter into a canonical form is called normalization. In this procedure, contractions are expanded, punctuation is removed, text is usually converted to lowercase, and words are reduced to their root or base form by stemming or lemmatization.

Removal of header/footer: Through the removal of any extraneous data included in the document's header as well as footer, the primary text associated with the document is extracted. Metadata, copyright information, page numbers, and various unrelated material are frequently included in headers and footers, which can cause problems for text analysis activities. Eliminating headers and footers directs attention towards the document's main text, which facilitates the extraction of valuable data by algorithms. Through the removal of noise as well as unnecessary information from the document, this procedure serves to increase the precision as well as effectiveness of text analysis.

Removal of stop words: Almost every document in every category includes terms like "A," "The," "To," "From," and so on. Eliminating these terms might potentially yield more precise categorization outcomes. It entails eliminating popular terms that, in a particular context, have little to no semantic significance. Articles, conjunctions, prepositions, and numerous often used words that don't significantly advance the text's analysis are examples of these stop words. This procedure can produce text analysis findings that are more efficient and effective by reducing the dimensionality related to the data.

Stemming of words: A few of the terms are derivatives of other ones. It would be preferable in these situations if entire variants of root words are treated as the original root word. It entails stripping words of their affixes, like prefixes and suffixes, to return them to their root or base form. By normalizing words with identical meanings but distinct forms, this approach makes it possible for computers to regard them as one and the similar. Stemming, for instance, would change "running" and "runs" to the more often used root form "run." Stemming can increase efficiency and accuracy by lowering text data's variance and vocabulary size by breaking words down to their stems. Word stemming prior to feature representation can improve classification precision.

3.4 Feature extraction by Chi-square approach

The features of the proposed document classification model are extracted from the pre-processed data employing the Chi-square approach. It is described as the process of evaluating the hypothesis of discrete data that is known via statistics; it determines if two variables are connected or correlated by analyzing their connection. By comparing the observed frequencies of a certain feature appearing with every class versus the predicted frequencies if the class and feature were independent, it assesses the degree of independence among every class labels (categories) and the feature (word). Since they are more likely to be connected to certain classes, features having high chi-square values are thought to be the major pertinent to the classification process. The document classification method's performance and accuracy can be enhanced by the classifier focusing on the most discriminative words by choosing features having important chi-square values.

Utilizing equation (1), the $\boldsymbol{\chi^2}$ value associated with every category $\boldsymbol{\mu}$ as well as term $\boldsymbol{u}$ may be determined.

$$\boldsymbol{\chi^2\left(u_l,\mu_j\right) = \frac{|U_s|\cdot\left[q(u_l,\mu_j)*q(\overline{u_l},\overline{\mu_j})-q(u_l,\overline{\mu_j})*q(\overline{u_l},\mu_j)\right]^2}{q(u_l)*q(\overline{u_l})*q(\mu_j)*q(\overline{\mu_j})}} \tag{1}$$

Additionally, the below equation is used to calculate $\boldsymbol{\chi^2}$:

$$\boldsymbol{\chi^2(u,\mu) = \frac{O*(\alpha\omega-V\beta)^2}{(\alpha+V)*(\beta+\omega)*(\alpha+\beta)*(V+\omega)}} \tag{2}$$

Here, $\boldsymbol{\beta}$ describes the frequency associated with $\boldsymbol{u}$ occurrences without $\boldsymbol{\mu}$, $\boldsymbol{\alpha}$ describes the frequency of $\boldsymbol{u}$ and $\boldsymbol{\mu}$ occurrences, $\boldsymbol{V}$ describes the frequency of $\boldsymbol{\mu}$ without $\boldsymbol{u}$, the number of documents is shown by $\boldsymbol{O}$, and $\boldsymbol{\omega}$ represents the frequency of none among the two events occurring.

3.5 Classification by ELSTM

The classification phase is done by the novel ELSTM for the proposed document classification model. Here, the parameters of LSTM are tweaked by NBO algorithm with the aim of deriving accuracy maximization as the main fitness function, thus referred to be novel ELSTM. One unique variety of Recurrent Neural Network (RNN) represents the LSTM. While conventional RNNs are quite good at processing information with sequential characteristics, they are not very good at capturing extended dependencies in sequential information because they frequently run into problems with exploding and vanishing gradients. The LSTM model incorporates a state structure as well as three gate structures (input gate, forget gate, cell state, and output gate) to enhance upon traditional RNNs. These improvements enable self-recurrent weights to be dynamically adjusted, so addressing problems associated with exploding as well as vanishing gradients and offering both long- and short-term memory functions.

Forget gate: The forget gate examines the input data for the present time step, represented as $\boldsymbol{y_u}$, as well as the output data for the earlier time step, represented as $\boldsymbol{i_{u-1}}$. The gate discards the read data when $\boldsymbol{g_u = 0}$. On the other hand, it keeps the read data when $\boldsymbol{g_u = 1}$. The following represents the formula used to calculate $\boldsymbol{g_u}$.

$$\boldsymbol{g_u = \sigma\left(X_g \cdot [i_{u-1}, y_u] + c_g\right)} \tag{3}$$

In this case, $\boldsymbol{c_g}$ represents the bias coefficient, $\boldsymbol{X_g}$ represents the weight matrix related to the forget gate, and $\boldsymbol{\sigma}$ denotes the sigmoid activation function.

Input gate: The fresh input data that should be stored in the neuron is chosen by this gate. The process begins with the creation of a candidate cell state, $\boldsymbol{\widetilde{D}_u}$, which is next updated by the input gate $\boldsymbol{j_u}$. The cell state is next updated with the novel data. The particular equation is as below.

$$\boldsymbol{\widetilde{D}_u = tanh(X_d[i_{u-1}, y_u] + c_d)} \tag{4}$$

$$\boldsymbol{j_u = \sigma\left(X_j \cdot [i_{u-1}, y_u] + c_j\right)} \tag{5}$$

$$\boldsymbol{D_u = g_u \times D_{u-1} + j_u \times \widetilde{D}_u} \tag{6}$$

The weight matrix for the cell state is denoted by $\boldsymbol{X_d}$ in the formula above, the bias coefficient related to the cell state is shown by $\boldsymbol{c_d}$, the input gate's weight matrix is denoted by $\boldsymbol{X_j}$, and the input gate's bias coefficient is indicated by $\boldsymbol{c_j}$.

Output gate: The output gate uses the cell state to calculate the final output $\boldsymbol{i_u}$. Processing the present input data $\boldsymbol{y_u}$ as well as the prior output data $\boldsymbol{i_{u-1}}$ comes first. To get the final output, $\boldsymbol{i_u}$, it multiplies these values by the cell state that the tanh layer processed. The particular equation is as below:

$$\boldsymbol{p_u = \sigma\left(X_p \cdot [i_{u-1}, y_u] + c_p\right)} \tag{7}$$

$$\boldsymbol{i_u = p_u \times tanh(D_u)} \tag{8}$$

The output gate's weight matrix is denoted by $\boldsymbol{X_p}$ in this formula, and its bias coefficient is denoted by $\boldsymbol{c_p}$.

Because of their gating processes and special memory cells, long-term dependencies in sequences may be captured by LSTM networks, which are advantageous in a number of ways. Compared to more straightforward RNN models, LSTMs demand more computing power, which can result in longer training durations and more resource needs. Moreover, debugging as well as interpreting LSTM designs might be difficult due to their complexity. Another disadvantage is that, even if LSTMs are made to address this problem, vanishing gradients may still happen in some situations. In general, these drawbacks must be taken into account when selecting a model for a given task, even if LSTMs are excellent at capturing long-term relationships. Therefore, the parameters of LSTM are tuned by NBO with the aim of deriving the accuracy maximization as the objective function, thus called to be novel ELSTM. This new ELSTM classifies the final output into classes such as image processing, deep learning, data mining, sports, networks, and machine learning respectively.

3.6 NBO algorithm

The NBO algorithm is selected here for optimizing the LSTM parameters that in turn leads to accuracy maximization as the fitness function. The NBO algorithm has been modelled after the way Namib beetles gather water in the desert. Since the NBO algorithm uses a population-oriented approach to slope optimization, it may be utilized to any topic optimization issue. The chance of pursuing every beetle in quest of moisture, water seeking behavior, and locating an appropriate mound for collecting water are utilized as a method to determine the best solution in the NBO algorithm. According to the suggested approach, every issue solution is seen as a beetle and may be encoded in the $\boldsymbol{E}$ dimensions using Equation (9):

$$\boldsymbol{nb} = [\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}, \cdots, \boldsymbol{y_E}] \tag{9}$$

$\boldsymbol{E}$ is the decision variable for every beetle. As the starting population, some among them can be produced at random as $\boldsymbol{O}$ in the range $\boldsymbol{l}$ and $\boldsymbol{u}$, as in Equation (10):

$$\boldsymbol{pop} = \begin{bmatrix} \boldsymbol{nb_{1,1}} & \boldsymbol{nb_{1,2}} & \cdots & \boldsymbol{nb_{1,E}} \\ \boldsymbol{nb_{2,1}} & \boldsymbol{nb_{2,2}} & \cdots & \boldsymbol{nb_{2,E}} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{nb_{O,1}} & \boldsymbol{nb_{O,2}} & \cdots & \boldsymbol{nb_{O,E}} \end{bmatrix} \tag{10}$$

$\boldsymbol{pop}$ represents the starting beetle population in this equation. However, $\boldsymbol{nb_{j,k}}$ additionally displays the $\boldsymbol{k}$ component linked to the beetle $\boldsymbol{j}$. Equation (11) is used to generate any random solution in the issue space.

$$\boldsymbol{nb_{j,k}} = \boldsymbol{l} + (\boldsymbol{u} - \boldsymbol{l}) \cdot \boldsymbol{Rand}(\boldsymbol{0}, \boldsymbol{1}) \tag{11}$$

Here, the fitness function may be used to assess any problem solution or beetle. The fitness function, which describes a maximization function that displays the hill's height, may be represented with the letter $\boldsymbol{g}$ in this case. The minimization kind of optimization issues may also be solved using this fitness function. Equation (12) may be used to describe the beetle population by applying the fitness function to the data.

$$\boldsymbol{fitness} = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{nb_{1,1}} & \boldsymbol{nb_{1,2}} & \cdots & \boldsymbol{nb_{1,E}}) \\ \boldsymbol{g}(\boldsymbol{nb_{2,1}} & \boldsymbol{nb_{2,2}} & \cdots & \boldsymbol{nb_{2,E}}) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{g}(\boldsymbol{nb_{O,1}} & \boldsymbol{nb_{O,2}} & \cdots & \boldsymbol{nb_{O,E}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{nb_1}) \\ \boldsymbol{g}(\boldsymbol{nb_2}) \\ \vdots \\ \boldsymbol{g}(\boldsymbol{nb_O}) \end{bmatrix} \tag{12}$$

Initially, every beetle or issue-solving solution is put in a random problem space and assessed utilizing the fitness function. Every beetle with a higher fitness function calculation value is better able to gather water and moisture. In light of this, it is possible to conclude that the beetle in question is situated in an ideal location, which may also serve as a draw for remaining beetles seeking water. Equation (13) may be used to determine the potential number of beetles that a given location, home to a beetle like $\boldsymbol{nb_j}$, can support:

$$\boldsymbol{D_j} = \boldsymbol{D_{max}} \cdot \boldsymbol{sin}\left(\frac{g(nb_j) - g_{min}}{g_{max} - g_{min}} \cdot \frac{\pi}{2}\right) \tag{13}$$

$\boldsymbol{D_j}$ describes the maximum count of beetles that can exist in the region in which beetle $\boldsymbol{nb_j}$ is found. The greatest count of beetles that may occupy a single space is known as $\boldsymbol{D_{max}}$. The competence related to the beetle $\boldsymbol{nb_j}$ is represented by $\boldsymbol{g}(\boldsymbol{nb_j})$, whilst the minimum as well as maximum competences associated with the population of beetles are represented by $\boldsymbol{g_{min}}$

and g_{max}, correspondingly. In every iteration, a number of random solutions may be generated in the problem space by computing the value of D_j. These are fresh approaches, and the beetles who are seeking water can choose any of the initial insects and approach them. Here, the two beetles with competence g_{min} and g_{max} have values of D_j of zero and D_{max}, accordingly. Equation (14) is used to determine the quantity of newly formed beetles that are behind the older beetles who are gathering water. It is possible to expand Equation (14) into Equation (15). Every beetle's D_j value indicates how much it may draw attention from and be observed by other insects. D in this connection represents the total number of beetles looking for water:

$$D = \sum_{j=1}^{O} D_j = D_1 + D_2 + \cdots + D_O \quad (14)$$

$$D = \sum_{j=1}^{O} D_{max} \cdot sin\left(\frac{g(nb_j) - g_{min}}{g_{max} - g_{min}} \cdot \frac{\pi}{2}\right) = D_{max} \cdot sin\left(\frac{g(nb_j) - g_{min}}{g_{max} - g_{min}} \cdot \frac{\pi}{2}\right) + \cdots + D_{max} \cdot sin\left(\frac{g(nb_O) - g_{min}}{g_{max} - g_{min}} \cdot \frac{\pi}{2}\right) \quad (15)$$

Every beetle or problem solver must choose locations that have enough moisture for them to locate water. It is reasonable to think that every insect has a charm, or that the dampness surrounding it attracts beetles. With increasing distance, this appeal for absorbing moisture will also diminish. Let's say there exists one beetle-like nb_j in the issue space and another beetle-like nb_k in another region. The count beetle D_j plans to go in the direction of the beetle nb_j. Using Equation (16), the separation among these two beetles can be first determined.

$$e_{jk} = \left\| nb_j - nb_k \right\| = \sqrt{\sum_{l=1}^{E} \left(nb_{j,l} - nb_{k,l}\right)^2} \quad (16)$$

Equation (17), which calculates the degree of attraction of a region to attract beetles, may be used. It is presumed that the beetle nb_k moves into the vicinity of the beetle nb_j:

$$humidity(s) = \rho * humidity_0 \cdot exp\left(-e_{jk}^{n}\right) \quad (17)$$

In this context, the starting humidity in the problem's surroundings as well as space is represented by $humidity_0$, and the wetness that nb_k perceives from the beetle nb_j's regions is represented by $humidity(s)$. The distance among the two beetles or the answer to the issue is represented by e_{jk} in this instance. The coefficient of rise of humidity is expressed with respect to closeness and is derived from Equation (18). As the number of iterations of the suggested algorithm grows, the coefficient of increase of humidity rises and attains the value of ρ. The beetle will sense more wetness as it gets closer to another beetle as the repetition grows, which will cause the p to rise. This increase in iteration will shift the focus of the search from local to global:

$$\rho = q_{max} - q_0 \cdot \left(1 - \frac{Iter}{maxiter}\right) \cdot Rand(0, 1) \quad (18)$$

In this case, q_0 represents the beginning coefficient associated with humidity rise, $Iter$ represents the algorithm's present iteration count, and $maxiter$ represents its maximum iteration count. Additionally, ρ represents the coefficient of increase in humidity that beetles experience in proximity to locations with higher levels of humidity.

As they are closer to the ideal solution or because they have more moisture and can absorb it with a higher coefficient in the region in which the ideal insects are with high moisture absorption, the beetles sense more moisture in the final iteration. The method of attraction between beetles, whereby the moisture sensing coefficient as well as the present location of one beetle are utilized, is depicted with regard to (19).

$$\boldsymbol{nb_k^{New} = nb_k^{Old} + humidity \cdot \left(nb_j - nb_k^{Old}\right) + Levy} \tag{19}$$

The present as well as novel locations associated with a moving beetle in this equation are denoted by the letters $\boldsymbol{nb_k^{Old}}$ and $\boldsymbol{nb_k^{New}}$, correspondingly. $\boldsymbol{nb_j}$ indicates the location of a beetle that draws in remaining beetles. Formulated from Equation (20), the levy represents a random vector for the movement of beetles.

$$\boldsymbol{Levy = \frac{v}{|w|^{\frac{1}{\beta}}} \cdot \left|\frac{\Gamma(1+\beta) \cdot sin\left(\frac{\pi}{2}\beta\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}}\right|^{\frac{1}{\beta}}} \tag{20}$$

$\boldsymbol{v}$ and $\boldsymbol{w}$ represents two independent random vectors that fluctuate within the interval (0, 1), whereas $\boldsymbol{\beta}$ remains constant at 1.5. With their sense of smell, beetles may identify regions that are more damp. The moist patches as well as the beetles' centre of gravity are utilized to represent this behavior. Like in Equation (21), the beetles also look for the area among the point of gravity as well as the best possible solution. Here, it is considered that every beetle possesses a certain quantity of moisture as well as water that it may utilize to its advantage to maximize its search radius from the centre of gravity of these liquids. Here, the zones of optimality and gravity may contain the place where there is a high likelihood of discovering water:

$$\boldsymbol{nb_j^{New} = nb_j^{Old} + Rand \cdot \left(nb^* - \overline{nb}\right) + Levy} \tag{21}$$

The place with the highest moisture content is denoted by $\boldsymbol{nb}^*$, whereas the location associated with the water gravity on the beetle's body corresponds to $\overline{\boldsymbol{nb}}$. The population gravity point in this relationship is represented by $\overline{\boldsymbol{nb}}$, which is computed using Equation (22). The role $\boldsymbol{nb}^*$ may be enhanced and the role $(\boldsymbol{nb})$ may be decreased to further alter Equation (21), allowing for a more local search for suitable responses to be conducted in the final iteration in accordance with Equation (23).

$$\boldsymbol{\overline{nb} = \frac{\sum_{j=1}^{O} nb_j}{O} = \frac{nb_1 + nb_2 + \cdots + nb_O}{O}} \tag{22}$$

$$\boldsymbol{nb_j^{New} = nb_j^{Old} + Rand \cdot \left(q \cdot nb^* - (1-q)\overline{nb}\right) + Levy} \tag{23}$$

Algorithm 1 displays the pseudo-code of the NBO method for resolving optimization issues.

Algorithm 1: NBO

Start

Setting fundamental parameters like $\boldsymbol{o}$ or $\boldsymbol{o\,pop}$, $\boldsymbol{max\ it}$, $\boldsymbol{dim}$ or $\boldsymbol{E}$, $\boldsymbol{humidity_0}$, etc. [LSTM parameters for the proposed document classification model]

$$nb = [y_1, y_2, y_3, \cdots, y_E]$$

Population initialization of namib beetle having random locations:

For $j = 1$ to o do

For $j = 1$ to E do

$$nb_{j,k} = l + (u - l) \cdot Rand(0, 1)$$

End for

End for

Initial population evaluation by cost function

For $j = 1$ to o do

$$fitness(nb_j) = cost\ function\left(fitness(nb_j)\right)$$

End for

$$fitness = \begin{bmatrix} g(nb_{1,1} & nb_{1,2} & \cdots & nb_{1,E}) \\ g(nb_{2,1} & nb_{2,2} & \cdots & nb_{2,E}) \\ \vdots & \vdots & \vdots & \vdots \\ g(nb_{O,1} & nb_{O,2} & \cdots & nb_{O,E}) \end{bmatrix} = \begin{bmatrix} g(nb_1) \\ g(nb_2) \\ \vdots \\ g(nb_O) \end{bmatrix}$$

While $(it \leq max\ it)$ do

$g_{max} = max(fitness)$, $g_{min} = min(fitness)$

Initial coefficient related to humidity rise:

$$\rho = q_{max} - q_0 \cdot \left(1 - \frac{Iter}{maxiter}\right) \cdot Rand(0, 1)$$

Merits associated with every area to gather water:

For $j = 1$ to o do

$$D_j = D_{max} \cdot sin\left(\frac{g(nb_j) - g_{min}}{g_{max} - g_{min}} \cdot \frac{\pi}{2}\right)$$

End for

For $j = 1$ to o do

For $k = 1$ to D_j do

Search around nb_j

End for

End for

For $j = 1$ to o do (Movement to wet areas):

For $\boldsymbol{k = 1}$ to $\boldsymbol{o}$ do

$$e_{jk} = \|nb_j - nb_k\| = \sqrt{\sum_{l=1}^{E}\left(nb_{j,l} - nb_{k,l}\right)^2}$$

$$humidity(s) = \rho * humidity_0 \cdot exp\left(-e_{jk}^{n}\right)$$

$$nb_k^{New} = nb_k^{Old} + humidity \cdot \left(nb_j - nb_k^{Old}\right) + Levy$$

$$Levy = \frac{v}{|w|^{\frac{1}{\beta}}} \cdot \left|\frac{\Gamma(1+\beta) \cdot sin\left(\frac{\pi}{2}\beta\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{\frac{\beta-1}{2}}}\right|^{\frac{1}{\beta}}$$

End for

End for

For $\boldsymbol{j = 1}$ to $\boldsymbol{o}$ do (Computation of population mass and movement to the wet mass)

$$nb_j^{New} = nb_j^{Old} + Rand \cdot \left(nb^{*} - \overline{nb}\right) + Levy$$

End for

$$it = it + 1$$

End while

Return the optimal solution from $\boldsymbol{pop}$ [maximized accuracy of the proposed document classification model]

Stop

4. Results and Analysis

4.1 Experimental Setup

The proposed ELSTM-NBO algorithm for the document classification model was implemented in MATLAB and the findings were analyzed. The population size was placed to be 10 and the iteration count was considered to be 100. Establishing a document classification experiment requires a number of crucial measures to guarantee the accuracy as well as consistency of the findings. Compare the suggested approach with other approaches, like SSW-based SVM [11], DoCA [14], MHGAT [16], and HCMBO [17], and evaluate the methodology utilizing relevant metrics, like accuracy, F1 Score, sensitivity, precision, and specificity.

4.2 Accuracy analysis

The accuracy analysis for document categorization techniques at various iterations (20, 40, 60, 80, and 100) is shown in detail in the Figure 3. A method's accuracy at a given iteration is displayed in every column, while the method itself is represented by a row. The accuracy of the SSW-based SVM consistently increases across the iterations, demonstrating its efficacy in document categorization tasks. DoCA exhibits consistent performance and a progressive increase in accuracy over time, however not as much as some other techniques. HCMBO shows a significant increase in accuracy, proving its effectiveness in document categorization. The suggested ELSTM-NBO accomplishes better than the others, exhibiting enhanced accuracy improvement throughout every iteration. The suggested ELSTM-NBO technique stands out as the major accurate as well as promising solution for document classification tasks. The proposed ELSTM-NBO for the document classification model in terms of accuracy is 11.39%, 22.17%, 6.25%, and 3.03% better than SSW-based SVM, DoCA, MHGAT, and HCMBO respectively. The findings demonstrate the efficacy related to every technique in properly classifying documents.

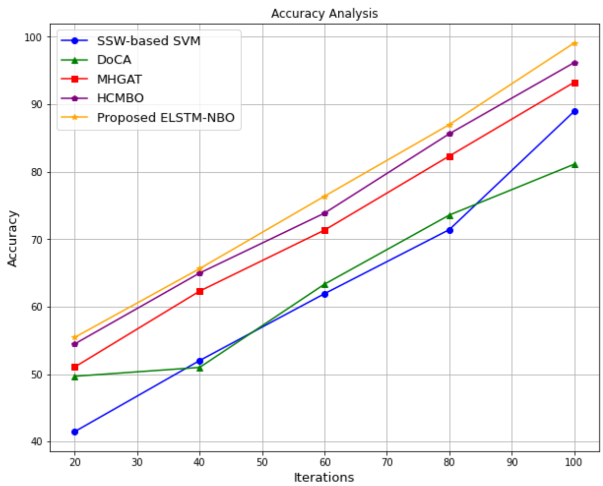


Figure 3. Accuracy analysis

4.3 F1 Score analysis

The F1 scores of many document categorization models across a number of iterations are shown in the Figure 4. A model's accuracy as well as recall are combined into a single number, or F1 score, which strikes a balance among the two measures. These document categorization models' F1 score analysis emphasizes how crucial iteration is to enhancing model performance. Higher F1 scores are often attained by models that have undergone more iterations of training, which indicates improved accuracy as well as recall trade-offs. When compared to the remaining models, the suggested ELSTM-NBO model has the greatest F1 score, indicating its superiority in document categorization. The proposed ELSTM-NBO for the document classification model in terms of F1 Score is 24.99%, 16.43%, 2.98%, and 2.86% advanced than SSW-based SVM, DoCA, MHGAT, and HCMBO respectively.

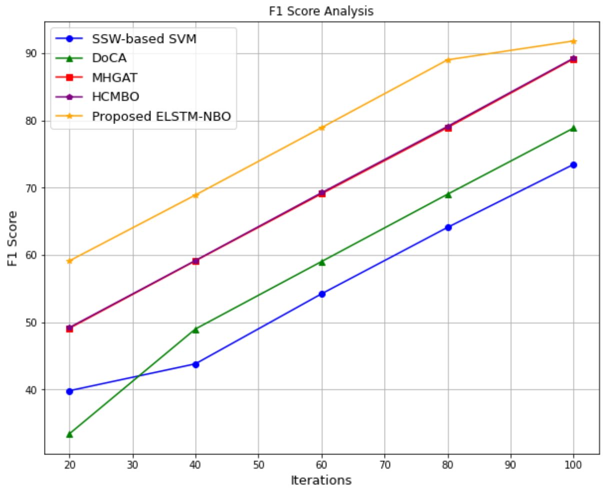


Figure 4. F1 Score analysis

4.4 Sensitivity analysis

The sensitivity analysis of many document categorization models via multiple iterations is shown in Figure 5. Greater sensitivity denotes an improved ability of the model to accurately detect positive cases. As the count of iterations grows, the sensitivity analysis of these document classification methods demonstrates the algorithms' accuracy in identifying affirmative situations. When compared to the remaining models, the suggested ELSTM-NBO method has the best sensitivity, demonstrating its effectiveness in properly recognizing positive situations. The proposed ELSTM-NBO for the document classification model with respect to sensitivity is 15.29%, 15.26%, 2.21%, and 2.49% higher than SSW-based SVM, DoCA, MHGAT, and HCMBO respectively.

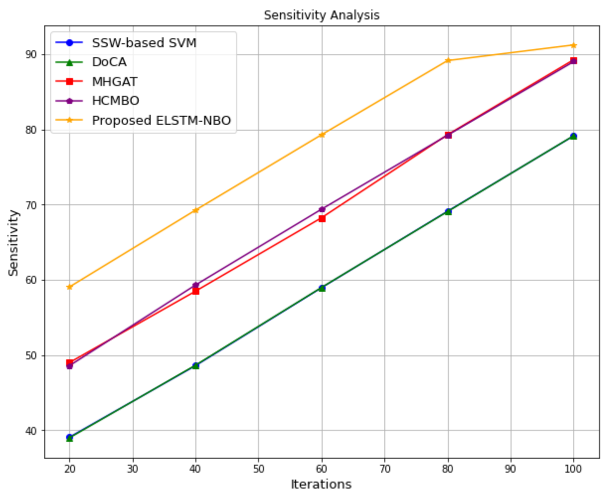


Figure 5. Sensitivity analysis

4.5 Precision analysis

The precision analysis of many document categorization models across multiple iterations is presented in Table 1. As training goes on, the precision analysis of these document classification methods demonstrates the models' great precision in properly identifying positive situations. When compared to the other models, the suggested ELSTM-NBO model exhibits the best precision, demonstrating its efficacy in correctly recognizing positive situations. The proposed ELSTM-NBO for the document classification model in terms of precision is 14.16%, 12.53%, 6.31%, and 2.83% greater than SSW-based SVM, DoCA, MHGAT, and HCMBO respectively.

Table 1 Precision analysis

Methods	Iterations				
	20	40	60	80	100
SSW-based SVM [11]	49.17	59.23	65.29	78.21	85.18
DoCA [14]	48.13	50.32	60.09	79.52	86.41
MHGAT [16]	59.19	68.42	77.72	86.89	91.47
HCMBO [17]	50.16	60.56	79.90	88.31	94.56
Proposed ELSTM-NBO	60.37	69.54	80.53	89.40	97.24

4.6 Specificity analysis

The specificity study of many document categorization models across multiple iterations is shown in Table 2. Higher specificity scores are often displayed by methods that have undergone more iterations of training, suggesting that they are more adept at preventing false negatives. When compared to the other models, the suggested ELSTM-NBO model has the best specificity, demonstrating its efficacy in correctly recognizing negative situations. The proposed ELSTM-NBO for the document classification model with respect to specificity is 20.63%, 16.63%, 5.31%, and 2.94% superior to SSW-based SVM, DoCA, MHGAT, and HCMBO respectively.

Table 2 Specificity analysis

Methods	Iterations				
	20	40	60	80	100
SSW-based SVM [11]	49.70	50.19	69.21	78.38	80.82
DoCA [14]	40.23	51.44	69.28	75.20	83.59
MHGAT [16]	55.39	64.97	76.40	87.48	92.57
HCMBO [17]	53.58	63.02	71.49	84.60	94.71
Proposed ELSTM-NBO	58.04	67.91	79.83	88.50	97.49

5. Conclusion

This research article used a revolutionary intelligent deep learning technology to perform document classification. The first step involved gathering the data from web sources, which comprise a collection of documents and their corresponding categories. Pre-processing was applied to the gathered data, and it included tokenization, normalization, stop word removal, header and footer removal, and word stemming. The pre-processing result was sent to the feature extraction stage, which used the Chi-square method to extract the features. The collected features move on to the last stage of classification, which was completed using the innovative ELSTM. The NBO algorithm, an optimization technique inspired by nature, was used to adjust the LSTM's parameters. Maximizing accuracy was the primary goal of the entire innovative document classification approach. Ultimately, the output was classified into many categories by this novel ELSTM-NBO, including image processing, deep learning, data mining, sports, networks, and machine learning. All things considered, managing and organizing enormous volumes of textual data was made feasible and efficient by the novel ELSTM-NBO approach for document classification. The proposed ELSTM-NBO for the document classification model in terms of accuracy was 11.39%, 22.17%, 6.25%, and 3.03% better than SSW-based SVM, DoCA, MHGAT, and HCMBO respectively. Similarly, the proposed ELSTM-NBO for the document classification model in terms of F1 Score was 24.99%, 16.43%, 2.98%, and 2.86% advanced than SSW-based SVM, DoCA, MHGAT, and HCMBO respectively.

References

1. F. Liu, L. Zheng, and J. Zheng, "HieNN-DWE: A hierarchical neural network with dynamic word embeddings for document level sentiment classification", Neurocomputing, vol. 403, pp. 21–32, Aug. 2020.
2. S. A. Devi and S. Siva, "A hybrid document features extraction with clustering based classification framework on large document sets", Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 7, pp. 1–11, 2020.
3. M. N. Asim, M. U. G. Khan, M. I. Malik, A. Dengel, and S. Ahmed, "A robust hybrid approach for textual document classification", Proc. Int. Conf. Document Anal. Recognit. (ICDAR), Sep. 2019, pp. 1390–1396.
4. J. Abreu, L. Fred, D. Macêdo, and C. Zanchettin, "Hierarchical attentional hybrid neural networks for document classification", in Proc. Int. Conf. Artif. Neural Netw. Munich, Germany: Springer, Sep. 2019, pp. 396–402.
5. A. Díaz-Manríquez, A. B. Ríos-Alvarado, J. H. Barrón-Zambran, T. Y. Guerrero-Melendez, and J. C. Elizondo-Leal, "An automatic document classifier system based on genetic algorithm and taxonomy", IEEE Access, vol. 6, pp. 21552-21559, 2018.
6. Sindhu Rashmi. H. R, Prof. Anisha. B. S, Dr. Ramakanth Kumar. P "Smart Document Analysis Using AI-ML", (IJIRCST), vol. 7, no. 3, 2019.
7. Ankit Basarkar "Document classification using Machine Learning", Springer International Conference 5-25-2017, vol. 531, 2018.
8. Anitha Jebamani Soundararaj, Godfrey Winster Sathianesan,"Task offloading scheme in Mobile Augmented Reality using hybrid Monte Carlo tree search (HMCTS)", Alexandria Engineering Journal, vol. 108, pp. 611-625, 2024.
9. Manoharan, J. S. et al. "A hybrid approach to accelerate the classification accuracy of cervical cancer data with class imbalance problems", International Journal of data mining, vol. 25, no. (3/4), pp. 234-259, 2021.
10. Rekha, V., Samuel Manoharan, J., Hemalatha, R. & Saravanan D. "Deep Learning Models for Muliple Face Mask Detection under a complex big data environment", Procedia Computer Science, vol. 215, pp. 706-712, 2022.

11. M. -J. Kim, J. -S. Kang and K. Chung, "Word-Embedding-Based Traffic Document Classification Model for Detecting Emerging Risks Using Sentiment Similarity Weight," in IEEE Access, vol. 8, pp. 183983-183994, 2020.
12. G. Mustafa et al., "Optimizing Document Classification: Unleashing the Power of Genetic Algorithms," in IEEE Access, vol. 11, pp. 83136-83149, 2023.
13. Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz and A. Dahou, "A Study of the Effects of Stemming Strategies on Arabic Document Classification," in IEEE Access, vol. 7, pp. 32664-32671, 2019.
14. S. Eken, H. Menhour and K. Köksal, "DoCA: A Content-Based Automatic Classification System Over Digital Documents," in IEEE Access, vol. 7, pp. 97996-98004, 2019.
15. S. Guo and N. Yao, "Document Vector Extension for Documents Classification," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 8, pp. 3062-3074, 1 Aug. 2021.
16. Yilun Jin, Wei Yin, Haoseng Wang, Fang He, "Capturing word positions does help: A multi-element hypergraph gated attention network for document classification", Expert Systems with Applications, vol. 251, October 2024.
17. Karpagalingam Thirumoorthy, Jerold John Britto, "A two-stage feature selection approach using hybrid elitist self-adaptive cat and mouse based optimization algorithm for document classification", Expert Systems with Applications, vol. 254, November 2024.
18. Zhewei Xu & Mizuho Iwaihara, "Self-training involving semantic-space finetuning for semi-supervised multi-label document classification", International Journal on Digital Libraries, vol. 25, pp. 25-39, May 2023.
19. Xiaoming Huang, Peihu Zhu, Yuwen Chen & Jian Ma, "A transfer learning approach to interdisciplinary document classification with keyword-based explanation", Scientometrics, vol. 128, pp. 6449-6469, November 2023.
20. N. S. Skoryukina, D. V. Tropin, J. A. Shemiakina & V. V. Arlazarov, "Document Localization and Classification As Stages of a Document Recognition System", Pattern Recognition and Image Analysis, vol. 33, pp. 699-716, 2023.