

# A Survey on Book Genre Classification System using Machine Learning

Dr. M. Narendra<sup>1</sup>, Dr. K.M. Rayudu<sup>2</sup>, Dr. T. Sivaratna Sai<sup>3</sup> Dr. Kadiyam Rajshekar<sup>4</sup>  
Venkateswarlu Lingala<sup>5</sup>

<sup>1, 2, 3</sup> Department of CSE, <sup>4</sup>Department of BS&H, <sup>5</sup>Department of EEE QIS College of Engineering and Technology, Ongole, AP, India, [qispublications@qiscet.edu.in](mailto:qispublications@qiscet.edu.in)

## Article Info

**Page Number:** 147-160

**Publication Issue:**

**Vol 69 No. 1 (2020)**

**Article Received:** 10 August 2020

**Revised:** 15 September 2020

**Accepted:** 20 October 2020

**Publication:** 25 November 2020

**Abstract.** The amount of complicated texts and documents that need a deeper understanding of machine learning techniques has expanded rapidly in recent decades. Several machine learning approaches have shown exceptional results in NLP. Complex models and non-linear data correlations help these learning systems function well. It is difficult for academics to discover appropriate text classification structures, designs, and procedures. Manually reading and classifying a book's category would be tedious. It takes a long time for a language beginner to absorb the complete text in order to discern its genre. Natural Language Processing (NLP), which is commonly used nowadays, can help overcome this issue. It includes text classification and summarization. The book synopsis is input to the Machine Learning algorithms, which output the genre. This review includes text feature extraction, dimensionality reduction, algorithms, and evaluation. Finally, the limitations of each strategy are assessed, as well as their practicality.

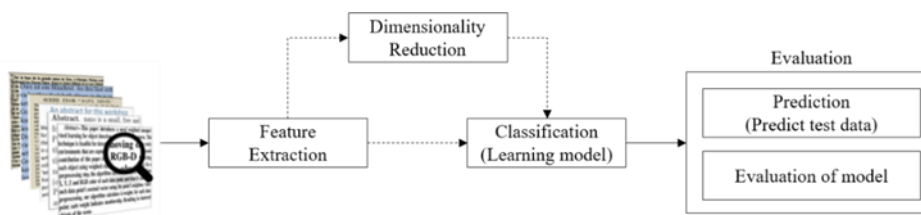
**Keywords:** Genre Classification, K-Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machine (SVM), TF-IDF Vectorizer, Natural Language Processing.

---

## 1 Introduction

Over the last few years, text classification problems have been comprehensively researched and addressed in numerous real-world applications [1–10]. Many researchers are now interested in developing applications that make use of text classification methods, especially in light of recent breakthroughs in Natural Language Processing (NLP) and text mining (TM). In most text and document classification frameworks, the four steps of extracting the features, dimension reduction, classifier selection and evaluation can be broken down. Architecture and implementation of text classification systems are described in this article in terms of a channel [11] [12]. The aim of the researcher is to teach computers how to better handle large amounts of data. Data pre-processing, text cleaning, feature selection, training model, classifier assignment and output evaluation are common stages in text classification. The classification of a book's genre has become the most common problem in libraries today. It is difficult for libraries and readers alike to categorize many books because they aren't labelled by genre. Genre predictions can be made based on the titles and summaries of books in classifying them. Developing a model that can identify a book's genre from the data provided is the goal. A book's classification will be based on its metadata or text [13][14]. According to the authors of this document, there are a number of different ways to categorize and classify texts that have been discussed elsewhere.

1. The document level is where the algorithm gets categories of a full document.
2. In the paragraph level, the algorithm finds the categories that are important for a single paragraph (a section of a document).
3. In the sentence level, you can find out which categories are important for one sentence (a piece of it). It is a paragraph.
4. One such level is called the sub-sentence level. In this level, the algorithm gets right kind of sub expressions contained in the sentences.



**Fig. 1.** Overview of Text Classification

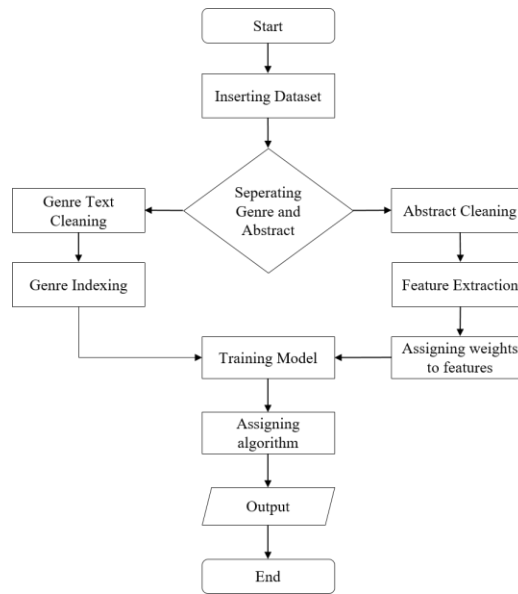
The text is usually staged for multiple phases as depicted in Fig. 1. Feature extraction methods are initially used to the cleaned text, followed by dimensionality reduction techniques to decrease the bulk of the text to smaller dimensions. After that, the classifier is applied to the pre-processed text, trained, and assessed for its performance on test data predictions. Here the overall work has been separated in different sections. In section II, it represents in detail survey of related work done by various researchers from last decade. Various machine learning approaches has evaluated by using certain predefined supervised algorithms and they have also listed out in the survey. Section III represents the conclusion and future enhancements.

## 2 Related Work Done

### 2.1 Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Logistic Regression (LR)

Panchal et al. [1] investigated some of the flaws in text classification algorithms and experimented with text classification on two datasets. The CMU book summary dataset was the first to be utilised. This dataset comprises plot summaries for 16,559 novels, as well as Freebase aligned metadata such as author, title, and genre, collected from Wikipedia. A second dataset was produced utilising information gathered from a variety of sources[31]. The book title, language, author name, genre, and abstract are all included in this dataset. The data collection contains around 200 novels that have been translated from Gujarati and Hindi into English. Data loading, reading, dividing, counting, and labializing were all part of the pre-processing. After that, the data was cleaned by removing any extraneous characters or stop words that were no longer required for classification. Later on, the TF-IDF[15] vectorizer was used to extract features from abstracts and apply weights to feature values using the TF-IDF vectorizer. Finally, using a variety of classifiers such as SVM, KNN, and LR, the model attempted to output the book's genre [16]. Fig. 2 depicts all of the preceding

phases.



**Fig. 2.** Flow Diagram

## 2.2 Using Adaboost Classifier

For book genre categorization, Shikha Gupta et al. [2] suggested the Adaboost Classifier approach. Each text is presumed to have a one-to-one relationship with the genre. To put it another way, a text cannot belong to more than one genre. Beautiful Soup, a Python module, was used to gather unstructured labelled data from the [30] website, totaling roughly 3600 books. Only the content lies between "Start of This Project Gutenberg E-book" and "End of This Project Gutenberg Ebook" is taken out since the material is unstructured. The tokens are then extracted and operations run on them. Only tokens that aren't included in the respective buzz words that are to be taken care. Moreover, stop words (such as 'a' and 'the') are eliminated from most manuscripts as they are not much significant for the respective category[32]. The tokens are then converted using Wordnet, and TF-IDF values are computed using the formula below for additional pre-processing [17][18].

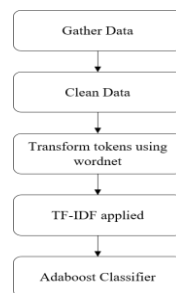
$TF - IDF = Term\ Frequency \times Document\ Frequency$

$$(2) \quad TF - IDF = t f_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

(1)

Where  $N =$  total number of document  $t f_{x,y} =$  total number of occurrences of  $x$  in  $y$  and  $df_x =$  total number of documents containing  $x$ .

The feature matrix is then generated once the data has been normalised. The sparse structure of the feature matrix created affects the training model's performance. To prevent this drawback, Principle Component Analysis is used. The feature matrix is then split into two sections: the training set and the test set. To learn from the training set, the Decision Tree Classifier model is utilised. Furthermore, Adaboost Classifier is utilised to increase Decision Tree accuracy by minimising bias and variance[33]. After the model being trained, it then uses the test set to classify the genre[19][20]. All the stages for text classification has been suggested in the Fig. 3.



**Fig. 3.** Steps executed for Text Classification

### 2.3 Using Character Networks

Rahul et al. [3] demonstrated a Character Network-based Genre Classification. The dataset was originally sourced from Fanfiction.net. N novels with a total length of 5,00,000 words or more were chosen at random. A tokenizer is used to break down a book into phrases, sentences, paragraphs, and individual words. Using a language model that reduces each word to its simplest structure, the resultant document is lemmatized, removing all inflections and derivative forms. This step simplifies the procedure for subsequent steps. The text is run through a part of speech (POS) tagger, which allocates a word to one of the parts of speech. Following the recognition of the characters by their parts of speech, various occurrences of the same character were linked together to ensure that if the character is referred to by several names, they are all linked together[21][22]. The character interaction graph as depicted in Fig. 4. can now be constructed utilising the data that was established once the characters and their interactions were formed. Characters and edges were used to represent the graph's vertices, indicating the presence of interaction between the characters.

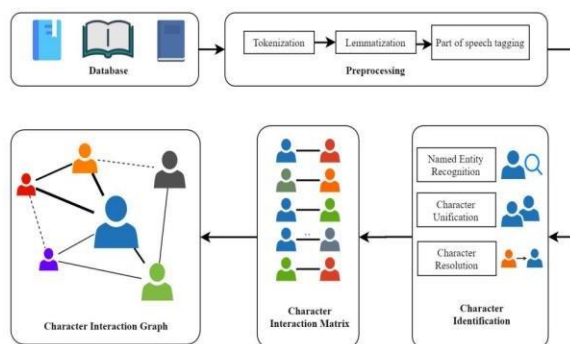


Fig. 4. Generation of Character Interaction Graphs

The interaction score is used to weigh the edges. As illustrated in Fig.5, the difference in the summation of eigenvalues of the Laplacian matrix is used to compare the graphs.

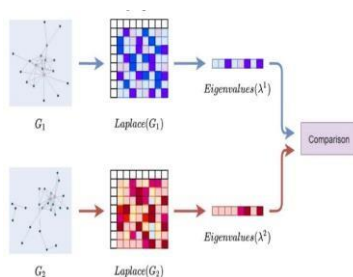


Fig. 5. Comparison of Character Graphs

From the whole dataset,  $k$  closest graphs are chosen for each book. For each book in the dataset, the procedure is repeated, yielding a  $(nsamples \times k)$  matrix that classifiers use as input data. On this dataset, several classifiers such as SVM, Gaussian Process Classifier, Multi-Layer Perceptron, Random Forest, Adaboost, Gradient Boosting, and Gaussian Nave Bayes were used, with each classifier attempting to categorise each sample based on its nearest samples[23][24].

#### 2.4 Using Word2vec algorithm and several machine learning models – CNN, RNN, GRU and LSTM

Eran Ozsarfati et al. [4] proposed a method for classifying a book's genre into up to 32 separate categories. This approach uses a dataset with 207575 data samples, each of which falls to one of the 32 genres. As depicted in Fig.6 initially the data was segmented and normalised to create a customised lexicon out of the unique phrases. Numbers and punctuation were removed, and the inputs were changed to lowercase.

The NLTK stop words collection was then used to eliminate English stop words from the data. The collected data was divided into terms, and stemming, a method that converts derivative terms to their bases, was used to it, lowering the vocabulary size without compromising content. Vectorized representations of words are called word embeddings[34]. The word2vec technique was applied, resulting in a 300-dimensional representation of words. On the entries (words within titles) matching vector form of words, several machine learning models such as Gated Recurrent Unit (GRU), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) were examined. Each approach was adjusted precisely to acquire optimal parameters with no changes to the dataset. Deep-learning techniques, particularly LSTM[25], tend to perform better because of their capacity to hold memory across long-term constraints[26].

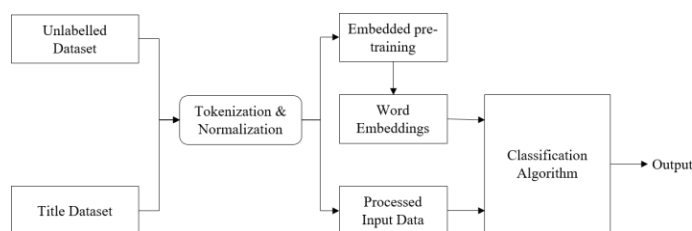
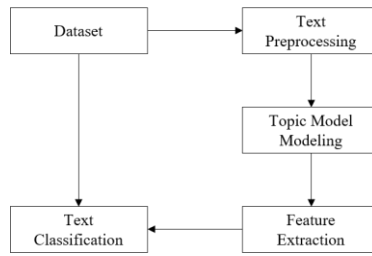


Fig. 6. Data preprocessing and Classification process pipeline

## 2.5 Using Latent Dirichlet Allocation (LDA) and Softmax Regression Technique

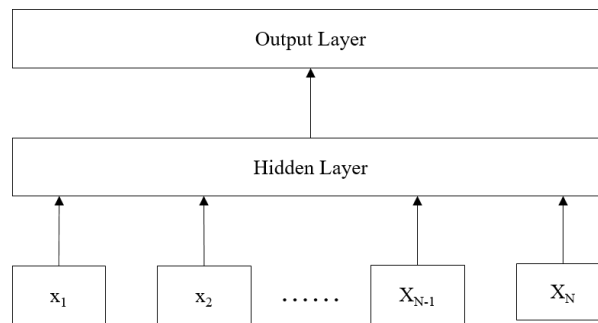
Latent Dirichlet Allocation (LDA) technique was used by Zhenzong Li et al. [5] to create a text classification model. The data is initially gathered from 20 news group data packets, which are frequently employed in categorization algorithms. Three different types of 20 news group data packets were chosen by the writers. The data is then pre-processed, and the LDA model is used to obtain the topic distribution of the training collection of news data. Topic distribution is done using a topic model to minimise the text dimension, which is excessively high, and to get features. In accordance with the news text categorization, a new text must be evaluated and placed in one of several categories. In data mining, this type of task is known as a multiclass classification problem. As a result, the authors proposed a novel Softmax Regression based multi-class text categorization model. The Softmax Regression technique uses the train\_X of news text and the class label train\_Y as input values. It is a Regression Logistic extension that is ideal for multi-classification issues, while Logistic Regression is a traditional two-classification technique that cannot be used to multi-classification situations. Multinomial Logistic Regression is another name for Softmax Regression. The predicted category results are generated using the Softmax Regression classifier[27] once the training and modelling has been completed. The architecture proposed by the authors illustrated in Fig. 7.



**Fig. 7.** Classification process pipeline

## 2.6 Using fastText approach

Tengjun Yao et al. [6] showed a fast Text-based text classification model as illustrated in Fig. 8 with data pre-processing, extraction of features, training, and assessment modules. All training samples were tokenized previously as part of data pre-processing because the sample utilized in this study is a Chinese source. To increase the corpus' quality, the useless garbled letters, phrases, and stop words are deleted once the word tokenization is done. Unlike standard bag-of-words model, the fast Text method's input layer takes the sentence's n-gram feature in addition to the word forms for each word in the sentence as an additional feature to input. Fast Text in addition with n-gram features can acquire word order information in a sentence to a level, generating a more exact sentence representation, which is challenging to do with traditional bag-of-words models. It creates field characteristics based on the text corpus's attributes. While in hidden layer, fast Text aggregates the word form and n-gram characteristics received from the input layer and fastText with hierarchical softmax is then used to find the tag of the input data in the output layer. The use of H- Softmax considerably reduces training time. The hyperparameters involved are learning rate, epoch, window size, bucket, loss, and dim[28].



**Fig. 8.** The model architecture of fastText

## 2.7 Using Mahalanobis distance based KNN classifier

The Mahalanobis distance based KNN classifier was proposed by Suli Zhang et al. [7]. As illustrated in Fig. 9, initially the documents go through a pre-processing stage first. Initially, all stop words in training samples were filtered out. Text documents in their original form cannot be processed directly by popular classifiers and learning algorithms. As a result, feature vectors are used to represent documents during a pre-processing stage. The TF-IDF technique calculates a word's weight in a document and

generates a weight table using term and document frequency. The term frequency indicates how many times the term  $e$  appears in document  $r$ . The number of documents that contain the term  $e$  is known as the document frequency  $DF(e)$ . The formula below is used to compute the inverse document frequency[29] of word  $e$ :

$$\dots\dots\dots(3) \quad IDF(e) = \log \left[ \frac{D}{DF(e)} \right]$$

denotes the total number of documents, and  $IDF(e)$  represents the

discretion degree of term  $e$  over the whole document in the formula above. The formula below is used to compute the word  $e$ 's weight in document  $r$ :

$$\dots\dots\dots(4) \quad W(e, r) = TF(e, r) \times IDF(e)$$

The feature vectors acquired in the preceding phase are then used to train the classifier. Finally, the model is used to predict the category of a text vector.

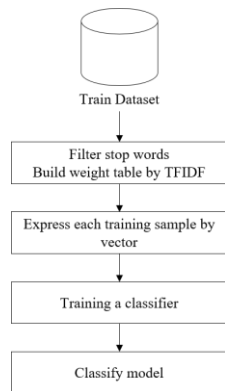


Fig. 9. Classification process

## 2.8 Using novel feature weight method – Gini Index

Wenqian SHANG [8] introduced the TF-Gini feature weight method, which considerably improves classification performance. It's a promising algorithm for weighing text features. There is no logarithm computation in this algorithm. Other text feature weight techniques have a higher computational cost. A non-purity split approach is the Gini index. Three classifiers were employed to assess the novel feature weight method Gini Index: SVM (Support Vector Machine), kNN, and fkNN. The kNN method is used to find  $k$  documents in training sets that have the highest degree of similarity (cosine similarity). It evaluates the test document's candidate classes based on which classes these neighbours belong to. The neighbour document's class weight is determined by the degree of similarity between it and the test document. To improve the kNN algorithm, it used fuzzy theory as well. Information Gain, Odds Ratio, Mutual Information, Expected Cross Entropy, the Weight of Evidence of Text, and CHI were used to compare with the revised Gini index technique for each classifier in the text pre-processing phase[30].



## 2.9. Using Vector Space Model

Shaohui Liu et al. [9] proposed a multi-hierarchy text categorization strategy. It is based on Vector Space Model. The content of a document is codified as a dot in multi-dimensional space and represented by a vector in this approach. Then, the associated classes of the provided vector are determined by computing and comparing the distances between the vectors. The TF-IDF vector format is the most often used document representation in VSM, in which term weight is calculated primarily using term frequency and inverse document frequency. To address TF-IDF's shortcoming, this work examines the TF.IDF.IG method and changes the term weight calculation algorithm to make it more logical. A VSM-based multi-hierarchy text classification system is also presented. All classes are organised as a tree based on hierarchical relationships, and all training records in a class are integrated into a single class document in this manner. Comparing the class documents connected to the same node of the same layer is all that is required to build the class models. While classifying texts, checking is performed hierarchically until a relevant subclass is observed.

## 2.10 Using Word Embeddings

In [10], Roger Alan Stein et al. suggested a hierarchical text categorization system based on word embeddings. With publicly accessible data, it used classification models with well-known ML techniques such as fastText, SVM, XGBoost, Keras' CNN, as well as well-known word embeddings generating techniques GloVe, word2vec, and fastText. The straightforward approach to transform text to a vector is to express text into a numeric representation. The Bag-of-words (BoW) model is a term used to characterize such a representation. Data purification and homogenization are generally the first steps in converting a document to BoW from plain text. Then the compound value obtained from TF and IDF is used to calculate the weight. Both TF and IDF can be computed using various methods but the most popular TF-IDF approach assigns the term  $x$  and the weight to document  $y$  as follows:

$$TF - IDF = tf_{x,y} \times \log\left(\frac{N}{df_i}\right) \dots \dots \dots (5) \text{ Where } tf_{x,y} = \text{total number}$$

of occurrences of a word  $x$  contained in document  $y$

$N$  = total number of records in the whole dataset

$df_i$  = total number of records containing term  $x$

Finally, to produce a final prediction, many classification models are trained using popular ML techniques— fastText, SVM, XGBoost, Keras' CNN.

**Table. 1. Work reported by various researchers**

Authors	Methodology	Future Scope / Enhancement	Accuracy(%)
Panchal et al. in [1]	Using Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Logistic Regression (LR) classifiers	Using new classification techniques and algorithms, accuracy can be improved with a larger amount of complicated and unstructured data.	KNN- 45.46% LR – 45.46% SVM –54.55%
Gupta et al. in [2]	Using Ada-boost Classifier	It can also be used to forecast news item and bloggenres.	81%
Rahul et al. in [3]	Using Character Networks	It can be improved by employing neural co-referencing techniques	60% - 70%
Eran et al. in [4]	Using Word2vec algorithm and several machine learning models - CNN, RNN, GRU and LSTM	With the application of attention mechanisms, the models at hand may be enhanced even further.	65%
Zhenzong et al. in [5]	Using Latent Dirichlet Allocation (LDA) and Softmax Regression technique	Improving the model.	Recall: >=79%
Yao et al. in [6]	Using fastText approach	Improving text categorization models by applying various weights to features.	Precision – 0.92 Recall – 0.93 F-value – 0.92
Zhang et al. in [7]	Using Mahalanobis distance based KNN	MDKNN can be utilized since it is more accurate than	70% - 90%

	classifier	KNN.	
Wenqian et al. in [8]	Using novel feature weight method – Gini Index	Improving the performance of the TF-Gini algorithm for categorization.	The TF-Gini method, when used to weight features in classifiers, produces better classification results than other feature weight algorithms.
Shaohui et al. in [9]	Using Vector Space Model	The VSM representation of a document loses a lot of information about term association, which would be a future topic.	High precision and recall
Alan et al. in [10]	Using Word Embeddings	Apply the algorithms to data to PubMed data.	LCA F <sub>1</sub> -0.89

### 3 conclusion

As the amount of data grows exponentially in the modern day, the need for text data classification and categorization grows. Machine learning techniques may be useful in resolving this problem. Email filtering, chat message filtering, news feed filtering, and other industries can all benefit from text categorization. It has also been observed in locations such as libraries, bookstores, and eBook sites where books are not classified by genre. By reiterating this point, the main aim here is to use machine learning techniques to classify the books by genre and text categorization tools, which will aid in the classification of books by genre based on the title and abstract. The extensive study of studies by various writers connected to the subject is done in this paper.

**Acknowledgments.** The heading should be treated as a 3<sup>rd</sup> level heading and should not be assigned a number.

#### References

1. Shiroya, P., Vaghasiya, D., Soni, M., Patel, V., & Panchal, B. Y. (2021). Book genre categorization using machine learning algorithms (K-nearest neighbor, support vector machine and logistic regression) using customized dataset. *Int. J. Comput. Sci. Mobile Comput*, 10(3), 14-25.
2. Gupta, S., Agarwal, M., & Jain, S. (2019, January). Automated Genre Classification of Books Using Machine Learning and Natural Language Processing. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering

- (Confluence) (pp. 269-272). IEEE.
3. Agarwal, D., & Vijay, D. (2021, May). Genre Classification using Character Networks. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 216-222). IEEE.
  4. Ozsarfati, E., Sahin, E., Saul, C. J., & Yilmaz, A. (2019, February). Book genre classification based on titles with comparative machine learning algorithms. In 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS) (pp. 14- 20). IEEE.
  5. Li, Z., Shang, W., & Yan, M. (2016, June). News text classification model based on topic model. In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS) (pp. 1-5). IEEE.
  6. Yao, T., Zhai, Z., & Gao, B. (2020, March). Text classification model based on fasttext. In 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIS) (pp. 154-157). IEEE.
  7. Zhang, S., & Pan, X. (2011, March). A novel text classification based on Mahalanobis distance. In 2011 3rd International Conference on Computer Research and Development (Vol. 3, pp. 156-158). IEEE.
  8. Shang, W., Dong, H., Zhu, H., & Wang, Y. (2008, October). A novel feature weight algorithm for text categorization. In 2008 International Conference on Natural Language Processing and Knowledge Engineering (pp. 1-7). IEEE.
  9. Liu, S., Dong, M., Zhang, H., Li, R., & Shi, Z. (2001, October). An approach of multi-hierarchy text classification. In 2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No. 01EX479) (Vol. 3, pp. 95-100). IEEE.
  10. Stein, R. A., Jaques, P. A., & Valiati, J. F. (2019). An analysis of hierarchical text classification using word embeddings. *Information Sciences*, 471, 216-232.
  11. Holly Chiang, Yifan Ge and Connie Wu, (2019). Classification of Book Genres By Cover and Title, IEEE International Conference on Intelligent Systems and Green Technology (ICISGT).
  12. Patel, S. H., & Aggarwal, D. BGCNet: A Novel Deep Visual Textual Model for BookGenre Classification
  13. Xu, Z., Liu, L., Song, W., & Du, C. (2017, July). Text genre classification research. In 2017 International Conference on Computer, Information and Telecommunication Systems (CITS) (pp. 175-178). IEEE.
  14. Feldman, S., Marin, M. A., Ostendorf, M., & Gupta, M. R. (2009, April). Part-of-speech histograms for genre classification of text. In 2009 iee international conference on acoustics, speech and signal processing (pp. 4781-4784). IEEE.
  15. Liu, R., Jiang, M., & Tie, Z. (2009, August). Automatic genre classification by using co-training. In 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery (Vol. 1, pp. 129-132). IEEE.
  16. Saini, T., & Tripathi, S. (2018, March). Predicting tags for stack overflow questions using different classifiers. In 2018 4th International Conference on Recent Advances in Information Technology (RAIT) (pp. 1-5). IEEE.

- i. Parwez, M. A., & Abulaish, M. (2019). Multi-label classification of microblogging texts using convolution neural network. *IEEE Access*, 7, 68678-68691.
17. Yang, Z., & Liu, G. (2019). Hierarchical sequence-to-sequence model for multi-label text classification. *IEEE Access*, 7, 153012-153020.
18. Helaskar, M. N., & Sonawane, S. S. (2019, September). Text Classification Using Word Embeddings. In 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA) (pp. 1-4). IEEE.
19. Holanda, A. J., Matias, M., Ferreira, S. M., Benevides, G. M., & Kinouchi, O. (2019). Character networks and book genre classification. *International Journal of Modern Physics C*, 30(08), 1950058.
20. Mason, J. E., Shepherd, M., & Duffy, J. (2009, January). An n-gram based approach to automatically identifying web page genre. In 2009 42nd Hawaii International Conference on System Sciences (pp. 1-10). IEEE.
21. Kundu, C. S. (2020). Book Genre Classification By Its Cover Using A Multi-view Learning Approach.
22. Kundu, C., & Zheng, L. (2020). Deep multi-modal networks for book genre classification based on its cover. arXiv preprint arXiv:2011.07658.
23. Worsham, J., & Kalita, J. (2018, August). Genre identification and the compositional effect of genre in literature. In Proceedings of the 27th international conference on computational linguistics (pp. 1963-1973).
24. Jordan, E. (2014). Automated genre classification in literature (Doctoral dissertation, Kansas State University).
25. Sobkowicz, A., Kozłowski, M., & Buczkowski, P. (2017, October). Reading Book by the Cover—Book Genre Detection Using Short Descriptions. In International Conference on Man–Machine Interactions (pp. 439-448). Springer, Cham.
26. Zheng, W., & Jin, M. (2020). Comparing multiple categories of feature selection methods for text classification. *Digital Scholarship in the Humanities*, 35(1), 208-224.
27. Zheng, Y. (2019, November). An exploration on text classification with classical machine learning algorithm. In 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI) (pp. 81-85). IEEE.
28. Dong, Y., Liu, P., Zhu, Z., Wang, Q., & Zhang, Q. (2019). A fusion model-based label embedding and self-interaction attention for text classification. *IEEE Access*, 8, 30548-30559.
29. De Dillmont, T. (1987). *Encyclopedia of Needlework*. Editions Th. de Dillmont.
30. Sarangi, P. K., Sahoo, A. K., Nayak, S. R., Agarwal, A., & Sethy, A. (2022). Recognition of Isolated Handwritten Gurumukhi Numerals Using Hopfield Neural Network. In *Computational Intelligence in Pattern Recognition* (pp. 597-605). Springer, Singapore.
31. A. Sethy, A. K. Raut and S. R. Nayak, "Face Recognition Based Automated Recognition System," 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2022, pp. 543-547, doi: 10.1109/Confluence52989.2022.9734135.
32. Sethy, A., Patra, P. K., Nayak, S., & Jena, P. M. (2017, October). Symmetric axis based

- off-line Odia handwritten character and numeral recognition. In 2017 3rd International Conference on Computational Intelligence and Networks (CINE) (pp. 83-87). IEEE.
33. Sethy, A., & Patra, P. K. (2021, March). Discrete Cosine Transformation Based Approach for Offline Handwritten Character and Numeral Recognition. In *Journal of Physics: Conference Series* (Vol. 1770, No. 1, p. 012004). IOP Publishing.
  34. P Ramprakash, M Sakthivadivel, N Krishnaraj, J Ramprasath. "Host-based Intrusion Detection System using Sequence of System Calls" *International Journal of Engineering and Management Research*, Vandana Publications, Volume 4, Issue 2, 241-247, 2014
  35. N Krishnaraj, S Smys."A multihoming ACO-MDV routing for maximum power efficiency in an IoT environment" *Wireless Personal Communications* 109 (1), 243-256, 2019.
  36. N Krishnaraj, R Bhuvanesh Kumar, D Rajeshwar, T Sanjay Kumar, Implementation of energy aware modified distance vector routing protocol for energy efficiency in wireless sensor networks, 2020 International Conference on Inventive Computation Technologies (ICICT),201-204
  37. Ibrahim, S. Jafar Ali, and M. Thangamani. "Enhanced singular value decomposition for prediction of drugs and diseases with hepatocellular carcinoma based on multi-source bat algorithm based random walk." *Measurement* 141 (2019): 176-183. <https://doi.org/10.1016/j.measurement.2019.02.056>
  38. Ibrahim, Jafar Ali S., S. Rajasekar, Varsha, M. Karunakaran, K. Kasirajan, Kalyan NS Chakravarthy, V. Kumar, and K. J. Kaur. "Recent advances in performance and effect of Zr doping with ZnO thin film sensor in ammonia vapour sensing." *GLOBAL NEST JOURNAL* 23, no. 4 (2021): 526-531. <https://doi.org/10.30955/gnj.004020> , [https://journal.gnest.org/publication/gnest\\_04020](https://journal.gnest.org/publication/gnest_04020)
  39. N.S. Kalyan Chakravarthy, B. Karthikeyan, K. Alhaf Malik, D.Bujji Babbu., K. Nithya S.Jafar Ali Ibrahim , Survey of Cooperative Routing Algorithms in Wireless Sensor Networks, *Journal of Annals of the Romanian Society for Cell Biology* ,5316-5320, 2021
  40. Rajmohan, G, Chinnappan, CV, John William, AD, Chandrakrishan Balakrishnan, S, Anand Muthu, B, Manogaran, G. Revamping land coverage analysis using aerial satellite image mapping. *Trans Emerging Tel Tech.* 2021; 32:e3927. <https://doi.org/10.1002/ett.3927>
  41. Vignesh, C.C., Sivaparthipan, C.B., Daniel, J.A. et al. Adjacent Node based Energetic Association Factor Routing Protocol in Wireless Sensor Networks. *Wireless Pers Commun* 119, 3255–3270 (2021). <https://doi.org/10.1007/s11277-021-08397-0>.
  42. C Chandru Vignesh, S Karthik, Predicting the position of adjacent nodes with QoS in mobile ad hoc networks, *Journal of Multimedia Tools and Applications*, Springer US, Vol 79, 8445-8457,2020