International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 11 | Nov - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

TEXT CLASSIFICATION IN MARATHI LANGUAGE

Mrs. Anudnya Suyog Sawant,
Computer Department
Pillai college of engineering, New panvel
Mumbai University
sanudnya23mtechcomp@student.mes.ac.in

Dr.Sharvari Govilker
Computer Department
Pillai college of engineering, New panvel
Mumbai University
sgovilkar@mes.ac.in

***Abstract*— In today's world, several digitized multiple language text documents are generated daily at the Government sites, news portals, and public and private sectors, which are required to be classified effectively into various mutually exclusive pre-defined categories. Many multiple language text-based processing systems exist in application domains of information retrieval, machine translation, text summarization, simplification, keyword extraction, and other related parsing and linguistic perspectives, but still, there is a wide scope to classify the extracted text of Marathi documents into predefined categories using a classifier. Text classification works for 10 Indian languages and Arabic language are reported in the literature. A Marathi Text Classification model is proposed, which accepts a set of known Marathi documents, preprocesses them at document, sentence and word levels, extracts features, and trains different classifiers, which further classifies a set of Marathi unknown documents. This research aims to understand and reduce the language challenges of Marathi. We want to make it easier to use text classification (sorting and organizing text) in Marathi by improving the methods specifically designed for this language. The goal is to advance and enhance the ways we categorize and analyze Marathi text more effectively.**

***Keywords*— Natural Language Processing, SVM, KNN, Naïve Bayes (Supervised learning methods)**

I. INTRODUCTION

Text classification is a vital task in the field of natural language processing (NLP), involving the categorization of text documents into predefined classes or categories. This process enables automated systems to organize, filter, and make sense of large volumes of textual data. Natural Language Processing (NLP) plays a crucial role in text classification tasks, enabling machines to understand and interpret human language.

Text Preprocessing: NLP is used for cleaning and preprocessing raw text data. Tasks such as tokenization, stemming, and lemmatization are performed to break down sentences into individual words, reduce words to their base forms, and standardize the text.

Feature Extraction: NLP techniques are employed to convert text data into numerical features that machine learning models, including Support Vector Machines (SVM), can work with Common methods include Bag-of-Words (Bow) and Term Frequency-Inverse Document Frequency (TF-IDF), where the frequency or importance of each word is considered.

Word Embedding's: Word embedding's, such as Word2Vec or GloVe, are used to represent words as dense vectors. These embedding's capture semantic relationships between words, providing a more nuanced representation of the textual content and improving the model's understanding of context.

Handling Stop Words and Noise: NLP helps in identifying and filtering out stop words and other noise in the text that might not contribute much to the classification task. This step improves the signal-to-noise ratio in the input data.

Handling Textual Variations: NLP techniques address variations in language, including synonyms and different forms of a word. This ensures that the text classification model can generalize well across different expressions of the same concept.

Named Entity Recognition (NER):For certain text classification tasks, NLP models may incorporate Named Entity Recognition to identify and classify entities such as names of people, organizations, locations, etc. This can be valuable, for example, in classifying documents related to specific entities.

Language Model Integration: Pre-trained language models, such as BERT or GPT, may be fine-tuned for specific text classification tasks. These models, based on deep learning architectures, capture intricate relationships and dependencies within the text, enhancing the model's ability to understand context and semantics.

Topic Modeling: NLP can be applied for topic modeling, a technique that identifies topics present in a collection of documents. This can be useful in certain text classification

scenarios where the goal is to categorize documents based on their underlying themes.

Text classification is the process of categorizing text into predefined categories based on its content. There are two main approaches to text classification: supervised and unsupervised learning.

Supervised Learning

Supervised learning is a machine learning technique that involves training a model on labeled data. In text classification, this means providing the model with a set of documents and their corresponding categories. The model then learns to associate certain features of the text with specific categories, allowing it to classify new documents based on those features.

Benefits

One of the main benefits of supervised learning is that it can achieve high accuracy in text classification tasks. This is because the model is trained on labeled data, which provides it with a clear understanding of what each category represents. Additionally, supervised learning allows for the use of a wide range of features, such as word frequency, n-grams, and part-of-speech tags, which can improve the accuracy of the model.

How to do it

To perform text classification using supervised learning, you can follow these steps:

1. Collect and preprocess your data: This involves gathering a set of documents and cleaning them by removing stop words, punctuation, and other irrelevant information.
2. Label your data: This involves assigning categories to each document in your dataset.
3. Split your data: This involves dividing your dataset into training and testing sets.
4. Train your model: This involves selecting a machine learning algorithm and training it on the labeled training data.
5. Evaluate your model: This involves testing your model on the testing data and measuring its accuracy.

Metatext

Metatext is a powerful text analysis platform that can be used to perform text classification using supervised learning. With Metatext, you can easily upload your data, label it, and train your model using a variety of machine learning algorithms. Metatext also provides a user-friendly interface for evaluating your model and visualizing your results.

Unsupervised Learning

Unsupervised learning is a machine learning technique that involves training a model on unlabeled data. In text classification, this means providing the model with a set of documents without any corresponding categories. The model then learns to identify patterns and similarities in the text, allowing it to group similar documents together.

Benefits

One of the main benefits of unsupervised learning is that it does not require labeled data, which can be difficult and expensive to obtain. Additionally, unsupervised learning can be used to discover new categories or topics in the data that may not have been previously identified.

How to do it

To perform text classification using unsupervised learning, you can follow these steps:

1. Collect and preprocess your data: This involves gathering a set of documents and cleaning them by removing stop words, punctuation, and other irrelevant information.
2. Vectorize your data: This involves converting your text data into numerical vectors that can be used by machine learning algorithms.
3. Cluster your data: This involves using a clustering algorithm to group similar documents together.
4. Evaluate your clusters: This involves analyzing the clusters to determine if they represent meaningful categories or topics.

Metatext

Metatext also provides tools for performing text classification using unsupervised learning. With Metatext, you can easily upload your data, preprocess it, and vectorize it using a variety of techniques. Metatext also provides a range of clustering algorithms that can be used to group similar documents together

Supervised learning and unsupervised learning are two main approaches to text classification, each with its own benefits.Supervised learning can achieve high accuracy but requires labeled data, while unsupervised learning does not require labeled data but may not be as accurate.

In text classification using Support Vector Machines (SVM), categorization refers to the process of assigning predefined categories or classes to text documents based on their content. SVM is a supervised machine learning algorithm commonly used for text classification tasks. Here's a breakdown of the categorization process in text classification with SVM:

1. Training Phase:

Data Preparation: Begin with a labeled dataset where each text document is associated with a specific category or class. The dataset is divided into two subsets: one for training the model and another for testing its performance.

Feature Extraction: Utilize NLP techniques (such as Bag-of-Words, TF-IDF, or word embeddings) to convert the raw text into numerical features that can be used as input for the SVM model. This process involves representing each document as a vector in a high-dimensional space.

Model Training: Train the SVM model using the labeled training data. The SVM algorithm aims to find the optimal hyperplane in the feature space that maximally separates the documents belonging to different categories.

2. Testing and Prediction Phase:

Feature Extraction for Test Data: Apply the same feature extraction techniques used during training to convert the raw text of the test documents into numerical vectors.
Model Prediction: Use the trained SVM model to predict the category of each test document based on its feature vector. The model assigns a category to each document by determining on which side of the hyperplane it falls.

3. Categorization Results:

Evaluation Metrics: Assess the performance of the SVM model by comparing its predictions with the true labels of the test data. Common evaluation metrics for text classification include accuracy, precision, recall, and F1-score.
Confusion Matrix: Create a confusion matrix to visualize the number of true positives, true negatives, false positives, and false negatives. This provides insights into how well the model is performing for each category.

4. Optimization and Fine-Tuning:

Based on the evaluation results, consider optimizing the SVM model. This may involve adjusting hyper parameters, exploring different feature representations, or addressing issues like over fitting or under fitting.

5. Deployment:

Once satisfied with the model's performance, deploy it for categorizing new, unseen text documents. The deployed model can be used to automatically assign categories to incoming textual data based on the patterns it learned during training.
In some researches marathi language is used for text classification. In that, multiple techniques are used. Naïve Bayes & Centroid Based gives 99.167% accuracy, K-Nearest Neighbor (KNN) gives 97.17% accuracy & Modified KNN (MKNN) gives 99.66% accuracy. supervised learning methods include Naïve Bayes (NB), Modified K Nearest Neighbor(MKNN) and Support Vector Machine (SVM) gives higher accuracy. The classification techniques MKNN, KNN, Naïve Bayes, Centroid gives best accuracy and one of the clustering techniques i.e. LINGO algorithm.
The Lingo algorithm employs term-document matrix dimensionality reduction techniques to figure out the structure of topics present in the input. The algorithm is reasonably fast and nicely separates diverse topics into separate groups. This will typically be the default algorithm you may want to use.

II. LITERATURE SURVEY

This chapter consists of the literature survey of various systems that are used in text classification of Marathi language by using various machine learning and deep learning algorithms.

Ehsan Othman, Ayoub Al-Hamadi proposed this system by using Arabic language[1]. They used the HRWiTD algorithm for classification. In this paper, the average of the overall classification accuracy for six categories is 86.84 %. The HRWiTD algorithm needs to be improved to get better results to classify all text categories. It needs to extend the experimental corpus from different resources to demonstrate efficiency.

Nawal Aljedani, Reem Alotaibi, Mounira Taileb developed this system by using Arabic language[2]. They used the Hierarchy Of Multilabel Classifier (HOMER) algorithm for classification. The results showed that the HMATC model accuracy is 75.80%. This study can apply different structured methods for selecting the number of clusters in the clustering algorithm (k).

Abdullah Y. Muaada,b, G. Hemantha Kumar J. Hanumanthappaa, J.V. Bibal Benifac M. Naveen Mouryaa Channabasava Cholac M.Pramodhaa, R. Bhairavaa uses Arabic language[3]. They used Multinomial Naïve Bayesian (MNB), Bernoulli Naïve Bayesian (BNB), Stochastic Gradient Descent (SGD), Logistic Regression (LR), Support vector classifier (SVC), Linear SVC, and convolutional neural networks (CNN). CNN gives a good result (98%) that outperforms all others. In this paper, author can collect more data, and work with different transfer learning models

Jasleen Kaur, Dr. Jatinderkumar R. SAINI uses Punjabi, Bengali, Urdu, Telugu, Tamil, Kannada languages[4]. They used Naive Bayes (NB), Support Vector Machine (SVM), Artificial Neural Network (ANN), and N-gram techniques for classification. This study shows that supervised learning algorithms (Naive Bayes (NB), Support Vector Machine (SVM), Artificial Neural Network (ANN), and N-gram) performed 90% to 93% for Text Classification tasks. This paper only get the accuracy up to 93.3%, we have to try to get more accuracy.

Md. Rajib Hossain , Mohammed Moshiul Hoque, Nazmul Siddique, Iqbal H. Sarker developed this system by using Bengali language[5]. They used VDCNN for classification. VDCNN model achieved the highest accuracy of 96.96% for Bengali text classification. In this study author can be try to add more datasets

Md. Anwar Hussen Waduda, Muhammad Mohsin Kabir a , M.F. Mridha b, M. Ameer Ali a, Md. Abdul Hamidc, Muhammad Mostafa Monowar c developed this system by using Bengali language[6]. They used the LSTM-BOOST algorithms for classification. The LSTM-BOOST algorithms outperform most of the baseline architecture, leading F1-score of 92.61% on the Bengali offensive text. author can be used here semi-supervised model & also try to use large datasets

Rajnish M. Rakholia and Jatinder kumar R. Saini uses Gujarati language[7]. They used Naïve Bayes for classification. Results show that the accuracy of NB classifiers without and using features selection was 75.74% and 88.96% respectively. Extend this work by adding new category in Ontology which can be used in other research in area of Natural Language Processing

Batoul Aljaddouh, Nishith A. Kotak uses English language[8]. They used SVM techniques. Accuracy is 96.6%**.** This study can be extended for other languages to make document classifiers more versatile.

Pooja Saigal Vaibhav Khanna developed text classification by using English language[9]. They used LS-SVM, TW-SVM, LS-TWSVM techniques. LS-SVM has 91% accuracy. TW-SVM Machine has 96% accuracy. LS-TWSVM has 98% accuracy. This study can be achieve for other SVM based multi-category approaches & also other different techniques

F R Lumbanraja, E Fitri, Ardiansyah A Junaidi ,Rizky Prabowo uses English language[10]. They used SVM for classification. It produces the greatest accuracy value, namely 58.3%. In this study the author can be try to apply more techniques with more features.

Xiaoyu Luo proposed using the English language. They used SVM, Naive Bayes and Logistic Regression technique[11]. SVM techniques provide more efficiency(88%) as compared to Naive Bayes and Logistic Regression. Here he can try for other techniques so by using other machine learning technique accuracy may be get more

Ramchandra Joshi, Purvi Goel and Raviraj Joshi proposed this system by using Hindi language[12]. They used Convolutional neural networks, BOW,BOW + Attention, LSTM,Bi-LSTM, CNN + Bi-LSTM, Bi-LSTM + Attention, LASER and BERT. Out of all the models CNN performs the best (90%)for all the datasets. They can be use more dataset and also try to find out more accuracy

Shalini Puri and Satya Prakash Singh use Hindi language[13]. They used SVM for classification. Its classification accuracy is 100%. This work can be extended into the areas of HTC implementation on a large set of documents also by using other techniques.

Mr. Ishaan Tamhankar, Dr.Ashish Chaturvedi by using Hindi language[14]. They used Naïve Bayes (NB), Support Vector Machines (SVM) and K-NN (K – Nearest Neighbors). They can achieve good accuracy by being more influential and related to the particular domain specific category. NB classifiers consider each word as an independent word in a document and needs training to implement. we can extend this work by adding new category in Ontology which can be used in other research in area of Natural Language Processing

Rupali P. Patil, R. P. Bhavsar, B. V. Pawar uses marathi language. They used Naïve Bayes, K-Nearest Neighbor, Support Vector Machine, Centroid Based and Modified KNN (MKNN) for classification[15]. Naïve Bayes and Centroid Based give best performance with 99.16% Micro and Macro Average of F-score and MKNN gives lowest performance with 97.16% Micro Average of F-Score and 96.99%. This study can try for effect of feature selection on classification using same classification methods

Pooja Bolaj, Dr. Sharvari Govilkar developed by using Marathi languages [16]. They used Support Vector Machine, Naïve Bayes, Modified K Nearest Neighbor methods for text classification. These techniques provide better results in the form of accuracy and time efficiency. This paper gives accuracy but by using marathi language we want 100% accuracy, it is not mentioned as it is in this paper.

The review on existing literature identifies some research gap points which are given below.

- Limited accuracy in some studies, indicating the need for improvement.
- Suggestions for extending studies by adding new categories to Ontology or exploring additional techniques.
- Emphasis on collecting more data and exploring different transfer learning models.
- Calls for studying the effect of feature selection on classification in specific languages.

III. PROPOSED SYSTEM

Proposed system explains the input code of the multiple algorithms. In “text classification in Marathi language” we use Marathi news dataset which is used to classify the news in different classes as label name. For Marathi text classification we create 4 classes which are ‘Maharashtra’, ‘Bollywood’, ‘Sports’ and ’Others’. So, Marathi News are classified in these 4 classes (label) with their accuracy, precision, F1-score and Recall.

.

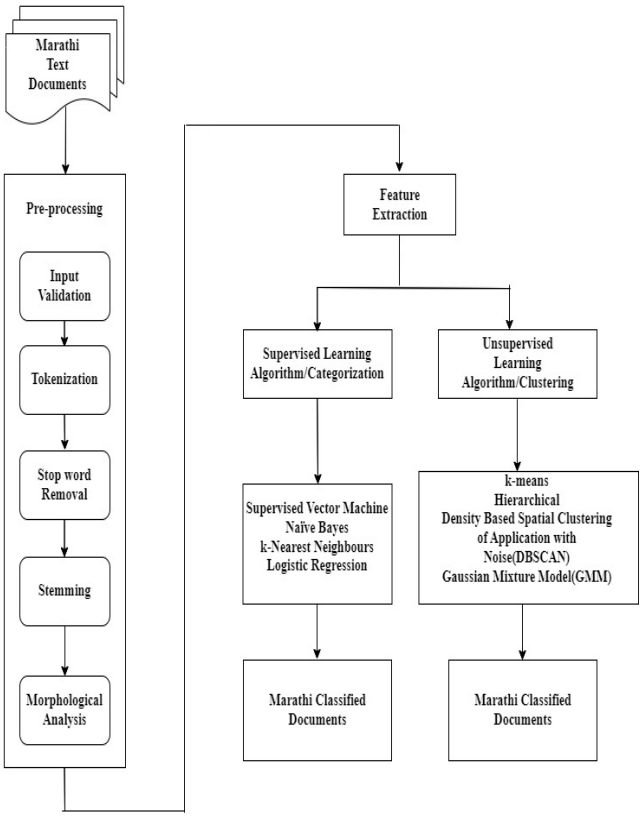

Fig. 1 proposed Architecture

IV. METHODOLOGY

The system takes input as a set of Marathi language documents. The documents undergo pre-processing steps which include input validation, tokenization, stop word removal, stemming and morphological analysis. Then the features are extracted from pre-processed tokens. Finally supervised and unsupervised machine learning method based classification is applied to get output as classified Marathi documents as per class label.

The existing system approach consists of following phases:

1. Preprocessing
    - 1.1) Input Validation
    - 1.2) Tokenization
    - 1.3) Stop word removal
    - 1.4) Stemming
    - 1.5) Morphological Analysis
2. Feature Extraction
3. Supervised Learning Methods
4. Output as classified documents

1. Preprocessing:

1.1 Input Validation:

Objective: Ensure the input Marathi text documents are in Devanagari script.Validate each document to check if it adheres to the Devanagari script, which is commonly used for writing Marathi.Remove any words or sentences that are not valid in Devanagari script.

1.2 Tokenization:

Objective: Break down the validated text into individual tokens (words). Use spaces to identify word boundaries in the text and segment the document into a sequence of tokens.

1.3 Stop Word Removal:

Objective: Eliminate common stop words to enhance processing speed. Remove frequently occurring and generally uninformative words (stop words) from the tokenized text. This step helps reduce noise in the data and focuses on more meaningful words.

1.4 Stemming

.Objective: Reduce words to their base or root form. Apply stemming using a suffix list to remove suffixes from words.
The result is a simplified form of each word, called its stem.

1.5 Morphological Analysis:

Objective: Analyze the structure of words after stemming. Examine the words to determine whether they are inflected. The goal is to recognize the inner structure of each word.

2. Feature Extraction:

Objective: Convert preprocessed text into numerical features for machine learning models. Use a Marathi Dictionary to compute a feature vector for each input text document. The feature vector includes important features from the text document and their frequencies.

3. Performance Evaluation:

The performance of Text Classification System can be evaluated by using four metrics: Accuracy, Precision, Recall and F1 measure. Precision measures the exactness of a classifier.

$$\text{Precision} = \frac{\text{No.of correct extracted text}}{\text{Total No. of extracted text}}$$

Recall measures the completeness, or sensitivity, of a classifier.

$$\text{Recall} = \frac{\text{No.of correct extracted text}}{\text{Total No. of annotated text}}$$

Precision and recall can be combined to produce a single metric known as F-1 score, which is the weighted harmonic mean of precision and recall. The main advantage of using F-1 score is it is able to rate a system with one unique rating.

$$\text{F-1 score} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

Accuracy measures the overall degree to which instances have been correctly classified, using the formula as defined below

$$\text{Accuracy} = \frac{\text{No.of correct classified instances}}{\text{Total No.of instances}}$$

4. Supervised Learning Methods:

Objective: Use supervised learning techniques to classify Marathi text documents. Employ algorithms such as Naïve Bayes (NB), Modified K Nearest Neighbor (MKNN), and Support Vector Machine (SVM). These algorithms are trained on labeled data, where each document is associated with a predefined category.

i. **Naïve Bayes Algorithm**:

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability. The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where, P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B. P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

ii. **Support Vector Machine (SVM) algorithm:**

The main idea of SVM is to find a hyper-plane that best separates the documents and the margin, distance separating the border of subset and the nearest vector document, is large as possible. The nearest samples of the hyper-plane named support vectors are selected. The calculated hyper-plane permits to separate the space in two areas. To classify the new documents, calculate the area of the space and assign them the corresponding category.

iii. **K-Nearest Neighbor Algorithm:**

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

5. Output as Classified Documents:

. The output consists of classified documents, each assigned to a specific category or class based on the applied learning methods. The classification is the result of the training and application of supervised and potentially unsupervised learning models.

V. Conclusion and future work

The scope of the proposed automatic Marathi Text Classification system using Supervised and Unsupervised Learning methods explores a new and progressively wide range of applicability in NLP and text document classification. Although many other Devanagari scripted language-based systems have been successfully implemented and presented with varying types of languages, structures, and contents, the Marathi Text Classification method is oriented toward accurate text (keyword) identification, extraction and finally its classification with more accuracy. According to above results in Marathi text classification system SVM algorithm achieves

more accuracy compared to Naïve Bayes and k-Nearest neighbors algorithms. As we have seen, no existing system classifies the text into predefined categories by accepting Marathi documents, performing a series of operations on them, and classifying the unknown documents using Supervised and Unsupervised Learning methods with highest accuracy.This work can further be extended into the areas of Marathi Text Classification implementation on a large set of documents, dimensionality reduction, and imaged document classification.

VI. ACKNOWLEDGEMENT

I remain immensely obliged to Dr. Sharvari Govilkar, for providing me with the idea of this topic, and for her invaluable support in garnering resources for me either by way of information or computers and also her guidance and supervision which made this seminar happen. I would like to thank our principal Dr. Sandeep Joshi, the Head of Department of the Computer Engineering Dr. Sharvari Govilkar, Dr. Prashant Nitnaware and Coordinator of Department of Computer Engineering Dr. Jyoti Deshmukh for their invaluable support. I would like to say that it has indeed been a fulfilling experience working out on this project idea. I would like to specially thank my project guide Dr. Sharvari Govilkar for supporting and helping me in all respects.

VII. REFERNCES

[01] Othman, Ehsan, and Ayoub Al-Hamadi. "Automatic Arabic Document Classification Based on the HRWiTD Algorithm." *Journal of Software Engineering and Applications* 11.04 (2018): 167.

[02] Aljedani, Nawal, Reem Alotaibi, and Mounira Taileb. "HMATC: Hierarchical multi-label Arabic text classification model using machine learning." *Egyptian Informatics Journal* 22.3 (2021): 225-237.

[03] Muaad, Abdullah Y., et al. "An effective approach for Arabic document classification using machine learning." *Global Transitions Proceedings* 3.1 (2022): 267-271.

[04] Kaur, Jasleen, and Jatinderkumar R. Saini. "A study of text classification and natural language processing algorithms for Indian languages." *VNSGU J Sci Technol* 4.1 (2015): 162-167.

[05] Hossain, Md Rajib, et al. ``Bengali text document categorization based on a very deep convolutional neural network." *Expert Systems with Applications* 184 (2021): 115394.

[06] Wadud, Md Anwar Hussen, et al. "How can we manage offensive text in social media-a text classification approach using LSTM-BOOST." *International Journal of Information Management Data Insights* 2.2 (2022): 100095.

[07] Rakholia, Rajnish M., and Jatinderkumar R. Saini. "Classification of Gujarati documents using Naïve Bayes classifier." *Indian Journal of Science and Technology* 5 (2017): 1-9.

[08] Batoul Aljaddouh, Nishith A. Kotak. “Document Text Classification Using Support Vector Machine”. Publication Since 2012 | ISSN: 2321-9939 | ©IJEDR 2020 Year 2020, Volume 8, Issue 1.

[09] Saigal, Pooja, and Vaibhav Khanna. "Multi-category news classification using support vector machine based classifiers." *SN Applied Sciences* 2.3 (2020): 458.

[10]. Lumbanraja, Favorisen R., et al. "Abstract classification using support vector machine algorithm (case study: abstract in a Computer Science Journal)." 28 *Journal of Physics: Conference Series*. Vol. 1751. No. 1. IOP Publishing, 2021.

[11]. Luo, Xiaoyu. "Efficient English text classification using selected machine learning techniques." *Alexandria Engineering Journal* 60.3 (2021): 3401-3409.

[12]. Joshi, Ramchandra, Purvi Goel, and Raviraj Joshi. "Deep learning for hindi text classification: A comparison." *Intelligent Human Computer Interaction: 11th International Conference, IHCI 2019, Allahabad, India, December 12–14, 2019, Proceedings 11*. Springer International Publishing, 2020.

[13]. Puri, Shalini, and Satya Prakash Singh. "An Efficient Hindi Text Classification Model Using SVM Computing and Network Sustainability Book." (2019).

[14]. Ishaan, T., and C. Ashyush. "Classification of spam categorization on Hindi documents using Bayesian Classifier." *IOSR J. Comput. Eng* 20.6 (2018): 53-58.

[15]. Patil, Rupali P., R. P. Bhavsar, and B. V. Pawar. "Automatic marathi text classification." *Int. J. Innovat. Technol. Expl. Eng* 9.2 (2019): 2446-2454.

[16]. Bolaj, Pooja, and Sharvari Govilkar. "Text classification for Marathi documents using supervised learning methods." *Int. J. Comput. Appl* 155.8(2016): 6-10.

.