Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 4s (2025)

Deep Learning-Based Hybridized Bi-LSTM Model for Author Identification in the Marathi Language

Sunil D. Kale[1], Chudaman D. Sukte[2], Sumit A. Hirve[3], Swapnil K. Shinde[4], Amar Buchade [5], Nilesh Sable[6]

[1, 5]Department of Computer Engineering & Information Technology, Veermata Jijabai Technological Institute, Mumbai, India. Email: kalesunild@gmail.com

[2]Department of Information Technology, Vishwakarma Institutte of Information Technology, Pune, India. Email: rajeshsukte@gmail.com

[3]Department of Computer Science & Engineering, MIT Art Design and Technology University, Pune, India. Email: sumit.hirve@gmail.com

[4,5]Department of Artificial Intelligence & Data Science, Vishwakarma Institute of Information Technology, Pune, India. Email: [4]swapnil.shinde@viit.ac.in, [5]amar.buchade@viit.ac.in

[6]Department of Computer Science & Engineering (Artificial Intelligence), Vishwakarma Institute of Information Technology, Pune, India. Email: nilesh.sable@viit.ac.in

Article History:

Received: *10-10-2024*

Revised: *20-11-2024*

Accepted: *01-12-2024*

Abstract:
Nowadays, author identification system for linguistics articles has imperatively needed for copy right violations. It is the activity in which linguistic attempts to identify the original author of an unknown textual information based on the utilized vocabulary and the writing style of the author. There have a lot of existing studies focusing on popular languages like English, Spanish, Chinese, and so on. In this paper, we propose an author identification system for Indian language Marathi based on hybridized Bi-directional long short-term memory (Bi-LSTM) model. The Marathi language totally differs from other popular languages because it uses the Devanagari script. It is one of the complicated scripts available in most Indian languages. The key idea of proposed author identification system is to extract high level feature representation from Marathi script using very deep convolutional neural network (VD-CNN) and hybridized Bi-LSTM model used for the identification of unknown authors. The VD-CNN based feature extraction is performed based on the input of two textual data analysis methods such as Term Frequency- Inverse Document Frequency (TF-IDF) and word embedding. The proposed model is tested with Author wise Marathi Language Text Corpus and simulation results show that the average accuracy for author identification has reached 96.01% when using TF-IDF and 99.16% with word embedding. Moreover, the proposed hybridized Bi-LSTM model provides better performance compared to stand-alone convolutional neural network (CNN) and Recurrent Neural Network (RNN) based models.

Keywords: Author Identification, Deep Feature Extraction, Bi-directional Long short-term Memory, Convolutional Neural Network.

1. Introduction

From the past years, many types of research has already done on the domain of text mining for identifying the author of unknown articles. Various text articles are shared by posting it on social media, blogs or websites by the internet users. The mining techniques are used here to identify the unauthorized usage of articles [1]. The detection of the real author with the text content is a difficult process and it can be done by matching some features like characters present in the document, structure of a document, the special character used, and so on. The author identification application is used in several domains like computer forensics, to detect plagiarized content, and civil law. The manual author identification is not possible with electronic contents because of some issues like huge amounts of content and different languages [2].

Most of the research articles used the writing style of the author as a feature to identifying the writer of the text content. The feature extraction process of text content is a major process and it includes commonly used words, length of words, sentence, paragraph, special characters, etc. The features will be obtained or be extracted from the text content which is used as a dataset. These obtained features are used for the identification of the author [3]. The features are classified as character-based, word-based, sentence-based, and syntactic based. Character-based features include the commas, periods, and other punctuations used by the author. In other cases, if the author is more emotional then, he will use more question marks and exclamatory symbols. Word-based features include the words frequently used by the author. It may make the result more bareness and it is mostly used on deep learning-based techniques. Sentence based features contain information about the sentence present in the document. Which means, the author may use either nor short or long sentences and paragraphs. Syntactic based features include the grammatical relation among the words in every sentence and which are collected with the syntactic analysis tool [4].

The authorship of the article is classified with the presence of the author and the profile of the author will be obtained based on the writing of the author [5]. The article mostly shared on the internet is in XML, PDF, or HTML format. Every electronic document includes three several levels, they are document level, sentence level, and internal level. The word distribution, research topic and ideas are included in the document level. Some relation among entities, syntax are included in the sentence-level and finally, sections and paragraphs are included in the internal level to convey the topic of the article [6].

The features extracted from the text document is used for identifying the author. The classifier is designed with the base of features to get the ability to detect the author. The collected features are enough to differentiate several authors and it is much hard to identify the authors with the same name. The best classifier will be designed only with suitable features and classifying algorithms. For classifying the right author various techniques are used for the classification. In [7] and [8] decision tree, Support Vector Machine (SVM) and Convolution neural network (CNN) is used for the classification. Researchers also claimed, however, that classification of Marathi scripts is a very challenging process because of language difficulty [17, 18]. Another big issue related with classification of Marathi script is the absence of standardized and imbalanced datasets [19]. There are so many languages are there in the world. Among one is English and most of the author identification techniques are focusing on this language because of its popularity and usage. Only limited studies

presents a scheme to recognize the nameless authors of articles written on an Indian language Marathi [20, 21].

The major intention of proposed model is to identifying the right author of Marathi article by using the help of Convolution neural network (CNN) and Bidirectional Long short-term memory (Bi-LSTM) based hybridized model. The proposed model investigates the suitability of VD-CNN for feature learning to obtain high level intrinsic feature representation. Moreover, we analyze the effectiveness of different word representations using VD-CNN model, as well as employing hybridized Bi-LSTM model for Marathi article author identification. The major contribution of this work is summarized as follows.

- We developed feature learning and extraction methods using CNN based deep learning model. In this process, two different word representations are individually inputted to the VD-CNN model to make high level and intrinsic feature representation that support to effective author identification of Marathi articles.
- Before initiating the feature learning process employed by VD-CNN, we represent article words using well recognized methods for textual data analysis. The first method is to employ term frequency-inverse document frequency (TFIDF) to find a chief word in an article corpus, which needs prior knowledge about article words. The second method is to employ dense representation of article words using word embedding model to obtain a similar meaning word representation. The proposed VD-CNN model does not need any prior knowledge about articles for feature learning.
- We employed a hybridized Bi-LATM model for large scale article author identification process and demonstrated that our proposed model can succeed the identification of a large number of authors while achieving high accuracy.

The rest of the paper is organized as follows. The relevant related works are reviewed in Section 2. The theoretical background of core methods needed for understanding our work is discussed in Section 3. In Section 4 we explain our CNN-Bi-LSTM based model for author identification. The comprehensive discussion of simulation results from our proposed model is presented in Section 5. At the end, we provide the conclusion part of the paper in Section 6.

2. RELATED WORKS

This section describes several recent articles that are focused on author identification based tasks with different methods. There most of the researchers are provided several solutions on nameless author identification of articles. The process of classifying the authorship is a much complex task and it aims to provide an answer for the question “who wrote the article” or “which author’s content is this”. For this process methods like deep learning [9], CNN [8], syntactic features [4] are used in recent research articles.

Stefano Ferilli et al. 2015 [10] offered a tool to classify the authors or text with the use of various linguistic features. This tool works based on the clustering technique and its results are can’t be read by humans. There is no need for state-of-art systems to handle some issues when using this tool. They conduct several activities on their pre-processing steps like collocation extraction, dependencies extraction, and normalization of the term. Every sentence present in the article will be translated into relational patterns. The procedure used in this paper for author identification estimates the similarities

among all sentences. The tool designed in this paper is used only in the English language based articles. The results of this work show that this tool got better performance even with short texts than state-of-art techniques. The researchers of this article tell that their tool will support all-natural languages. They also planned to improve the identification performance of their future work.

NektariaPotha et al. 2017 [11] suggested a method for author identification which uses the updated imposter's technique it increases the identification process. The updated technique used in this work is also said to as General Impostors (GI). The similarity of text while analyzing the article is used instead of using the impostor texts. The method read the location of the author's text and give a grade in decreasing order based on the comparisons. They first insert both types of documents (known and unknown authors) and provide the score for that and it will be noticed as a possibility of an optimistic answer. This GI based method, analyze the known and unknown articles of similar author. It makes to increase the performance of author identification for these different articles of similar author. In their article, they presented an algorithm for the updated GI technique. On the GI method, two changes are carried out, imposter article per verification and improve the data that is saved in every repetition. They conducted various verification for authorship and verified that it feasibly enhance the performance. Their results describe the efficiency of the GI method and their future work focusing on observing other text representation patterns.

The author verification process is carried out on different languages such as English, Arabic [12], and Tamil [3] and so on. Mahmoud Al-Ayyoubet al.2017 [13] presented an article for identifying the author of Arabic articles. Their method focuses on the extraction and selection features of the text content. They collected the features based on the MADAMIRA features, Stylometric features, and BOW features. Which are designed by several researchers and every feature extracting method has its own properties to collect the features. They used several classifying algorithms to classify the text content. They perform the classification process with the selected features. In this article, they only focused on the Arabic language-based articles because of the limitation in size, slang and informal languages. They achieved good results by merging all the feature sets and they got big improvements in the running time of classifiers. In their future work, they planned to include other feature selection and classifying techniques. In addition, they enlarge their work for supporting a huge dataset.

KazemTaghva 2017 [14] proposed a name identification system with a formal concept analysis (FCA) algorithm. Initially, they obtain the characteristics of the name in exact language. These characteristics may include linguistic clues, attributes and symbols. With this characteristic of the text content, the FCA will be processed. Hidden Markov Model (HMM) is used for identification and CNN is used for classifying the name. This method mainly focused on DOE documents and achieved good results. They expected to enlarge their method for identifying the date, place and organization along with the name.

Natural Language Processing (NLP) is an important task for mining the text and identifying the author of documents. The NLP includes three activities such as, authorship attribution (AA), author profiling (AP), and discriminating between similar languages (DSL). These can perform operations like detecting the author of the text, describing the aspects of the author, and predicting the language variety [15]. Stefano Ferilli 2016 [16] proposed a method for the classification of the author based on sentence structure. The relational settings used in this work helps to handle difficult syntactic structures. The

text content will be pre-processed and change for the structural illustration for identifying and modeling the author and algorithm. The pre-processing process is carried out based on term normalization, collocation extraction, parsing, and dependencies extraction. The structure of the sentence will be obtained during the representation formalism process. Similarities present in the text content is also collected as the feature. They presented the algorithm of their method in their article. This work is compared with state-of-art techniques, they achieved better results and this method is much portable for analyzing the short text contents. It can able to classify the author more efficient and its classification process is much reliable than other state-of-art methods. They are trying to apply their technique for detecting the plagiarized contents on text documents. They also planned to improve the prediction time of nameless large article of several authors.

3. Theoretical Background

In this section, we discuss the core methods used for author identification with respect to article feature representation and feature learning.

3.1 Article Text Representation

Normally in NLP applications, textual information especially sentences are represented in the form of matrix where the representation of tokens and words are stored in the matrix rows. These word vectors representation of rows are referred as word embedding's. Generally, word embedding's are low-dimensional representations learned from a specific article corpus using a frequency-based method or prediction based method. The frequency based methods comprise techniques based on TF-IDF and n-gram occurrence models. The prediction based methods comprise techniques such as word2vec models and word embedding's. In word embedding technique, the word in the article is represented as one-hot vector. By the one-hot vectors as input, the deep learning techniques learning the embedding of sentence and articles during the process of training.

The TF-IDF technique eases the model architecture by the computation of one-step fixed-sized vector sequences from article files before preceding the training process. On the other hand, word embedding technique is working as an integral portion of the training process produces a dense and optimized feature representation that conserve the arrangements of sentences in the article files. This creates word embedding's more preferable to capture the article features when representing small parts of textual information's. This paper delivers certain insights for employing various techniques for articles feature representation and evaluating their importance on the performance of the proposed author identification model.

3.2 TF-IDF Representation for Articles

In author identification system, feature extraction process plays an important role to reduce the dimensionality of the input data to ensure the identification accuracy and enhance the time efficiency. In this author identification model, related features are extracted from the terms returned through the pre-processing phase utilizing a normalized TF-IDF method. Normalized TF-IDF is a vector space method that simply extracts the weight of terms numerically. Here, TF-IDF is integrated due to its highly precise performance when compared to other statistical methods. For every term i, the weight is calculated as follows:

$$W_i = \frac{\left(TF_i \times \log\left(\frac{N}{n_i}\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(TF_i \times \log\left(\frac{N}{n_i}\right)\right)^2}} \quad (1)$$

Where n_i is the number of article comprising term i and N is the total number of article. TF represents the number of occurrence of each term in a article, while IDF denotes the length normalization. A weight-term matrix with articles creating the rows and TF-IDF weights creating the columns through computing the TF-IDF for each feature, where w_{ij} is the weight of term i in article j, R is a sample article, T represents a term.

$$\begin{bmatrix} & T_1 & T_2 & \cdots & T_i \\ R_1 & w_{11} & w_{12} & \cdots & w_{1i} \\ R_2 & w_{21} & w_{22} & \cdots & w_{2i} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_j & w_{j1} & w_{j2} & \cdots & w_{ij} \end{bmatrix} \quad (2)$$

The pseudo code for the feature extraction process is illustrated in Algorithm 1. The related-term matrix features acquired at this stage are fed into the hybrid Bi-LSTM based author identification module.

Algorithm 1: TF-IDF based Feature Extraction

Input

T: the distinct terms in all reviews

R: the reviews of the training set

Output: weight Matrix

Step:

1: **for** each term $t_i \in T$ **do**

2: **for** each review $r_j \in R$ **do**

3: $w_{ij} = frequency\ of\ trem\ t_i\ in\ review\ r_j$

4: **end for**

5: **end for**

3.3 Create Word Embedding Matrix for Articles

In word embedding model the input is the article sentence S, this specific sentence comprises L words $S = \{w_1, w_2, ..., w_L\}$, the real-valued vector is used to represent each word. For every sentence create the matrix $W \in \Re^{d^w |V|}$, where, the sizes of vocabulary and word vectors are denoted as V and d^w. Here, the word to vec model is trained in the pre-training matrix for representing the matrix W. Further, each sentence is represented as word vectors $W_{emb} = \{e_1, e_2, ..., e_L\}$.

3.4 Middle Layer Feature Extraction

The CNN model can learn the features and differentiate them automatically and hence CNN does not need hand crafted features which reduces labor effort, time and excludes the requirement of prior knowledge. The features extracted in the middle layer have a supreme generalized capability and it acts as a representative. The extraction of features from the last layer is lesser superior to the middle layer. In this paper, we intend to examine the features obtained from middle layers significantly support to resolve the author identification problem in Marathi script's. As we know, it is not a simple process to train a deep CNN with large dataset. Though, the recent studies on CNN exhibits that the CNN trained on large scale text has strong interpretation capability which cloud be utilized to extract features to support text classification problem [22], [23].In this paper, we mainly employ the pre-trained deep CNN based word embedding model to extract the intrinsic middle layer feature representation to achieve a better classification result. The research paper [24] provided use of statics for Marathi text analysis. [25] Presented Marathi text summarization by NLP. [26], [27] Researchers presented comprehensive review on sentiment analysis for Indian regional languages.

4. Proposed Model for Author Identification

Researches in the field of author identification of Marathi texts are still limited. For that, this paper proposed an approach for identifying the author of Marathi articles based on deep learning. Deep learning has been successfully applied to various natural language processing tasks producing performance results beating previously state of the art technique. The architecture of the proposed model is shown below.

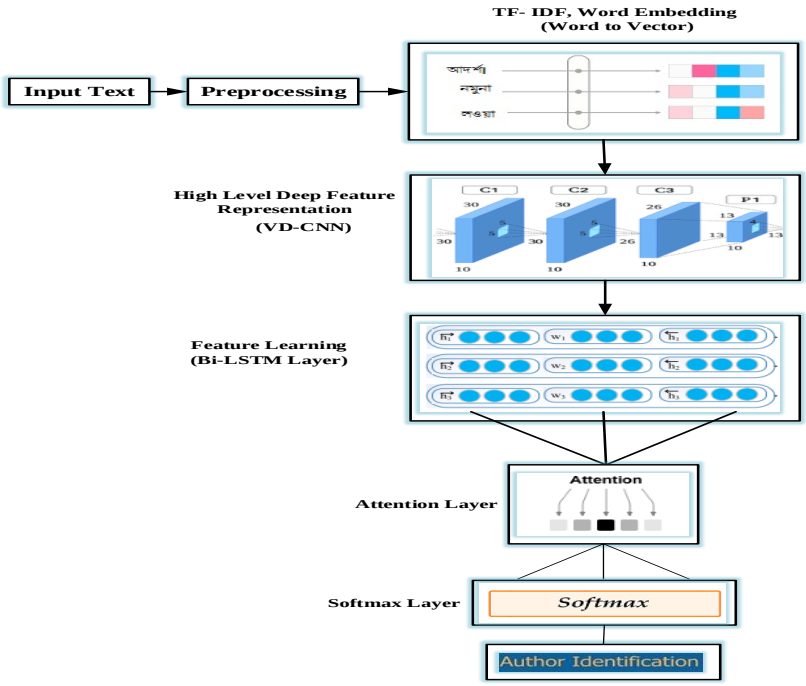

Figure 1: Overview of the proposed approach

The above figure 1, showing the overview of our proposed method. Here, we are initializing three Marathi text datasets with different genres. After getting the input, the articles will be pre-processed by performing three operations namely, tokenization, stop word removal, and stemming. Then, the TF-IDF and word embedding models are employed to obtain the word to vector matrix. Further, the vector-

matrix of the particular article sentence is fed into the VD-CNN model to obtain higher-level feature representation. Lastly, the authors of the articles find out with the high-level features using Bi-LSTM with attention-based softmax classifier.

4.1 Very Deep CNN Model for Intrinsic Feature Representation

In our work, we utilized pre-trained models for the process of middle layer feature extraction. The low-level information is represented by the lower layer from the extracted features each character of the word. The top layers can able to extract the features with the rich semantic behavior of a specific sentence. Therefore, the proposed model can accomplish the effective outcomes of classification by using the features from the middle layer to the top layer of CNN. VD-CNN (Very Deep-Convolutional Neural Network) is a convolutional neural network model for Large-Scale Feature representation. It makes the by replacing large kernel-sized filters with multiple 3×3 kernel-sized filters one after another. The structure of VD-CNN Model is shown in Figure2. For the word embedding matrix, the input vectors are initially convolved to obtaining the convolution vectors of the first layer and then move to the second convolution blocks. In the two convolutional blocks, a total of two max-pooling layers are presented in it, in addition to this, the ReLU activation function and the Batch Normalization layer are also available in these two blocks. The feature description of the VD-CNN model is tabulated in Table 1.

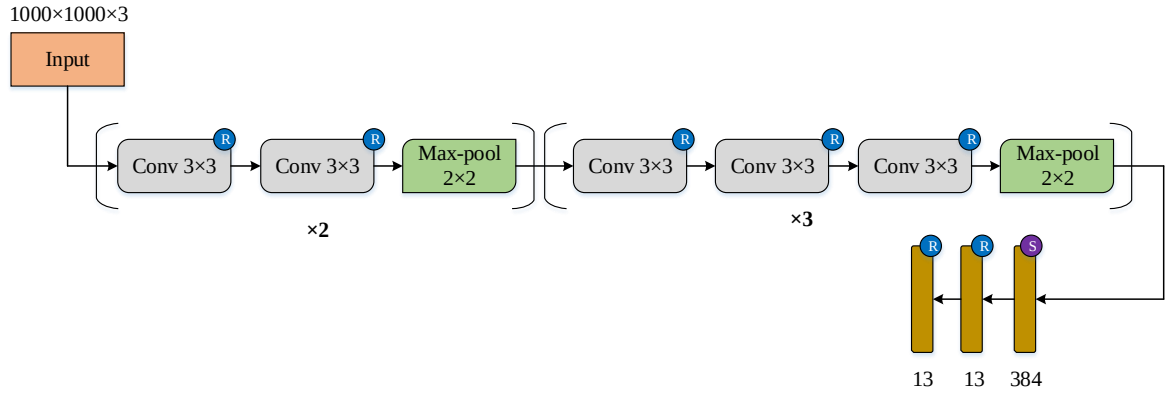

Figure 2: Proposed VD-CNN Model

Table 1: Features description of VD-CNN Model

Layer Name	Kernel Size	Size of Feature Map	Filter Size	Number of Strides	Activation Function	Number of Padding
Input Layer		1000×1000×3				
Conv1	96	500×500×96	15×15	2	ReLu	0
Max pool	1	250×250×96	2×2	2	-	0
Conv2	128	125×125×128	5×5	1	ReLu	0
Conv3	96	125×125×96	3×3	2	ReLu	0
Max pool	1	61×61×96	2×2	2	-	0
Conv4	128	29×29×128	3×3	2	ReLu	0
Conv5	384	29×29×384	1×1	1	ReLu	0
Avg pool	1	13×13×384	2×2	2	-	0

4.1.2 Complexity Analysis

To conduct the complexity analysis of VD-CNN Model, assume that the number of iterations is d , the number of samples per input sentence is p, the number of words in a sentence is r , the word embedding vector size is s, the convolution filter size is k, and the number of max pooling function is n. The model executes the inputted sentences with a time complexity of $O(r*n(2s*\alpha^2+k-1))$. Hence, the time complexity of the VD-CNN model can be defined as $O(\alpha^2*d*p*n*s*r)$.

4.2 Author Identification using Hybridized Bi-LSTM Model

The author identification process is carried out by one of the deep learning techniques called Bi-LSTM network also it is hybridized with an attention mechanism. Here VD-CNN yielded high-level feature matrix is selected as input vector, it includes the deep features of specific article sentences. For each point in the input sequence of the output layer, the entire context information about the past and the future are provided by the structure of Bi-LSTM. In this, μ denotes the hyperbolic tangent function and sigmoid function, o_t, i_t and f_t represents the output, input and the forget gates, x_t denotes the current input, h_t and h_{t-1} represents the output of the current and previous time LSTM network, at the memory unit, the state values for both the previous and the current time are denoted as C_{t-1} and C_t. Figure 3 demonstrates the cell structure of the LSTM.

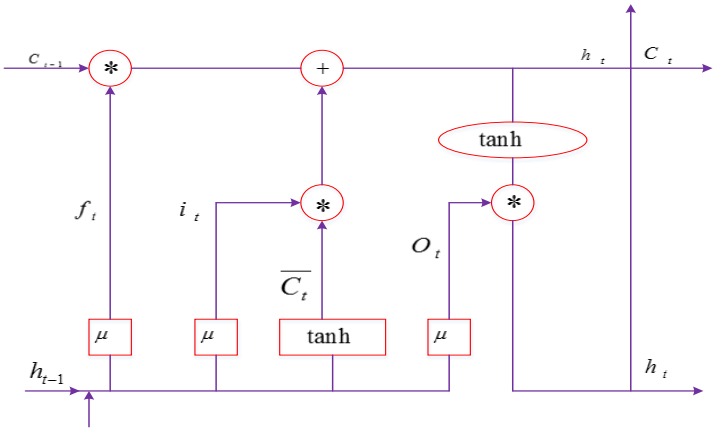

Figure 3: LSTM cell structure

The cell calculation process is achieved by using the below formulas (3) to (8). In this LSTM cell structure, at the previous movement, according to the cell outcome h_{t-1}, the input gate acts as input at a time t. At the current moment, the input x_t determines whether the current information is updated into the cell through the calculation. Then, the memory cell value of the current candidate is determined by the LSTM hidden layer cell output results and the current input data at the previous moment. The current candidate cell $\overline{C_t}$ and its own state C_{t-1} and the forget gate and input gate in the current moment helps to adjust the state value of the memory cell C_t. The state value of the memory cell is controlled by the output gate O_t and h_t is the output of the last cell that can be expressed in equation (8). The element-wise matrix multiplication is represented by the Character $*$ W and b is the weight and bias of neurons both are obtained through training.

$$i_t = Sigmoid\,(W_i.[h_{t-1}, x_t]+b_i) \quad (3)$$

$$f_t = Sigmoid\,(W_f.[h_{t-1}, x_t]+b_f) \quad (4)$$

$$\overline{C_t} = \tanh(W_C.[h_{t-1}, x_t]+b_c) \quad (5)$$

$$C_t = f_t * C_{t-1} + i_t * \overline{c_t} \quad (6)$$

$$O_t = Sigmoid\,(W_O.[h_{t-1}, x_t]+b_o) \quad (7)$$

$$h_t = O_t * \tanh(c_t) \quad (8)$$

The sequence data is processed by standard LSTM cell. The future context information is often ignored from the processing of data in time series. Each and every sequence of training consist of backward and forward LSTM neural network layers this is the basic idea of Bi-LSTM. The structure of Bi-LSTM is shown in figure 4.

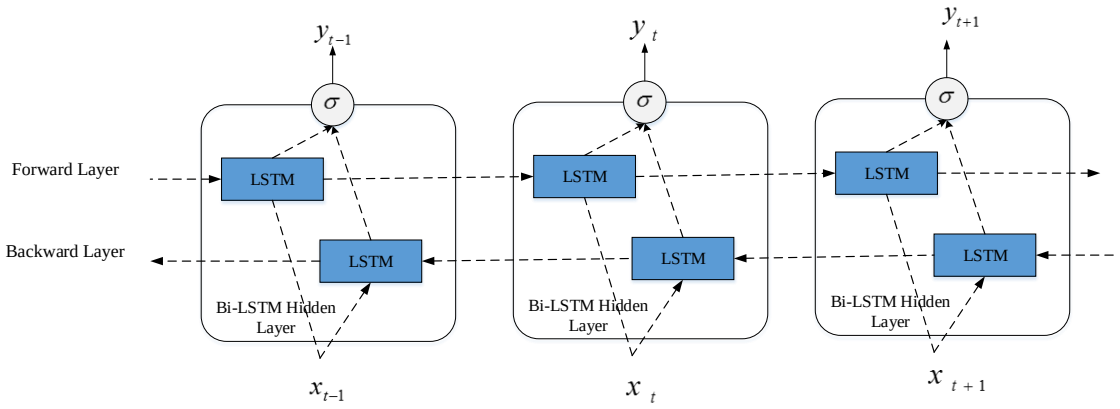

Figure 4: Bi-LSTM Structure

The speech signals are oppositely encoded with the help of the backward LSTM layers and directly encoded (starting to end) by the layers of forward LSTM. The weighted summation of both the forward and the backward hidden layer states($\overrightarrow{h_t}$), helps to determine the Bi-LSTM hidden layer state at the time t and they are specified as follows,

$$\overrightarrow{h_t} = LSTM\ (x_t, \overrightarrow{h_{t-1}}) \quad (9)$$

$$\overleftarrow{h_t} = LSTM\ (x_t, \overleftarrow{h_{t-1}}) \quad (10)$$

$$H_t = w_t \overrightarrow{h_t} + v_t \overleftarrow{h_t} + b_t \quad (11)$$

Where, v_t, w_t represents the weight to the state of backward and the forward hidden layer ($\overleftarrow{h_t}$) corresponding to the hidden layer state of Bi-LSTM at the time t and b_t represents the bias corresponding to the hidden layer.

4.2.1 Bi-LSTM with Attention Mechanism

In the upper layer, the output vector which activated by the layer of Bi-LSTM and it is the attention mechanism layer input. The Attention Mechanism is expressed as follows,

$$u_t = \tan h(W_w H_t + b_w) \quad (12)$$

$$a_t = \frac{\exp u_t^T u_w}{\sum_t \exp u_t^T u_w} \quad (13)$$

$$v_t = \sum_t a_t H_t \quad (14)$$

Where H_t is the upper layer output vector, energy value determined by H_t is denoted as u_t, bias coefficient is represented as b_w and weight coefficient is represented as W_w. In the new hidden layer state, at each hidden layer, the specific gravity of the weighting coefficient is defined as a_t. The attention matrix is represented as u_w which indicating the random initialization during the process of training and the output vector is denoted as the term v_t. The activation function is used to add nonlinear factors because of the presence of inadequate expressive power in the linear model. In neural networks, the most commonly used activation function is a Relu function.

4.2.2 Softmax Function for Calculating Identification Score

In this setting, for a high sensitive feature vector S we use a softmax classifier to predict label yˆ from a discrete set of classes Y. The input of classifier is a hidden state that is represented as h^*

$$\hat{p}(y/S) = soft\max(W^{(s)} h^* + b^{(s)}) \quad (15)$$

$$\hat{y} = \arg\max_y \hat{p}(y/S) \quad (16)$$

The hidden layer units are used as a sigmoid activation function. The softmax layer is also known as the output layer with output nods corresponding to the list of authors. The bias vector is represented as b, w represents the weight matrix and the target class is represented as Y. In the output layer, corresponding to the input vector x, the i^{th} node output is given as,

$$P(Y = i/x, W, b) = soft\max_i (W_x + b) \quad (17)$$

$$= \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x} + b_j} \quad (18)$$

Then, evaluate the predicted author $y\ pred$,

$$y_{pred} = \arg\max_i P(Y = i/x, W, b) \quad (19)$$

The highest score corresponding to the label evaluated in (4) can be obtained using

$$y_{\max} = \max_i P(Y = i/x, W, b) \quad (20)$$

5. Simulation Results and Discussion

This section describes the dataset, performance metrics, and results obtained on conducted simulations. The simulation is conducted with the help of the Python platform in view of two scenarios with the collected corpus. The simulation on the first scenario is taking place without the attention mechanism

and in the second scenario, the simulation is conducted with an attention mechanism. The results obtained in two different scenarios are shown under the result part.

5.1 Dataset

Various Marathi articles are utilized as dataset, collected fromhttp://tdil-dc.in/ [28]; additionally from several sites that available for free. These collected different articles are also called as corpus. The articles are separated into three datasets in different categories. The first dataset (DS-1) includes 8comedy articles, second dataset (DS-2) includes 9 Historical articles, and third dataset (DS-3) includes 8mixed articles. The details of datasets are given belowin table 2.

Table 2: Dataset description

DS-1 (Comedy articles)		DS-2 (Historical articles)		DS-3 (Mixed articles)	
S.No.	Author Name	S.No.	Author Name	S.No.	Author Name
1	Charudutta R	1	Dr. Bho R Soap	1	Anandini
2	DhananjayAwale	2	Dr. Vs Ba Prabhudesai	2	LalitaPreeti
3	Befikir	3	Hemant Vishnu Inamdar	3	VidyaBhutkar
4	AniketYewale	4	ChandramohanHangekar	4	ChaitanyaRaskar
5	Kiran	5	Mohini Verde	5	Javherganj
6	DhananjayDiwan	6	MrinaliniJoglekar	6	SammerGaikawad
7	VinayKhandagale	7	Mathav T. Patwardhan	7	MohanaJogalekar
8	SatyajeetKharkar	8	Swati SuhasKarve	8	SaiKeskar
-	-	9	VinayaKhadpekar	-	-

5.2 Model Training

In this paper, an end to end type of Bi-LSTM model with neural attention method is employed to explore the process of author identification through high level feature representation yielded from VD-CNN model. In order to validate the effectiveness of the proposed model, we employ two kinds of word representation individually inputted to VD-CNN to obtain high level feature representation. Further, the high level feature is fed into the hybridized Bi-LSTM model to correctly identify the author of particular article. The proposed model comprises of three improved Bi-LSTM layers, an attention layer and final softmax classification layer. The related model parameter setting is listed in Table 3. In every iteration d , the high level word representations are inputted to the Bi-LSTM layer. The Bi-LSTM layer follows our attention weighting process which is considered as an output. At last, the index of attention layer output is fed into the softmax classification layer to identify the correct author of specific articles.

Table 3: Parameter setting of proposed Bi-LSTM Model

Parameter	**Value**
Number of Bi-LSTM layers	3
Input layer size	40
Hidden layer size	80
Output layer size	20
Learning rate	0.002

Batch size	8
Decay factor	0.2
Decay period	5
Optimizer	Adam

The learning rate is initially set to 0.002, the Adam gradient descent optimization model is employed to adjust the hyperparametrs during model training. The batch size is set to 8, and the state and hyper parameters in the proposed model are marginally adjusted on the testing process for correct identification.

5.3 Simulation Results

The author identification process is conducted with 3 different data sets in two scenarios. Every dataset contains 8-10 articles. In this section, the results obtained in both scenarios with the proposed method are described.

5.3.1 Results Obtained for Three Datasets on Scenario-1

Performance tables of scenario-1

Table 4: Author wise results obtained on DS-1in scenario-1

Scenario-1 - Results obtained on DS-1												
S. No	Name of Author	RNN				CNN				CNN+Bi-LSTM			
		Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.
1	Charudatta	96.57	78.75	90.00	84.00	98.71	97.22	100	98.59	97.35	97.43	99.24	98.32
2	Dhananjay A	92.71	59.05	88.57	70.86	97.00	94.37	95.71	95.04	98.03	96.26	97.07	96.66
3	Befikir	98.14	91.30	90.00	90.65	97.71	94.20	92.86	93.53	99.12	98.78	98.21	98.49
4	Amit Yewale	95.86	78.08	81.43	79.72	96.29	98.51	94.29	96.35	99.34	98.54	97.74	98.13
5	Kiran	99.43	100	94.29	97.06	98.29	95.77	97.14	96.45	97.34	98.05	98.32	98.18
6	Dhananjay D	96.71	87.30	78.57	82.71	96.57	83.82	81.43	82.61	97.45	97.36	97.79	97.57
7	Vinay K	95.86	97.67	60.00	74.34	98.50	92.86	92.86	92.86	98.19	98.01	99.06	98.53
8	Satyajeet	99.71	100	97.14	98.55	97.43	98.53	95.71	97.10	99.34	96.76	98.08	97.41

In table 4, the results of different articles of DS-1 in the first scenario is described for two existing (RNN and CNN) methods and proposed CNN and Bi-LSTM based method. Dynamic performance

values are obtained for all articles available on our DS-1. The experiment conducted with DS-1, the proposed approach performed well when compared with other existing methods.

Table 5: Author wise results obtained on DS-2in scenario-1

Scenario-1 Results obtained on DS-2												
S. No	Name of Author	RNN				CNN				CNN+Bi-LSTM			
		Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.
1	Bho R Soap	94.47	83.08	83.47	83.27	96.73	97.28	93.14	95.16	99.36	98.73	94.48	96.55
2	Prabhudesai	98.28	95.79	92.05	93.88	98.14	95.42	94.28	94.84	97.74	99.03	97.46	98.23
3	Hemant	97.26	94.48	91.63	93.03	99.85	96.20	99.25	97.70	94.26	97.46	100	98.71
4	Hangekar	94.84	64.47	87.39	74.20	96.35	95.42	96.34	95.87	99.85	96.76	97.72	97.23
5	Mohini	98.91	90.42	93.28	91.82	94.83	84.67	80.37	82.46	96.49	97.03	99.74	98.36
6	Mrinalini	95.89	93.23	92.58	92.90	96.54	94.74	91.30	92.98	98.62	100	98.35	99.16
7	Mathav	99.20	94.73	96.66	95.68	96.85	97.35	93.67	95.47	97.74	96.37	99.62	97.96
8	Swati	97.41	93.48	92.57	93.02	96.48	96.82	95.87	96.34	100	98.86	98.73	98.79
9	Vinaya	96.29	96.78	93.75	95.24	97.85	93.67	98.34	95.94	98.06	99.49	98.74	99.11

In table 5, the results obtained for the DS-2 in the first scenario are described. The proposed method achieved better performance than other existing methods.

Table 6: Author wise results obtained on DS-3 in scenario-1

Scenario-1 Results obtained on DS-3												
S. No	Name of Author	RNN				CNN				CNN+Bi-LSTM			
		Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.
1	Anandini	96.57	89.66	74.29	81.25	97.43	83.33	92.86	87.84	98.67	100	99.47	99.73
2	LalitaP	96.00	100	90.00	94.74	96.57	98.55	97.14	97.84	96.64	99.33	98.33	98.82
3	Vidya B	98.43	86.42	100	92.72	97.43	83.33	92.86	87.84	99.21	100	100	100
4	Chaitanya R	97.57	100	95.71	97.81	98.14	92.54	88.57	90.51	97.45	98.75	97.34	98.05

5	Javherganj	97.43	98.53	95.71	97.10	96.14	98.48	92.86	95.59	97.33	96.28	98	97.13
6	Sameer G.	98.27	100	94.29	97.06	97.29	95.77	97.14	96.45	99.45	100	100	100
7	Mohana J.	96.57	98.55	97.14	97.84	98.71	90.67	97.14	93.79	99.64	95.12	97.06	96.58
8	Sai K	96.00	97.01	92.86	94.89	97.86	91.04	87.14	89.05	98.37	97.55	98.56	98.05

In table 6, the results of DS-3 in the first scenario are described for the existing and proposed methods. Table 5, shows that the proposed approach beaten the performance of the existing method. Various values obtained with different methods on several articles is shown in above table 5.

Table 7: Average of results obtained on a different dataset in scenario-1

Scenario-1 Average results obtained on three different datasets													
S. No	Algorithms	DS-1				DS-2				DS-3			
		Acc.	Pre.	Rec.	F-m	Acc.	Pre.	Rec.	F-m.	Acc.	Pre.	Rec.	F-m.
1	RNN	96.87	86.51	85	84.73	96.95	89.60	91.48	90.33	96.97	96.27	92.57	94.17
2	CNN	97.62	94.41	93.97	94.06	97.06	94.61	93.61	94.08	97.44	91.71	93.21	92.36
3	CNN+Bi-LSTM	98.27	97.66	98.11	97.91	98.01	98.19	98.31	98.23	98.34	98.37	98.36	98.41

The average results of every dataset are shown in table 7. There are four different performance parameters are used for comparing the average results (shown in table 7). The CNN guided Bi-LSTM achieved up to 98% accuracy, which is better when compared with the existing systems. Other metrics also performed well and got a maximum rate than the existing systems.

Performance graphs of scenario-1

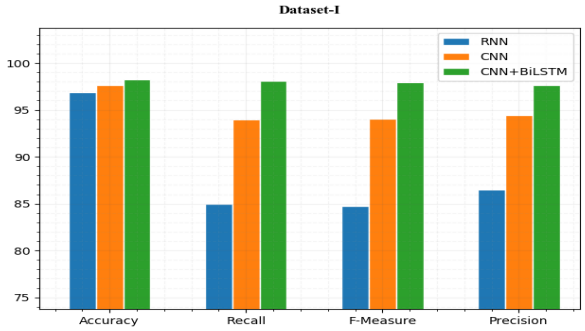

Figure 5: Average of Accuracy, Precision, Recall, and F-measure on DS-1 in scenario-1

On figure 5, the average values given in table 6 are plotted for the better observation. The performance of the proposed and existing methods for DS-1 in the first scenario is clearly shown in figure 5. The proposed CNN guided Bi-LSTM based method got a performance rate of above 98% except for Precision. It is much better when compared with the other two methods (RNN and CNN).

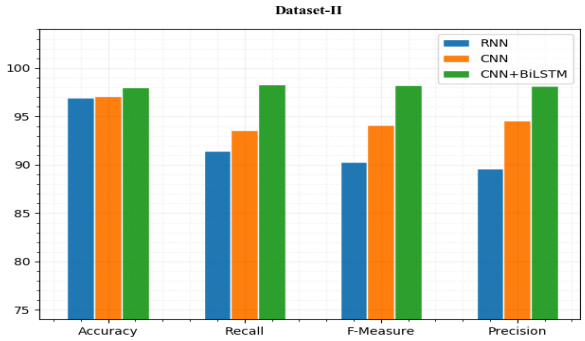

Figure 6: Average of Accuracy, Precision, Recall, and F-measure on DS-2 in scenario-1

In DS-2, the proposed method got average values of 98.30, 98.01, 98.19, 98.31, and 98.02 on Accuracy, Precision, Recall, and F-measure. This is shown in table 6 and graphical representation is shown in figure 6.

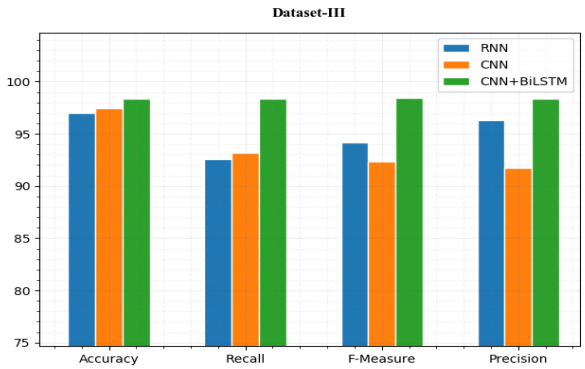

Figure 7: Average of Accuracy, Precision, Recall, and F-measure on DS-3in scenario-1

The average values of Accuracy, Precision, Recall, and F-measure on DS-3 in the first scenario are shown in table 6. In DS-3, the proposed method earns better results when comparing with other methods and it is shown in above figure 7.

5.3.2 Results obtained for three datasets on scenario-2

Performance Tables of Scenario-2

Table 8: Author wise results obtained on DS-1in scenario-2

Scenario-2 - Results obtained on DS-1													
S. No	Name of Author	RNN				CNN				CNN+Bi-LSTM			
		Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.
1	Charudatta	96.84	81.35	93.62	87.05	99.43	98.62	100	99.30	98.64	98.74	99.48	99.10
2	Dhananjay A	93.76	67.55	91.89	77.86	98.64	96.15	97.26	96.70	99.83	99.36	99.17	99.26
3	Befikir	98.93	92.48	94.75	93.60	95.31	95.59	93.64	94.60	99.37	99.42	99.73	99.57
4	Amit Yewale	96.53	78.08	85.84	81.77	98.09	98.76	96.64	97.68	99.46	99.24	98.84	99.03
5	Kiran	99.64	99.36	95.59	97.43	98.54	96.56	98.83	97.68	98.39	98.83	99.47	99.14
6	Dhananjay D	97.51	89.82	83.54	86.56	97.33	86.48	86.92	86.69	97.97	99.42	98.86	99.13
7	Vinay K	96.46	97.67	64.10	77.40	98.84	93.82	94.42	94.11	98.57	99.52	99.41	99.46
8	Satyajeet	99.86	100	98.27	99.12	99.13	98.83	96.96	97.88	100	98.85	97.93	98.38

Table 8 showing the various performance rate achieved by every author on three different methods in the second scenario. In the second scenario, Adam optimization is included to enhance the performance of the system. With Adam, the performance of all the methods is slightly increased. The CNN guided Bi-LSTM based method achieved 99% accuracy in most of the documents.

Table 9: Author wise results obtained on DS-2in scenario-2

Scenario-2 Results obtained on DS-2													
S. No	Name of Author	RNN				CNN				CNN+Bi-LSTM			
		Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.
1	Bho R Soap	96.64	88.26	84.57	86.37	97.47	99.01	94.60	96.75	99.54	99.06	98.65	98.85
2	Prabhudesai	97.79	96.32	94.85	95.57	99.87	96.94	95.26	96.09	98.02	99.24	98.75	98.99
3	Hemant	98.86	95.68	93.70	94.67	99.89	97.64	97.54	97.58	99.63	98.84	99.63	99.23
4	Hangekar	95.63	72.74	89.03	80.06	97.94	99.03	98.50	98.76	100	100	98.92	99.45

5	Mohini	97.38	92.02	95.47	93.71	98.54	87.38	87.41	87.39	99.28	98.73	99.50	99.11
6	Mrinalini	96.74	94.77	93.68	94.22	97.67	95.36	94.05	94.70	98.89	99.79	98.86	99.32
7	Mathav	99.43	93.84	97.58	95.67	98.06	98.73	95.83	97.25	99.42	98.73	99.00	98.86
8	Swati	98.73	97.68	95.84	96.75	97.59	97.47	96.37	96.91	100	99.03	99.16	99.09
9	Vinaya	98.08	98.52	92.90	95.62	99.42	94.84	99.63	97.17	98.86	99.74	99.59	99.66

The performance of DS-2 in the second scenario is shown in table 9. Here also the CNN based Bi-LSTM model achieved better results and two of the documents achieved 100% accuracy. Different performance values achieved for the authors of DS-2, the CNN guided Bi-LSTM based proposed method performed much better than the existing methods.

Table 10: Author wise results obtained on DS-3in scenario-2

Scenario-2 Results obtained on DS-3												
S. No	Name of Author	RNN				CNN				CNN+Bi-LSTM			
		Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.	Acc.	Prec.	Rec.	F-m.
1	Anandini	98.73	93.47	76.36	84.05	97.99	86.33	93.89	89.95	99.24	100	99.27	99.63
2	Lalita P	98.24	100	91.94	95.80	98.78	97.76	98.73	98.24	98.78	99.26	99.80	99.52
3	Vidya B	99.73	88.09	98.73	93.10	97.64	85.30	94.31	89.57	99.87	100	100	100
4	Chaitanya R	98.37	99.48	96.78	98.11	98.79	93.69	90.97	92.30	99.15	98.94	98.95	98.94
5	Javherganj	98.63	98.53	96.63	97.57	99.32	99.00	93.48	96.16	98.60	99.20	99.05	99.12
6	Sameer G.	96.55	99.75	96.88	98.29	98.56	96.91	98.90	97.89	100	100	100	100
7	Mohana J.	97.73	99.04	97.97	98.50	99.06	91.99	99.20	95.45	99.24	98.91	99.16	99.03
8	Sai K	97.48	98.24	93.90	96.02	98.61	93.27	89.73	91.46	99.62	98.88	99.49	99.18

The table 10 shows the performance values achieved for every author of DS-3 in the second scenario. Varied results are obtained on all methods, the CNN based Bi-LSTM model got better performance than the previous methods.

Table 11: Average of results obtained on a different dataset in scenario-2

Scenario-2 Average results obtained on three different datasets												
S. No	Algorithms	DS-1				DS-2				DS-3			
		Acc.	Pre.	Rec.	F-m	Acc.	Pre.	Rec.	F-m.	Acc.	Pre.	Rec.	F-m.
1	RNN	97.44	88.28	88.45	87.59	97.69	92.20	93.06	92.51	98.18	97.07	93.64	95.18
2	CNN	98.16	95.60	95.58	95.58	98.49	96.26	95.46	95.84	98.59	93.03	94.90	93.87
3	CNN+Bi-LSTM	99.02	99.17	99.11	99.13	99.29	99.24	99.11	99.17	99.31	99.39	99.46	99.42

Table 11 shows the average result of all datasets. The proposed CNN based Bi-LSTM model got an average of 99% on all parameters. Other existing systems are getting varied performance on every dataset. When comparing it with the CNN based Bi-LSTM model based method, which is lower.

Performance Graphs of Scenario-2

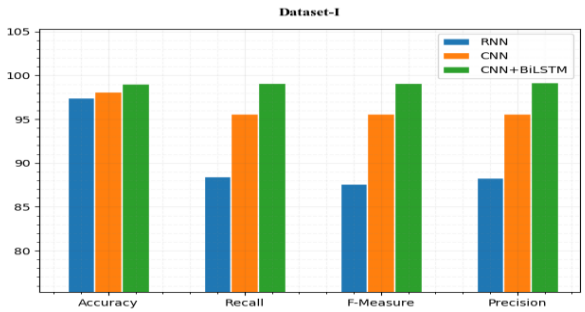

Figure 8: Average of Accuracy, Precision, Recall, and F-measure on DS-1in scenario-2

The above figure 8 describes the average performance of all techniques with dataset 1 in second scenarios in a graphical manner. The values shown in table 10 are plotted as a graph in figure 8. From figure 8, the higher the rate achieved by the proposed CNN based Bi-LSTM model is clearly represented.

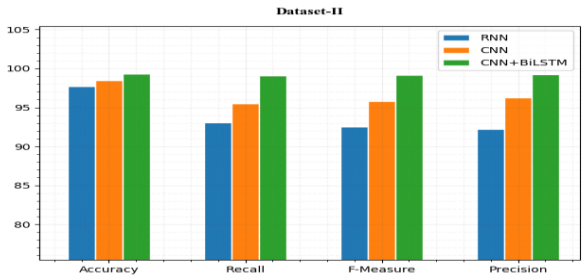

Figure 9: Average of Accuracy, Precision, Recall, and F-measure on DS-2in scenario-2

The average performance achieved in DS-2 in the second scenario is described in figure 9. Here the CNN based Bi-LSTM model based technique got better performance than the existing system. (The performance values of DS-2 is shown in table 10).

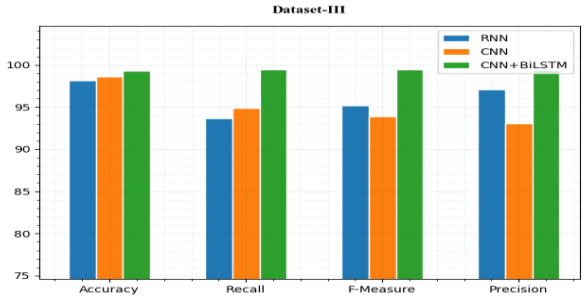

Figure 10: Average of Accuracy, Precision, Recall, and F-measure on DS-3in scenario-2

The figure 10, shows the average values achieved by DS-3 in the second scenario. The performance of the CNN based Bi-LSTM model based system got enhanced performance on all datasets, which are shown in Figures 8, 9, and 10.

5.3.3 Overall Performance of System

Here the global performance of CNN based Bi-LSTM model, utilizing the optimization and without utilizing the optimization algorithm is discussed.

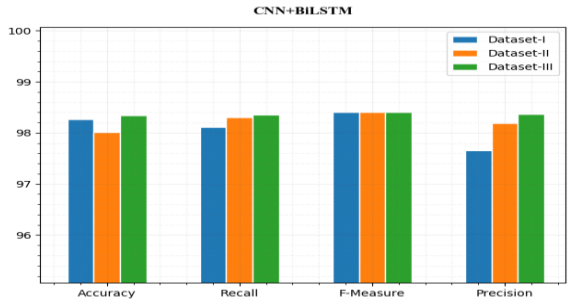

Figure 11: Overall performance without Adam

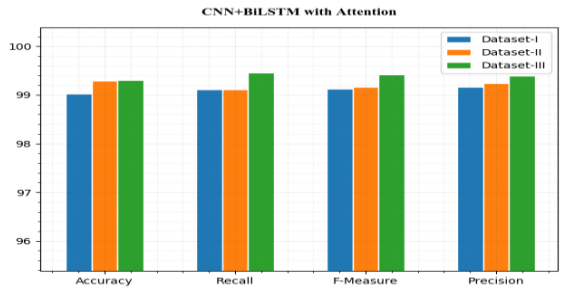

Figure 12: Overall performance with Adam

From the above figures 11 and 12, the proposed CNN based Bi-LSTM model earned up to 98% of performance on overall datasets without the use of optimization algorithm. It earned more than 99% on all compared performance by using the optimization algorithm.

Differed values are achieved in every dataset and at last, the CNN based Bi-LSTM model outperformed in all datasets with and without the optimization algorithm. Because of the different contents of the Marathi corpus, all the techniques are getting a different performance.

Table 12: Performance improvements for all algorithms with two word representations

Algorithms	Average Accuracy (TF-IDF)	Average Accuracy (Word Embedding)	Performance Improvement	Time of Training(S)
RNN	95.31	97.77	2.46	980
CNN	95.46	98.41	2.95	835
CNN+Bi-LSTM	96.01	99.16	3.15	880

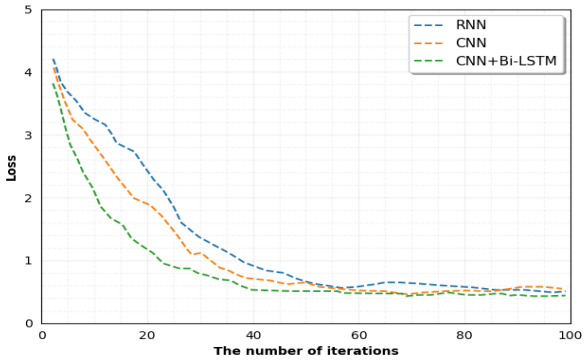

Figure 13: The convergence curves over iterations on the Marathi Language Text Corpus

The average accuracy values for the all algorithm with respect to two word representation such as TF-IDF and Word Embedding are shown in Table 12. The Word Embedding representation achieves better accuracy compared to TF-IDF based word representation for all algorithm. Moreover, the proposed model has been achieved better performance improvement compared to other algorithms. The convergence curves on the Marathi Language Text Corpus dataset has been displayed in figure 13. From the convergence curve graph, we can see that our model has better performance than the baseline CNN and RNN models. It has faster convergence speed in the training process due to high level feature representation.

6. Conclusion

In this paper, the development of an efficient model based on deep learning for identifying the authors of Marathi articles has been proposed. The author identification is a long exploited field that has been applied to multiple applications for the identification of the original author of articles for copy right violations. The model proposed in this article, the various process is held for identifying the right authors. Initially, the unknown articles are initialized into the system and pre-processed. At the pre-processing stage, processes such as tokenization, stop word removal, and stemming are carried out. After completing the pre-processing stage, the deep level features are extracted using VD-CNN model which provides higher level feature representation supports to enhance the identification performance. At last, the articles are classified using the proposed hybridized BI-LSTM based Deep learning model. The results obtained from our simulation show that the proposed model is preforming more efficiently than the existing models.

Refrences

[1] Alsmearat K, Al-Ayyoub M, Al-Shalabi R and Kanaan G (2017) Author gender identification from Arabic text. Journal of Information Security and Applications 35: 85-95.
[2] Srinivasan L and Nalini C (2017) An improved framework for authorship identification in online messages. Cluster Computing, 1-10.
[3] Pandian A, Ramalingam VV and Preet RPV (2016) Authorship Identification for Tamil Classical Poem (MukkoodarPallu) using C4. 5 Algorithm. Indian Journal of Science and Technology 9(47).
[4] Zhao C, Song W, Liu L, Du C and Zhao X (2017) Research on Author Identification Based on Deep Syntactic Features. In Computational Intelligence and Design (ISCID), 2017 10th International Symposium on, IEEE, 1: 276-279.
[5] Rexha A, Kröll M, Ziak H and Kern R (2018) Authorship identification of documents with high content similarity. Scientometrics 115(1): 223-237.

[6] Lu W, Huang Y, Bu Y and Cheng Q (2018) Functional structure identification of scientific documents in computer science. Scientometrics 115(1): 463-486.
[7] Otoom FA, Abdullah EE, Jaafer S, Hamdallh A and Amer D (2014) Towards author identification of Arabic text articles. In Information and Communication Systems (ICICS), 2014 5th International Conference on IEEE, 1-4.
[8] Waijanya S and Promrit N (2017) The Poet Identification Using Convolutional Neural Networks. International Conference on Computing and Information Technology, Springer, Cham, 179-187.
[9] Mohsen AM, El-Makky NM and Ghanem N (2016) Author identification using deep learning. In Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on IEEE, 898-903.
[10] Ferilli S, Redavid D and Esposito F (2015) Unsupervised Author Identification and Characterization. In Italian Research Conference on Digital Libraries, Springer, Cham, 129-141.
[11] Potha N and Stamatatos E (2017) An Improved Impostors Method for Authorship Verification. In International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, Cham, 138-144.
[12] Albadarneh J, Talafha B, Al-Ayyoub M, Zaqaibeh B, Al-Smadi M, Jararweh Y and Benkhelifa E (2015) Using big data analytics for authorship authentication of Arabic tweets. In Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on IEEE, 448-452.
[13] Al-Ayyoub M, Jararweh Y, Rabab'ah A and Aldwairi M (2017) Feature extraction and selection for Arabic tweets authorship authentication. Journal of Ambient Intelligence and Humanized Computing 8(3): 383-393.
[14] Taghva K (2017) Name identification and extraction with formal concept analysis. International Journal of Machine Learning and Cybernetics 8(1): 171-178.
[15] Sanchez-Perez MA, Markov I, Gómez-Adorno H and Sidorov G (2017) Comparison of Character n-grams and Lexical Features on Author, Gender, and Language Variety Identification on the Same Spanish News Corpus. In International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, Cham, 145-151.
[16] Ferilli S (2016) A sentence structure-based approach to unsupervised author identification. Journal of Intelligent Information Systems 46(1): 1-19.
[17] Kale Sunil. Digamberrao and R. S. Prasad, "Author Identification on Literature in Different Languages: A Systematic Survey," 2018 International Conference On Advances in Communication and Computing Technology (ICACCT), Sangamner, 2018, pp. 174-181.
[18] Kale S.D., Prasad R.S. (2020) Influence of Language-Specific Features for Author Identification on Indian Literature in Marathi. In: Reddy V., Prasad V., Wang J., Reddy K. (eds) Soft Computing and Signal Processing. ICSCSP 2019. Advances in Intelligent Systems and Computing, vol 1118. Springer, Singapore.
[19] Sunil D. Kale, Rajesh S. Prasad, "Author Identification on Imbalanced Class Dataset of Indian Literature in Marathi," International Journal of Computer Sciences and Engineering, Vol.6, Issue.11, pp.542-547, 2018.
[20] Kale Sunil Digamberrao, Prasad R. S. Author identification using sequential minimal optimization with rule-based decision tree on Indian literature in Marathi. Procedia computer science. 2018 Jan 1;132:1086-101.
[21] Kale SD, Prasad R. S. A systematic review on author identification methods. International Journal of Rough Sets and Data Analysis (IJRSDA). 2017 Apr 1;4(2):81-91.
[22] Liang H, Sun X, Sun Y, Gao Y (2017). Text feature extraction based on deep learning: a review. EURASIP journal on wireless communications and networking. 1:1-12.
[23] Wang J, Li Y, Shan J, Bao J, Zong C, Zhao L(2019). Large-Scale Text Classification Using Scope-Based Convolutional Neural Network: A Deep Learning Approach. IEEE Access. 7:171548-171558.
[24] Amidwar S. Baxi S, Rao K, Kale S. "Text Analysis for Author Identification Using Machine Learning" Journal of Emerging Technologies and Innovative Research 2017:4(6):138-41
[25] Sunil D. Kale et.al, Marathi text summarization through NLP and deep learning mechanism, Journal of Autonomous Intelligence (2023) Volume 6 Issue 3 doi: 10.32629/jai.v6i3.1009 https://jai.front-sci.com/index.php/jai/article/view/1009
[26] Kale, S.D., Prasad, R., Potdar, G.P., Mahalle, P.N., Mane, D.T. and Upadhye, G.D. 2023. A Comprehensive Review of Sentiment Analysis on Indian Regional Languages: Techniques, Challenges, and Trends. International Journal on Recent and Innovation Trends in Computing and Communication. 11, 9s (Aug. 2023), 93–110. https://doi.org/10.17762/ijritcc.v11i9s.7401
[27] Sunil D. Kale, "Sentiment Analysis on Indian Regional Languages: A Comprehensive Review," International Journal of Computer Sciences and Engineering, Vol.7, Issue.1, pp.966-974, 2019.
[28] Sunil Digamberrao Kale and Rajesh S. Prasad, "Author wise Marathi Language Text Corpus", 2018, Indian Language Technology Proliferation and Deployment Centre, http://tdil-dc.in/index.php?option=com_download&task=showresourceDetails&toolid=2006&lang=en