IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 26, 2021, accepted June 28, 2021, date of publication July 9, 2021, date of current version July 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095967

Authorship Classification in a Resource Constraint Language Using Convolutional Neural Networks

MD. RAJIB HOSSAIN[1], **(Graduate Student Member, IEEE),**
MOHAMMED MOSHIUL HOQUE[1], **(Senior Member, IEEE),**
M. ALI AKBER DEWAN[2], **(Member, IEEE),**
NAZMUL SIDDIQUE[3], **(Senior Member, IEEE),**
MD. NAZMUL ISLAM[1], **(Graduate Student Member, IEEE),**
AND IQBAL H. SARKER[1], **(Member, IEEE)**

[1]Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong 4349, Bangladesh
[2]School of Computing and Information Systems, Faculty of Science and Technology, Athabasca University, Athabasca, AB T9S 3A3, Canada
[3]School of Computing, Engineering and Intelligent Systems, Ulster University, Londonderry BT47 7JL, U.K.

Corresponding author: Mohammed Moshiul Hoque (moshiul_240@cuet.ac.bd)

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2020-06080, and in part by the Establishment of Sheikh Kamal IT Business Incubator in CUET Project, BHTPA, ICT Division, Bangladesh.

ABSTRACT Authorship classification is a method of automatically determining the appropriate author of an unknown linguistic text. Although research on authorship classification has significantly progressed in high-resource languages, it is at a primitive stage in the realm of resource-constraint languages like Bengali. This paper presents an authorship classification approach made of Convolution Neural Networks (CNN) comprising four modules: embedding model generation, feature representation, classifier training and classifier testing. For this purpose, this work develops a new embedding corpus (named WEC) and a Bengali authorship classification corpus (called BACC-18), which are more robust in terms of authors' classes and unique words. Using three text embedding techniques (Word2Vec, GloVe and FastText) and combinations of different hyperparameters, 90 embedding models are created in this study. All the embedding models are assessed by intrinsic evaluators and those selected are the 9 best performing models out of 90 for the authorship classification. In total 36 classification models, including four classification models (CNN, LSTM, SVM, SGD) and three embedding techniques with 100, 200 and 250 embedding dimensions, are trained with optimized hyperparameters and tested on three benchmark datasets (BACC-18, BAAD16 and LD). Among the models, the optimized CNN with GloVe model achieved the highest classification accuracies of 93.45%, 95.02%, and 98.67% for the datasets BACC-18, BAAD16, and LD, respectively.

INDEX TERMS Natural language processing, authorship classification, resource constraint language, semantic feature extraction, deep learning.

I. INTRODUCTION

Authorship classification is a long-established research topic in Natural Language Processing (NLP) that deals with the difficulty of identifying the author against a particular text. Authorship classification is conducted mainly to verify the authorship of a particular text. The authorship classification's primary purpose is to infer the author of a document based on their distinct writing styles and features. For example, some authors may practice one or two following words more commonly than the other authors. Other authors may prefer to apply particular clauses, specific tense, distinguished sentence structure or open and close sentences with an appropriate grammatical constituent. These features can be used in identifying the authorship of a particular writing. Authorship classification is a well-established research topic for high resource languages (e.g., English and other European languages) due to the availability of authorship corpus, feature extractors and classification techniques. However, it is a challenging task for a low-resource language like Bengali due to the shortage of linguistic resources and techniques [1]. On the other hand, feature extraction and classification

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Shariq Imran.

techniques in high-resource languages cannot be directly applied to low-resource languages because of the variations of local dialects and structural divergences.

In recent years, authorship classification has attracted much attention from Bengali Language Processing (BLP) researchers. However, the primary difficulty of performing research on authorship classification in Bengali is the lack of linguistic resources in digital form, unavailability of necessary tools for processing language and inadequate corpora. Many prominent writers in Bengali literature can be discovered from their writing styles and variations. These properties can be utilized in literary, historical, social and cultural studies. There are no functional BLP tools to identify the anonymous author and plagiarism for Bengali texts. As a result, Fake news and textual forgeries in the Bengali language have rapidly increased on the digital platform. Thus, the Bengali authorship classification approach may assist in controlling the textual forgeries on the internet. Moreover, authorship classification can be used for author identification, identifying plagiarism, classification of authors of threats, computer forensics, author profiling and many more [2]. This can also be used for checking students' submissions especially in online learning.

There are several statistical and machine learning (ML) techniques that addressed the authorship attribution problem. The statistical methods are used for extracting stylometry features [2]–[5], character level embedding features [6], [7], and n-gram features [8], [9] for authorship attribution. Several ML techniques such as support vector machine (SVM), Naive Bayes (NB) [10]–[12], CNN [6], [13]–[15] are the most commonly used methods for authorship classification. Nevertheless, stylometry, character level, and n-gram feature extractors cannot capture the sentence and document level semantic and syntactic features [16]. On the other hand, SVM, NB, and decision tree-based classification models cannot carry the local and global classification features. This paper presents a deep learning technique to address the authorship classification problem in Bengali texts. The deep learning method offers automatic feature selection, selects the best classification model during training, alleviates model overfitting and underfitting problems, and represents the text's semantic features. The key contributions of the current research include:

- Development of a new embedding corpus consisting of 949, 062 unlabelled texts with 2, 744, 680 unique words and an authorship classification corpus containing 18 prominent authors with 25, 749 labelled texts which are more extensive than existing corpora (Sec. III).
- Generation of *ninety* embedding models for Bengali authorship attribution using a combination of three embedding techniques (Word2Vec, GloVe and FastText) selection of hyperparameters and evaluation their performance by the intrinsic evaluators to select suitable embedding for Bengali authorship classification (Sec. VII-A).
- Development of a CNN based authorship classification model by investigating a set of optimized hyperparameters with improve classification accuracy and complexity reduction (Secs. IV and V).
- Investigation of the performance of the proposed Bengali authorship attribution model on three standard datasets and its comparative performance analysis (Sec. VII).

II. RELATED WORK

Although authorship classification is a well-established research issue in well-resourced languages, it is in it's primitive stage to date in the realm of low-resource languages. This work deals with the authorship classification problem concerning a low-resource language, such as Bengali. Thus, past research on authorship classification can be broadly categorized into two ways: Non-Bengali language-based authorship classification and Bengali language based authorship classification.

A. NON-BENGALI LANGUAGE-BASED AUTHORSHIP CLASSIFICATION

There is enormous advancement in authorship classification research in high-resource languages such as English and other European languages. Moreover, research has been conducted in other languages, such as: Arabic, Latin, and Urdu. Tweedle *et al.* [17] applied the neural network technique with stylometry for authorship classification in English texts. A stylometry methodology is applied to the automatic analysis of English literary texts [3]. This work experimented with a corpus of five million words consisting of male and female authors. Ruder *et al.* [7] presented a character-level and multi-channel CNN for authorship attribution for English texts. Rocha *et al.* [18] studied different machine learning algorithms that worked well on small sample size. Yeang *et al.* [19] used a backpropagation based particle swarm operation to detect the author from the source code written in English. They have considered lexical, structural and syntactic feature metrics for detecting authors from 2, 022 java files and achieved an accuracy of 91.06%. Alsulami *et al.* [20] used Long short-term memory (LSTM) for source code authorship classification with 200 source files from 10 programmers and gained 85.00% accuracy. Enrique *et al.* [13] assessed the quality of neurally generated English text using an established authorship identification. Koppel *et al.* [21] used a naive similarity-based techniques for authorship classification in English texts. Their technique achieved 93.20% precision for 1, 000 authors. Kabala [22] developed a computational technique of authorship classification in medieval Latin corpus. This work used Bray–Curtis distance and logistic regression for the classification task with accuracy of 99.86%. Zafar *et al.* [23] proposed a character-level CNN model with keywords and stylistic features to classify authorship of source code for three programming languages and achieved 84.94% accuracy.

Sarwar *et al.* [24] developed a multi-authorship classification method that achieved 76.92% accuracy for 1, 360 text documents. Their classification method depends on co-author information. A Latent Dirichlet Allocation (LDA) based text attribution method is used in Urdu language [25]. This method achieved an F_1 score of 92% for 6, 000 text documents. Agun *et al.* [26] evolve a statistical-based authorship text attribution system and evaluated on self-developed datasets. An extensive scale attribution system was developed by Ullah *et al.* [27], which works on 1, 000 programmers source code. This system used TF-IDF feature with deep CNN learning technique and achieved 99.00% accuracy on limited programming languages. Al-Sarem *et al.* [28] developed an ensemble technique based Arabic authorship attribution system. This method used various stylometric features and evaluated on self-build Fatwas datasets. Anwar *et al.* [25] presented a word n grams of LDA with improved sqrt similarity technique to attribute authorship and achieved 92% F_1-score on 6000 Urdu newspaper articles. Neocleous *et al.* [29] developed an ML-based authorship classification technique using several ML and stylometric features. This work showed that SVM and DT classifiers gained the highest performance on 27 essays.

Corpus availability is the prerequisite to develop any NLP tool or method in any language. The main difficulty of developing authorship classification in the Bengali language is the unavailability of the author dataset. Extracting relevant or dominating features depends on the language itself. Embedding models are generally used to extract syntactic and semantic features from the corpus of related language. As a result, embedding models developed for one language can not be applied directly to another language due to their feature unlikeness (such as embedding size, window size, epoch and min frequency words count). To address this difficulty, this work developed an embedding model for the Bengali language and evaluated it intrinsically. Moreover, the classification model and its hyperparameters should be optimized for the concerned language. Hyparameters designed for one language do not perform well in another language. The classification model developed in non-Bengali languages can not be applied for the Bengali language due to the variation of hyperparameters. Thus, we developed a Bengali authorship classification model by investigating optimized hyperparameters (such as learning rate, dropout value, kernel size, number of CNN layers, batch size, and epoch).

B. BENGALI LANGUAGE BASED AUTHORSHIP CLASSIFICATION

Although the high-resourced languages (e.g., English) have significant contributions to automatic authorship identification and author profiling, this research problem is at an early stage until now in the low-resourced languages. Although Bengali is the 7th most widely spoken language globally, it is considered one of the notable resource-constrained languages in south Asia [30]. In recent years, few studies have been conducted on authorship classification in the Bengali language with limited corpus and author classes. Phani *et al.* [1] proposed a machine learning-based technique for author identification on 3, 000 literary texts of 3 prominent Bengali authors. They used uni-gram, bi-grams and word n-grams features with the random forest algorithm, which achieved about 98.00% accuracy. A multi-layer perceptron based technique [31] used for authorship attribution, which achieved 99% accuracy. However, this approach considered only 3 author categories and ignored the writings of many modern Bengali authors. Das *et al.* [5] showed that several stylometric features could be used to distinguish Bengali authors. However, automatic identification of authors using their proposed features was absent in their work.

Chakraborty *et al.* [2] presented the performance comparison among different machine learning algorithms for Bengali authorship identification and showed that the support vector machines (SVM) outperformed other classifiers. Tamboli *et al.* [12] reviewed a few approaches for authorship identification in Bengali. They claimed that n-gram based features gained 90.00% accuracy. Hossain *et al.* [4] used a stylometry and voting based classification model for authorship classification and achieved the accuracy of 90.67% using a corpus consisting of 700 blog articles. Anisuzzaman *et al.* [9] collected a data set consisting of 107, 380 words, and Naive Bayes algorithm is used to identify the Bengali authorship. Pal *et al.* [10] proposed a Bengali authorship classification model using SVM and Naive Bayes. This method achieved an accuracy of 90.74% (SVM) and 86.21% (Naive Bayes) for 20 Bengali bloggers datasets. A multi-class SVM is used to classify the authorship from Bengali poetry [11]. They showed that the semantic and stylistic features achieved a better accuracy (92%) to identify the poet. Islam *et al.* [8] used the random forest algorithm to detect author from Bengali texts. They built a corpus consisting of 3125 passages and gained the highest accuracy (96%) with random forest than Naive Bayes (62%) and decision tree (85%) classifiers. Khatun *et al.* [6] introduced a character-level CNN for attributing Bengali authorship. This method's performance decreased with an increased number of authors and sample texts. Phani *et al.* [30] presented a Bengali authorship attribution method using n-gram feature with information gain techniques (feature ranking). The developed method worked only for three Bengali authors with 3, 000 text documents and gained an accuracy of 95% to 99%. A summary of the important aspects of recent techniques of authorship classification is presented in Table 1.

Past authorship classification studies on the Bengali language were conducted with limited author classes and a smaller corpus. None of the studies undertaken on embedding model generation and evaluation concerning the Bengali language to the best of our knowledge. Previous researches also suffered from the Out-of-Vocabulary (OOV) problem with model over-fitting. To alleviate the OOV and model over-fitting problems, the proposed work developed a new embedding corpus and a classification corpus (BACC-18) which is more extensive concerning author classes and the

TABLE 1. Authorship Classification Literature Summary.

Techniques	No. of Author	Corpus	Language	Weakness
Stylometry + Neural nets [17]	-	-	English	Failed to handle the sentence and document level semantics
Stylometry [3]	2	Online archive	English	Consider smaller number of authors
Char-CNN [7]	144, 62	Emails, IMDb	English	Consume huge memory
Stylometry + Ensemble [18]	50	Self-build	English	Data acquisition standard was not maintained
Backpropagation + Swarm operation [19]	40	Self-build	Java	Unable to address the Java code pattern
LSTM [20]	10	Self-build	Programming	Consider programming text only, literacy texts is unexplored
Naive similarity [21]	1000	Self-build	English	Word, sentence and document level semantic and syntactic features are not captured
Char-CNN + KNN [23]	20458	GCJ	Programming	Single author code, not tested on real dataset
Stylometry + Ensemble [28]	11, 8, 5	Self-build	Arabic	Consume huge time, not experimented with benchmark corpus
LDA + Sqrt similarity [25]	15	Self-build	Urdu	Supervised methods are unexploited, smaller dataset
SVM, DT, KNN [29]	3	Self-build	English	Evaluated on very limited dataset (27 documents)
N-gram + MLP [32]	3	LD	Bengali	Unable to capture local/global features
Stylometry + SVM [2]	3	Self-build	Bengali	Contextual and higher dimensional features cannot managed
Stylometry + Voting [4]	6	Self-build	Bengali	Data acquisition method is not standard
N-gram + Naive bayes [9]	3	Self-build	Bengali	Suffer from vanishing gradient problem
Naive bayes [10]	5	Self-build	Bengali	Limited number of authors and test samples
Lexical-Stylometry+SVM [11]	4	Self-build	Bengali	Failed to consider multi level semantics
Stylometry + RF [8]	10	Self-Build	Bengali	Model overfitting, dataset contained redundant text files
Char-CNN [6]	14	BAAD14	Bengali	Unable to captured Bengali language diversity and word semantics

number of unique words. Ninety tuned embedding models are generated using Word2Vec, GloVe and FastText algorithms and evaluated using the intrinsic evaluation method. Moreover, the proposed method uses word-level embedding techniques instead of previous character-level embedding, which reduces computational complexity. A superior authorship classification model is also presented by investigating a set of optimized hyperparameters with improved classification accuracy.

III. CORPUS

The word embedding technique uses an unlabeled corpus for embedding model generation, whereas the classification model uses labeled corpus for training and testing. It is a usual practice to use separate corpora: one for generating and evaluating embedding model and the another for classification model's training and testing [33]. There is a high probability that a model undergoes overfitting problem while the embedding and the classification models are learned from the same corpus [34]. Separating the embedding corpora is a good practice to achieve a better generalization for the classification models. Thus, based on the past survey [34], we have used two types of corpora: word embedding corpus and authorship classification corpus.

A. WORD EMBEDDING CORPUS (WEC)

Word embedding corpus (WEC) is a collection of unlabelled texts crawled from the various Bengali newspaper portals. Thus, the corpus is used as a main ingredient and input of the embedding algorithm to produce a lower-dimensional feature space for input words. The lower space feature vector is mainly responsible for bearing the authorship classification feature. A Python-based crawler is used to crawl the data during the period 01-01-2015 to 30-12-2020 and cleaned by data preprocessing module (described in Sec.III-B3).We have checked the textual data crawling policy of the newspapers, where the automatic crawling policy with robots.txt is allowed. We have confirmed that there is no legal embargo on crawling. Table 2 represents the various statistics of WEC. This corpus contains 0.95M unlabelled texts and a total of 854.16M words with 2.75M unique words.

TABLE 2. Key Statistics of Word Embedding Corpus.

Attributes	Values
Total no. of texts (M)	0.95
Total no. of sentences (M)	28.47
Total no. of words (M)	854.16
Total no. of unique words (M)	2.75
Total no. of unique words at min count 2 (M)	1.97
Embedding technique	GloVe
Contextual window size	13
Total no. of iteration	150
Context	symmetric
Truncating vocabulary at min count	2
Alpha	0.750000
Embedding dimension	250
Training time	16.2 hours
Number of workers thread	8
X_{Max}	100.0
GEM size	(1963483 × 250)

The learning cutoff value (X_{Max}) and quadratic terms boosting value ($Alpha$0 settled to 100.0 and 0.75, respectively). The word-word co-occurrence context is symmetric. The minimum vocabulary count is considered to 2 with the size of a window of 13.

B. AUTHORSHIP CLASSIFICATION CORPUS

This research uses three corpora to evaluate the performance of the proposed model in the authorship classification task. Two of them are benchmark corpora such as Bangla Authorship Attribution Dataset (BAAD16) [35] and Literary Dataset (LD) [30]. Owing to the limited number of author's classes and amount of data in the existing benchmark corpora, this work has developed a corpus (hereafter called 'Bengali Authorship Classification Corpus (BACC-18)'). To improve the readability, we have assigned each author a unique code in three corpora, such as *Bankim Chandra Chattopadhyay* denoted by 01, and *Rabindranath Tagore* represented by 02.

1) BAAD16

The BAAD16[1] corpus consists of 16 Bengali prominent authors with a total of 17, 966 texts (14374 training and 3592 testing texts) and approximately 13.4 million words. The average length of the authors' texts was 750 words. The author 07 found the maximum training (3, 612) and validation (906) texts. The author 21 contains the minimum training and validation texts, which amount to 148 and 37.

2) LD

Literary Dataset(LD)[2] consists of 3 eminent Bengali authors with a total of 3, 000 texts (1500 training, 750 testing and 750 validation texts). Each of the authors contained an equal number of texts (i.e., 1000). This dataset includes texts of authors 01 (Bankim Chandra Chattopadhyay), 02 (Rabindranath Tagore), and 10 (Sarat Chandra Chattopadhyay).

3) BENGALI AUTHORSHIP CLASSIFICATION CORPUS (BACC-18)

The developed BACC-18 contains the text of 18 famous authors of Bengali literature. To build this corpus, we crawled texts from four online sources namely NLTR society for natural language technology research [36], Ebanglalibrary [37], Git repository [38] and Blogs [39]–[41]. The maximum number of texts (13,308) are collected from NLTR source whereas minimum number of texts (240) are crawled from Blogs. A self-built automatic web crawler[3] is used to scrapping the data from four sources. Due to HTML page structure variation of sources, we used various web crawler instead of a typical crawler. In particular, the proposed research has developed 31 Python crawler which can automatically crawl textual data based on the robots.txt policy. The robots.txt policy ensures the search engine whether a crawler can or cannot crawl the particular text contents from a source.[4] Initially, we manually selected the famous and authentic web portal's hyperlink to collect the author's texts. Web crawler starts with the hyperlink and a spider explore all the pages under the hyperlink to scrapping the author text. After collecting all the authors' text and we prepared the authorship classification corpus with annotation based on the hyperlink. A single hyperlink contains only a single author text. This hyperlink based web crawling reduces the manual annotation time and cost of human efforts.

TABLE 3. **BACC-18 Data Distributions.**

Author Code	NLTR	Ebanglalibrary	Git	Blog	No. of Text
01	1,106	-	-	-	1,106
17	-	-	503	-	503
19	-	381	-	-	381
03	285	189	-	-	474
02	4,643	515	-	-	5,158
08	-	822	-	-	822
09	-	1,762	-	-	1,762
10	3,000	-	-	-	3,000
11	-	2,007	-	-	2,007
12	-	1,293	-	-	1,293
18	-	1,947	-	-	1,947
24	-	486	-	-	486
05	-	444	-	-	444
15	4,274	-	-	-	4,274
16	-	-	627	69	696
14	-	299	42	87	428
07	-	644	-	71	715
13	-	240	-	13	253
Total	13,308	11,029	1,172	240	25,749

a: DATA PREPROCESSING

The web crawled data contain plenty of irrelevant characters, symbols, or mathematical expressions that cannot be converted to UTF-8 format. Thus, each collected text file requires the following pre-processing:

- Replacement of all non-Bengali alphabets and digits.
- Removal of regular expression and symbols by a single white space.
- removal HTML tags, hashtags, URLs, punctuation and whitespaces.
- Replacement of multiple new lines by a single new line.
- Elimination of the duplicate text.

If an author's text contains less than fifteen words and its size is greater than 50 KB, then this text is eliminated from the corpus.

b: DATA DISTRIBUTIONS

Table 3 shows the source-wise distribution of text documents with corresponding author code where the code numbering performed by Python script.

The $BACC - 18$ corpus is unbalanced due to the variation of text amount in each category. The highest amount of data belong to author 02 (20.03%) whereas the lowest data to author 13 (0.98%).

Table 4 shows the various statistics of the three corpora.

Table 4 indicates that the BACC-18 is the most voluminous corpus compared to BAAD16 and LD. The BACC-18 contains 25,749 texts, whereas BAAD16 contains 17,998 and LD belong to only 3000. In addition to that, 18 authors' data are accumulated in BACC-18, which is greater than BAAD16 (16 author's text) and LD (3 author's text).

[1] https://data.mendeley.com/datasets/6d9jrkgtvv/4

[2] https://web.eecs.umich.edu/ lahiri/literary.zip

[3] https://github.com/mrhossain/scrapping

[4] https://developers.google.com/search/docs/advanced/robots/create-robots-txt#format-and-location

TABLE 4. Statistics of Authorship Classification Corpora.

Attributes	BAAD16 [35]	LD [30]	BACC-18
Total no. of text	17,998	3,000	25,749
Total size (MB)	210.8	21.6	160.1
Total no. of words (M)	13.47	1.25	9.86
Total no. of unique words (M)	0.59	0.08	0.38
Total no. of sentences (M)	0.71	0.08	0.75
Train max. (min.) text size (KB)	16.3 (9.6)	31.4 (1.4)	30.9 (702)
Test max. (min.) text size (KB)	14.6 (10.0)	37.5 (1.2)	118.5 (818)
Validation max. (min.) text size (KB)	14.6 (10.0)	35.4 (1.3)	80.7 (792)
Total no. of classes	16	3	18

4) TRAINING AND TESTING SETS

The classifier models are evaluated with BACC-18, BAAD16 and LD datasets. The dataset is partitioned into training and testing sets. We eventually partitioned the data set intuitively and started with partitioning into 90% for training and 10% for testing. While verifying the partitions experimentally, we encountered problems where the test set fails to include the maximum (text file size $>$ 118.5 KB) and minimum (text file size $<$ 818 bytes) length of author texts. The intuitive approach is then developed into a heuristic approach whereby ≈5% of text files are moved from training set into test set to minimize the text length deficiency. This manual addition of ≈5% samples into the test set helps reinforce the data distribution of the test set so that the model can perform classification task with diverse data samples. The heuristic partitioning rule is formulated as Algorithm 1. This Algorithm 1 takes BACC-18 labeled corpus as the input and produces a training set (X_train: ≈85% of data) and a testing set (X_test: ≈15% of data).

Algorithm 1 Random and Rule Based Heuristic for BACC-18 Train and Test Sets Partition

```
$Input \leftarrow BACC - 18\ Corpus$
$MxL \leftarrow 118.5$ {Maximum texts size in KB}
$MnL \leftarrow 818$ {Minimum text size in Byte}
$X\_train, X\_test, y\_train, y\_test \leftarrow train\_test_{s}plit(BACC - 18, labelled, 0.10, random\_state : 42)$ {Randomly partition of train, test sets}
procedure $DataSplit(BACC - 18)$ {Rule based heuristic}
    for $i \leftarrow 1 \rightarrow len(BACC - 18)$ do
        $L \leftarrow len(BACC - 18[i])$
        if $BACC - 18[i]\ notin\ X\_test\ \&\ L \geq MxL$ then
            $X\_test.append(BACC[i])$
        end if
        if $BACC - 18[i]\ notin\ X\_test\ \&\ L \leq MnL$ then
            $X\_test.append(BACC[i])$
        end if
    end for
    return $X\_train, X\_test$
end procedure
```

Table 5 shows the author-wise training and testing data distributions in all datasets.

TABLE 5. Training and Testing Summary.

Corpus	Author Code	No. of Testing text	No. of Training text
BACC-18	01	179	927
	17	76	427
	19	20	361
	03	84	390
	02	720	4438
	08	34	788
	09	57	1705
	10	461	2539
	11	47	1960
	12	196	1097
	18	42	1905
	24	61	425
	05	61	383
	15	608	3666
	16	22	674
	14	58	370
	07	97	618
	13	32	221
Total	18	2,855	22,894
BAAD16	06	220	880
	01	112	450
	07	906	3612
	04	93	376
	03	44	179
	22	95	381
	02	252	1007
	23	210	838
	08	282	1126
	25	177	711
	10	261	1051
	11	169	680
	05	393	1570
	20	155	620
	14	186	745
	21	37	148
Total	16	3,592	14,374
LD	01	250	500
	02	250	500
	10	250	500
Total	3	750	1,500

The training set contains 22894 text data and testing set contains 2855 with 18 different authors. Maximum training samples are taken from author 02 (4438 texts) and minimum number of samples from author 13 (221 texts). Due to shortage of enough samples, the testing and validation datasets are remain the same. The BAAD-18 is available on public repository.[5]

IV. MODEL ARCHITECTURE

The authorship classification proposed architecture comprises four main modules: embedding model generation, feature representation, classifier training and classifier testing. The embedding model generation function produces the embedding models from unlabelled embedding corpus using an unsupervised technique (e.g., GloVe, FastText or Word2Vec). The feature representation is a function that converts the author text to numeric lower dimension

[5] https://github.com/mrhossain/Bengali-Authorship-Classfication-Corpus-BACC-18

representation (2D), whereas the rows indicate the word index and columns represent the feature value. The training module generates the classifier model with labelled datasets. The testing module evaluates the performance of the classifier models with the unlabeled datasets. Fig. 1 illustrate the schematic representation of the proposed authorship classification framework using CNN [15].

A. PRE-TRAINED WORD EMBEDDINGS (PWE)

Embedding model generation is an essential pre-requisites of authorship classification framework. An authorship embedding corpus (AEC) is used to generate the Embedding Model (EM). In the proposed framework we used three standard embedding techniques such as Global Vectors for Word Representation (GloVe) [42], Distributed Representations of Words (Word2Vec) [43], and Enriching Word Vectors with Subword Information (FastText) [44]. Three embedding techniques in combination with different hyperparameters generate ninety embedding models.

The pre-trained word embeddings (PWE) process initializes the unlabeled authorship embedding corpus (AEC) and produces a Embedding Model (EM) as the output. The AEC can be defined as $T = \{aet_1, aet_2, aet_3, \dots, aet_M\}$, where aet_i denotes the i^{th} author embedding text (aet^i) for $i = 1, 2, 3, \dots, M$. M represents the total number of text documents in AEC and $aet^i = [S_1, S_3, S_3, \dots, S_l]$, where l indicates the total number of sentence (S^i) of i^{th} text. Single File Conversion (SFC) process takes all the *eat* and concatenates the sentences one after another and finally produces a single embedding file (Ψ), where $\Psi = \{ [S_{1:1}, S_{1:2}, S_{1:3}, \dots, S_{1:p}] \oplus [S_{2:1}, S_{2:2}, S_{2:3}, \dots, S_{2:q}] \oplus \dots \oplus [S_{M:1}, S_{M:2}, S_{M:3}, \dots, S_{M:r}\}$ and, p, q and r denotes the length of sentences respectively. Now the embedding file (Ψ) is fed to the three different embedding techniques (e.g., GloVe, Fasttext and Word2Vec). The next subsections describe the embedding techniques.

1) GloVe

The Train Embedding Model (TEM) takes Ψ as the input, and the GloVe algorithm generates the word co-occurrence statistics represented by a matrix X. Each entry of $X_{(i,j)}$ matrix denotes how often word i appears in context of word j. The context terms are defined by a *window_size* before the term and a *window_size* after the term. Semantic and syntactic feature representations are obtained from the center and context words. The X_{max} represents the maximum word co-occurrence in Ψ. The GloVe performance partially depends on the cut-off value of X_{max}. After the training, TEM produces an embedding model (EM) for word-wise feature representation. GloVe algorithm produces *eighteen* embedding models (EM). Various combination of hyperparameters are used to analyze the model's performance such as ED : $\{50, 100, 150, 200, 250, 300\}$, *VOCAB_MIN_COUNT* : $\{2\}$, *X_MAX* : 100, *Epochs* : 35 and *window_size* : $\{10, 13, 15\}$ respectively. Four other parameters (BINARY, VERBOSE, MEMORY & NUM_THREADS) settled to default values. A total of *eighteen* embedding models are generated using GloVe techniques. The *eighteen* embedding models come from six embedding dimensions and three window sizes.

2) Word2Vec

Word2Vec algorithm takes Ψ as the input and generates a embedding model EM as output. Two versions of Word2Vec pre-trained models such as Skip-gram and continuous bag of words (CBOW) are used with similar hyperparameters for tuning: ED : $\{50, 100, 150, 200, 250, 300\}$, *window_size* : $\{10, 13, 15\}$, *min_count* : $\{2\}$, learning rate (lr) : 0.093, *thread* : 8 and *epochs* : 30 respectively. There are thirty-six embedding models (e.g., 18 for CBOW and 18 for skip-gram) are produced using various hyperparameters combinations (i.e., six embedding dimensions and three window size).

3) FastText

FastText technique considers Ψ as the input and produces an EM as the output. The developed EMs are modified with different hyperparameters such as ED : $\{50, 100, 150, 200, 250, 300\}$, *window_size* : $\{10, 13, 15\}$, *min_count* : $\{2\}$, learning rate (lr) : 0.13, *thread* : 8 and *epochs* : 30 respectively. The proposed hyperparameters generate thirty-six models (e.g., 18 for FastText-skip-gram and 18 for FastText-CBOW) using the combinations of six embedding dimensions and three window sizes.

The intrinsic evaluators measure the performance of a total of *ninety* embedding models (18 for GloVe, 36 for Word2Vec and 36 for FastText). Out of these models, the top nine performing models are selected for extrinsic evaluation, such as authorship classification (described in Sec.VII-A).

B. FEATURE REPRESENTATION

To represent the features, a labelled Authorship Classification Corpus (ACC) is used during the training while an unlabeled text is used in testing. In both training and testing phases, features can be represented as an input feature matrix ($\mathcal{M}$). ACC can be defined as $\chi = \{lt_1, lt_2, lt_3, \dots, lt_N\}$, and $\chi \in \mathbb{R}^N$, where N denotes the total number of text in ACC. Author Text to Word List Conversion (ATWLC) process sequentially takes the input lt_i and output as a list vector which is defined as $\mathcal{L} = \{l_1, l_2, l_3, \dots, l_{\mathcal{W}}\} \in \mathbb{R}^{\mathcal{W}}$, where $\mathcal{W}$ denotes the maximum number of words allowed in $\mathcal{L}$. This framework truncates the first 1896 words and assigns null value when the text contains less-than 1896 words.

The feature matrix generation (FMG) has two constituents: Word Lookup Table (WLT) and Feature Map (FM). The WLT takes the i^{th} word l_i, where $l_i \in \mathcal{L}$ and produces a Hash Index (H_i). The WLT consists of a $(key, value)$ vocabulary table, where *key* denotes the word (l_i) and its value is the index (H_i). If a word found in the vocabulary table then it returns the corresponding index (H_i), otherwise returns zero value. Feature Map takes two inputs (H_i and EM) and produces an output (input feature matrix ($\mathcal{M}$)). The EM can be defined as $\mathcal{G} = H_i \times F$, where $H_i = \{h_1, h_2, h_3, \dots, h_{\mathcal{P}}\}$, $F = \{f_1, f_2, f_3, \dots, f_{ED}\}$ and

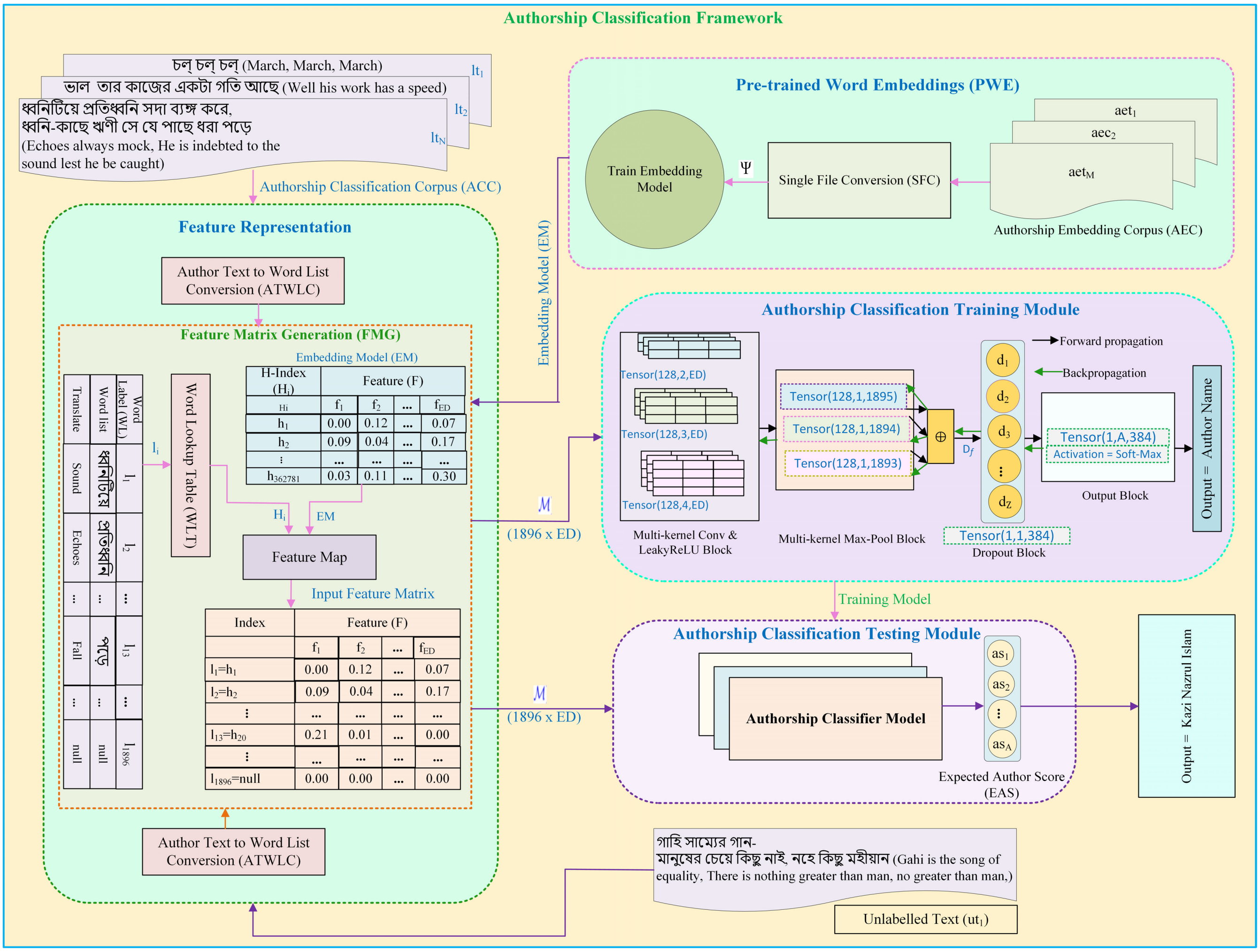

Authorship Classification Framework
চল্ চল্ চল্ (March, March, March)
lt_1
ভাল তার কাজের একটা গতি আছে (Well his work has a speed)
lt_2
ধ্বনিটিয়ে প্রতিধ্বনি সদা ব্যঙ্গ করে,
ধ্বনি-কাছে ঋণী সে যে পাছে ধরা পড়ে
(Echoes always mock, He is indebted to the sound lest he be caught)
lt_N
Authorship Classification Corpus (ACC)
Feature Representation
Author Text to Word List Conversion (ATWLC)
Feature Matrix Generation (FMG)
Embedding Model (EM)
H-Index (H_i)
Feature (F)
Hi
f_1
f_2
...
f_{ED}
h_1 0.00 0.12 ... 0.07
h_2 0.09 0.04 ... 0.17
⋮ ... ...
h_{362781} 0.03 0.11 ... 0.30
Word Label (WL)
Word list
Translate
l_1 ধ্বনিটিয়ে Sound
l_2 প্রতিধ্বনি Echoes
l_{13} পড়ে Fall
l_{1896} null null
l_i
Word Lookup Table (WLT)
H_i
EM
Feature Map
Input Feature Matrix
Index
Feature (F)
f_1
f_2
...
f_{ED}
l_1=h_1 0.00 0.12 ... 0.07
l_2=h_2 0.09 0.04 ... 0.17
⋮ ... ...
l_{13}=h_{20} 0.21 0.01 ... 0.00
⋮ ... ...
l_{1896}=null 0.00 0.00 ... 0.00
Author Text to Word List Conversion (ATWLC)
Embedding Model (EM)
$\mathcal{M}$
(1896 x ED)
$\mathcal{M}$
(1896 x ED)
Pre-trained Word Embeddings (PWE)
Train Embedding Model
Ψ
Single File Conversion (SFC)
aet_1
aec_2
aet_M
Authorship Embedding Corpus (AEC)
Authorship Classification Training Module
Tensor(128,2,ED)
Tensor(128,3,ED)
Tensor(128,4,ED)
Multi-kernel Conv & LeakyReLU Block
Tensor(128,1,1895)
Tensor(128,1,1894)
Tensor(128,1,1893)
Multi-kernel Max-Pool Block
⊕
D_f
d_1
d_2
d_3
⋮
d_Z
Tensor(1,1,384)
Dropout Block
Forward propagation
Backpropagation
Tensor(1,A,384)
Activation = Soft-Max
Output Block
Output = Author Name
Training Model
Authorship Classification Testing Module
Authorship Classifier Model
as_1
as_2
⋮
as_A
Expected Author Score (EAS)
Output = Kazi Nazrul Islam
গাহি সাম্যের গান-
মানুষের চেয়ে কিছু নাই, নহে কিছু মহীয়ান (Gahi is the song of equality, There is nothing greater than man, no greater than man,)
Unlabelled Text (ut_1)

FIGURE 1. Proposed authorship classification framework.

ED $\in$ (50, 100, 150, 200, 250, 300). Here, h_i denotes i^{th} vocabulary Hash-Index and F indicates feature vector. $\mathcal{P}$ and ED represents the total number of vocabulary and embedding dimension respectively. In the proposed framework, $\mathcal{P}$ = 362781 and ED $\in$ (50, 100, 150, 200, 250, 300). If $l_i \in H_i \times F$ then the process extracts corresponding feature from $\mathcal{G}$ and adds a row major order to IFM. If $l_i \notin H_i \times F$ then adds ED times zero for l_i. In this way, the process finally produces an $\mathcal{M}$ of (1896×ED), where 1896 denotes the number of words and ED indicates number of features in each word.

C. CLASSIFIER TRAINING MODULE

The training module takes the feature matrix ($\mathcal{M}$) as the input where $\mathcal{M} \in$ (1896×ED) and produces a classifier model as the output (Fig. 1). Training module extracts the author text level stylometric and semantic features from multi-kernel CNN block. Initially, the filter weights initialized by Xavier initialization [45] function, and weights are adjusted using the backpropagation technique [46]. The Multi-kernel Conv and LeakyReLU Block uses a matrix $\mathcal{M}$ = (1896×ED) which generates three different kernels tensors shapes: *Tensor*(128, 2,ED), *Tensor*(128, 3,ED) and *Tensor*(128, 4,ED). These tensors use to conduct the individual convolution operation. A special type of convolution produces the single dimension feature vector using the Eq. 1.

$$V_{[k^{th}:]} = \sum_{i=1}^{R} M[i:end] \otimes K[k^{th}:end] + B_i \tag{1}$$

here, k^{th} kernel produce a $V_{[k^{th}:]}$ single dimension feature vector and R denotes the input feature matrix rows. The bias value B_i is added with the output convolution value. In the proposed method, output of the convolutions are *Tensor*(128, 1, 1895), *Tensor*(128, 1, 1894) and *Tensor*(128, 1, 1893) respectively. The first value of the tensor 128 indicates the number of kernels. The Leaky ReLU operation is also applied to all Tensors.

Convolution operation followed by an activation operation within a block. The activation function is the data distribution re-scaling function which reduces the model overfitting and underfitting problems [47]. The types of activation functions depend on the nature of data distribution and network depth. In the proposed architecture, we used ReLU [48] activation function, which can be represented by the Eq. 2.

$$V(i,j)^{k^{th}} = \begin{cases} a \times V(i,j)^{k^{th}} \ if \ V(i,j)^{k^{th}} \leq 0 \\ V(i,j)^{k^{th}} \ otherwise \end{cases} \tag{2}$$

here $V(i,j)^{k^{th}}$ represents i^{th} rows and j^{th} columns of k^{th} kernel cell value where $V(i,j)^{k^{th}} \in \mathbb{R}^V$. The small multiplicand a is multiplied with the input cell, when the cell value is negative. The convolution and activation operations are followed by a pooling operation for Multi-kernel Max-Pool Block.

Pooling is a dimension reduction operation applied to the pooling layers. A max-pool operation is conducted over the whole feature ($V_{k^{th}}$) using the Eq. 3.

$$FP_{k^{th}}[1:128] = max_{j=1}^{j=128}(V_{k^{th}}[j:end]). \tag{3}$$

here, k^{th} tensor produces a 128 single dimension feature vector. In the proposed architecture, three individual tensors produce the three 128 single dimension feature vectors. The feature vector is concatenated using the Eq. 4.

$$D_f[1:N_d] = FP_1[1:128] \oplus FP_2[1:128] \oplus FP_3[1:128] \tag{4}$$

where, N_d indicates the concatenate dimension with 384 feature values and 128 kernels.

The dropout operation randomly drops out some nodes whose value is settled to zero, and this operation controls the overfitting problems [49]. The dropout operation can be conducted using the Eq. 5.

$$D_f[1:N_d] = (F[1:N_d] \otimes Dr[1:N_d]) + B[1:N_d] \tag{5}$$

here, $D_f[1:N_d]$ denotes probability described by Bernoulli distribution. Each filter computes 384 value and drops out some node value according to the dropout threshold value.

Output block is the last block of the classification model which uses a tensor with $Tensor(18, 383)$ and Soft-Max activation function. Tensor internal value 18 denotes the number of authors, and 384 are the feature values. This block contains a classifier matrix Θ (soft-max layer output), where a row represents the author identity and column represents the trainable weights. For any author, a, the expected author's value can be calculated by using Eq.6.

$$EX(Author = a|\mathcal{Z}) = \frac{(e^{\mathcal{Z}^T\Theta_a})}{\sum_{i=1}^{A} e^{\mathcal{Z}^T\Theta_i}} \tag{6}$$

where, $\mathcal{Z}$ is the author's feature value (i.e. 384), A is the total number of authors (i.e. 18) and Θ_a is the ath author feature values. Thus, the probable author is computed from the maximum expected value ($Max(EX(Author = a|\mathcal{Z}))$).

In a forward pass, the deviation or error between the input and output is calculated from the maximum expected value ($1 - Max(EX(Author = a|\mathcal{Z}))$). The error value is adjusted using the backpropagation technique [46], and each of the kernels is trained with adjustable values. In this way, the whole datasets pass through the networks 200 times and well-tuned the hyperparameter's value. At the end of the training, the model file is saved for the testing purpose. The layer-wise weight values are stored through the metafile. To investigate the effect of authorship classification performance, LSTM [20], Char-level-CNN [6], SVM [10], SGD [50], Multilingual pre-trained BERT (M-BERT) [51] and Distil-BERT [52] classifiers are also implemented on the same datasets.

D. CLASSIFIER TESTING MODULE

This module takes feature matrix ($\mathcal{M}$) as the input where $\mathcal{M} \in (1896\times$ ED) and generates an Author Expected Score (AES) vector as the output. In testing, the feature representation process starts with an unlabeled text (ut_1) and generates an input feature matrix ($\mathcal{M}$). The feature matrix generation procedure is the same as described in Sec. IV-B. If ut_1 contains values smaller than 1896 words then zero padding is added and truncates the first 1896 words if the value is more than 1896. The generated feature matric ($\mathcal{M}$) is fed into the authorship classifier model and is passed forward through the architecture with kernels initialized by trained kernels weights. The proposed architecture settles at a dropout value of 0.5, and the test module produces a feature vector of 385. The feature vector is projected into an output block weight matrix with a shape of 18 × 384. The Soft-Max function produces AES using equation (6). The AES is defined as $\mathcal{A} = \{as_1, as_2, as_3, \ldots, as_{18}\}$ and finds the maximum expected value (Max($\mathcal{A}$)) with a corresponding index of the predicted author name. For example, the authorship classifier model produces the author name (i.e., class) as *KaziNazrulIslam* for an unlabeled input text ut_1.

V. HYPERPARAMETERS IDENTIFICATION AND OPTIMIZATION

Hyperparameters with corresponding values have a more substantial effect on supervised machine learning method [53]. The optimized hyperparameters may reduce the training time of the classifier model and improve its accuracies. To optimize the CNN parameters, we initialized our network parameters using Dennybritz[6] network values (Table 6).

The first step is to select a feature extractor to optimize the hyperparameters and reduce the classification errors. Three embedding techniques and various hyperparameter combinations have generated *ninety* embedding models. Among *ninety* models, the best *nine* models are selected based on intrinsic evaluations (Sec.VII-A). The *nine* models are evaluated for hyperparameters optimization. The embedding models initial hyperparameters and BACC-18 performance are

[6] https://github.com/dennybritz/cnn-text-classification-tf

TABLE 6. Initial Hyperparameters.

Parameters	Parameters Value
Batch size	64
Embedding models	Word2Vec, FastText, GloVe
Embedding dimension	128
Filter size	3, 4, 5
Number of epoch	200
Number of filter	128
Pooling size	3, 4, 5
Dropout rate	0.50
Activation	ReLU, Soft Max
Pooling type	Max

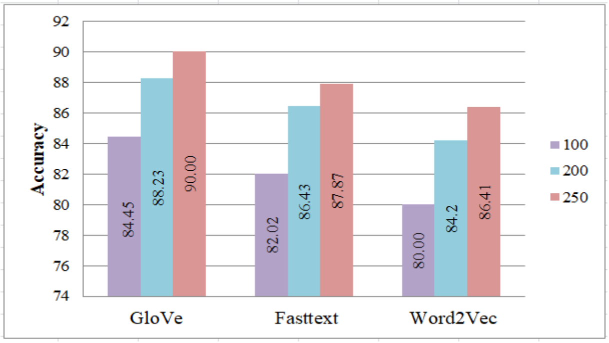


FIGURE 2. Impact of ED on embedding models with BACC-18.

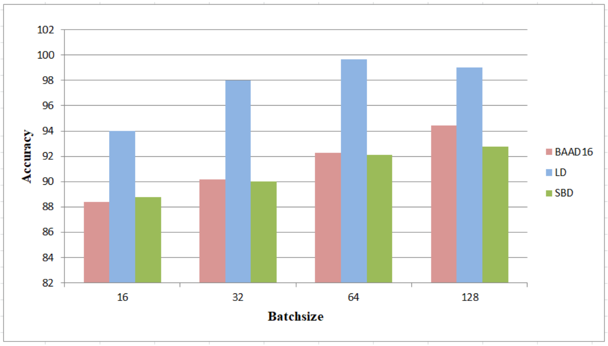


FIGURE 3. Effect of batch size on classifier performance.

shown in Fig. 2. Among all embedding techniques, GloVe achieved the highest accuracy in all three embedding dimensions (ED) such as 84.13% for ED 100, 88.05% for ED 200 and 89.98% for ED 250. Since GloVe outperformed the other models, we selected this model to estimate other suitable hyperparameters.

The batch size depends on the number of training samples, network layers, physical memory and GPU memory capacity [54]. The effect of different batch sizes on accuracy are shown in Fig. 3. The proposed approach has been verified on 16, 32, 64 and 128 batch sizes with BACC-18, BAAD16 and LD datasets.

Fig. 3 shows that the classifier model achieved 88.10% accuracy for *BAAD*16, 94.00% accuracy for *LD* and 88.89% accuracy for a batch size of 16 of dataset *BACC* − 18. The *LD* achieved the highest accuracy of 99.87% for a batch size of 64 whereas *BAAD*16, and *BACC* − 18 gained the highest accuracies of 94.34% and 94.86% for a batch size of 128. The *LD* dataset's performance degrades at a batch size of 128 due to small numbers of training and testing samples. Thus, the impact of batch size on accuracy is strongly associated with the number of authors, volumes of the training set, and samples.

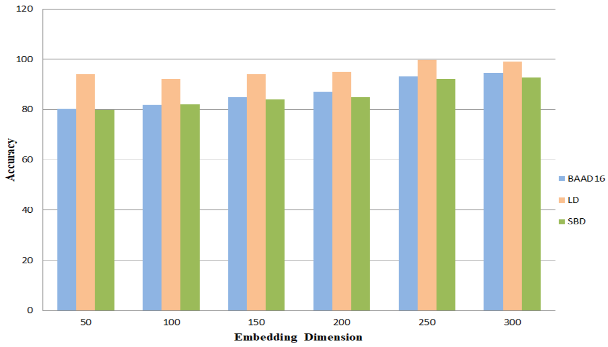


FIGURE 4. Impact of embedding dimension on classifier performance.

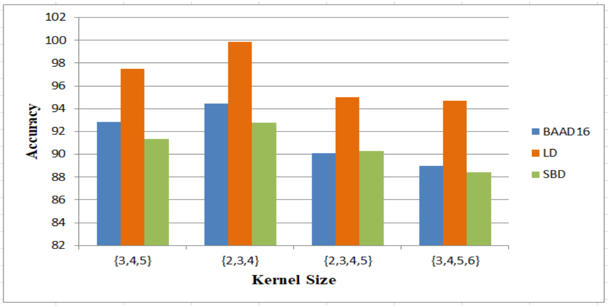


FIGURE 5. Impact of filter size on classification accuracy.The numeral values 2, 3, 4, 5 and 6 denotes the filter size.

ED is the most impactful hyperparameter in the classification task and refers to the number of feature values represented by each word. Fig. 4 shows the influence of ED on accuracy.

Initially, three datasets performed poorly at ED 50, whereas the highest accuracies are obtained for ED of 250 to 300. The ED hyperparameter tuning analysis shows that classification accuracy goes upwards from 50 to 250. However, accuracy turns downwards after 250 ED (for *LD*) and upwards for *BAAD*16 & *BACC* − 18. Therefore, the classifier accuracy depends on the ED while the ED depends on the number of authors and training samples.

Filter size represents how many features are masking with the embedding matrix [55]. Impact of filter size on accuracy is shown in Fig. 5. The maximum accuracy is achieved for the filter set {2, 3, 4} whereas the minimum accuracy is achieved for {3, 4, 5, 6}. The result shows that an increase in the filter size decreases the accuracy and increase in the number of filters decreases the accuracy for all datasets. Thus, the filter size, number of filter and accuracy depends on distinguished feature values of datasets.

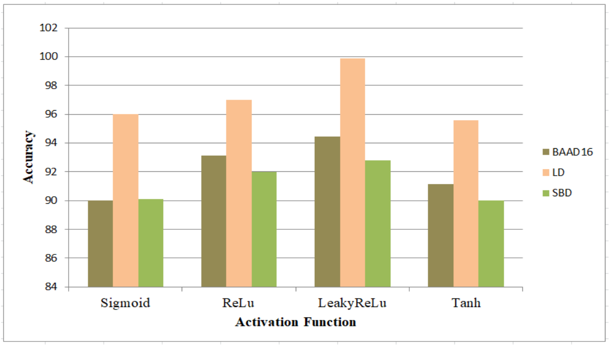


FIGURE 6. Impact of activation function on network performance.

TABLE 7. Optimized Hyperparameters.

Parameters	Proposed (CNN+GloVe)	Char-CNN [6]	TF+NB [30]	GloVE+SVM [1]
Attributes	Values			
Number of layer	1	2	-	-
Batch size	128	32	-	-
Embedding model	GloVe	Fastext	TF	GloVe
Embedding dim.	250	300	300	550
Filter size	2, 3, 4	7, 3	-	-
Number of epoch	200	10	-	-
Number of filter	128	128, 256	-	-
Pooling size	2, 3, 4	3	-	-
Dropout rate	0.54	-	-	-
Activation	LeakyReLU, Soft-Max	ReLU	-	-
Pooling type	Max	Max	-	-
Classifier	CNN	CNN	NB	SVM

An activation function is a non-linear data normalization function which decreases the model training time and aided to overcome the overfitting problems [56]. The non-linear function captures the complex features of the classification tasks. Fig. 6 shows the influence of activation function on accuracy in *three* datasets. The maximum accuracy is achieved from LeakyReLu [57] activation, whereas the minimum accuracy is gained from Sigmoid [58] activation function. Thus, the proposed approach is tuned by LeakyReLu non-linear activation function.

Table 7 summarizes the optimized hyperparameters of the proposed method including previous approaches (e.g., char-CNN [6], TF+NB [30], and GloVE+SVM [1]. The hyperparameters are tuned based on trial and error technique [59].

VI. EXPERIMENTS

The Python environment implements the proposed CNN based authorship classification with TensorFlow. A multiple core-i7 CPU with NVIDIA 1070 GPU, 32 GB physical memory and 8 GB GPU memory is used for implementation. The CNN architecture converses at 200 epoch with a minimum loss value. In convolution layers, hyperparameters are tuned and adjusted for the number of filters in each layer. Filter values are initialized by the Xavier initialization function [45].

A. EVALUATION MEASURES

The proposed authorship classification approach is evaluated in three ways: embedding model evaluation, training phase evaluation and testing phase evaluation. Embedding model evaluation is refers to the quality judgement of feature vectors which is an essential tasks for the low-resource languages [60]. The intrinsic and extrinsic evaluations are used for evaluating the embedding model. The intrinsic evaluators evaluate the semantic, syntactic and relatedness quality whereas the extrinsic evaluators evaluate the downstream tasks e.g., classification [14], machine translation [61], word-sense disambiguation [62], and question answering [63]. Spearman ($\hat{\rho}$) and Pearson ($\hat{r}$) correlations are used for intrinsic evaluation such as semantic word similarity ($S_{s\hat{\rho}}/S_{s\hat{r}}$) and syntactic word similarity ($S_{y\hat{\rho}}/S_{y\hat{r}}$).

The extrinsic performance can be calculated from classification measures [64]. Loss and accuracy measures are used to evaluate the training and validation phases. The loss value is calculated by Eq.7.

$$L^i = -\sum_{c=1}^{N_s} A_c^i \times log(P_c^i) \tag{7}$$

where, L^i represent the i^{th} iteration loss value, P_c^i and A_c^i represents i^{th} iteration predicted value and the actual value with class labeled c respectively. Total number of classes is represented by N_s and its value is 18. Training and validation accuracy are calculated by Eq.8.

$$Acc^i = \frac{H_s}{S_s} \tag{8}$$

where, Acc^i denotes i^{th} class accuracy, S_s indicates the total number of samples and H_s denotes the total number of hit or predicted.

Several statistical measures such as precision (Pr), recall (Rr), Micro F1-score, confusion matrix (CM), Macro-F1 (M-F1), Weighted-F1 (W-F1) and accuracy (A) are used to evaluate the performance of the proposed authorship attribution approach [65].

VII. RESULTS

A common empirical approach is to apply different combinations of embedding and classification techniques to the authorship classification task.

A. EMBEDDING MODELS EVALUATION

Various combinations of hyperparameters of three embedding techniques (e.g., GloVe, FastText and Word2Vec) have generated 90 local contextual embedding models [18 for GloVe, 36 for FastText (Skip-gram & CBOW), 36 for Word2Vec (Skip-gram & CBOW)]. Intrinsic evaluators are used to evaluate a total of 90 models using syntactic and semantic similarity measures [60]. Based on the intrinsic evaluation performance, a total of 9 top-performing embedding models are selected for the authorship classification task. In particular, three models are chosen from GloVe, three

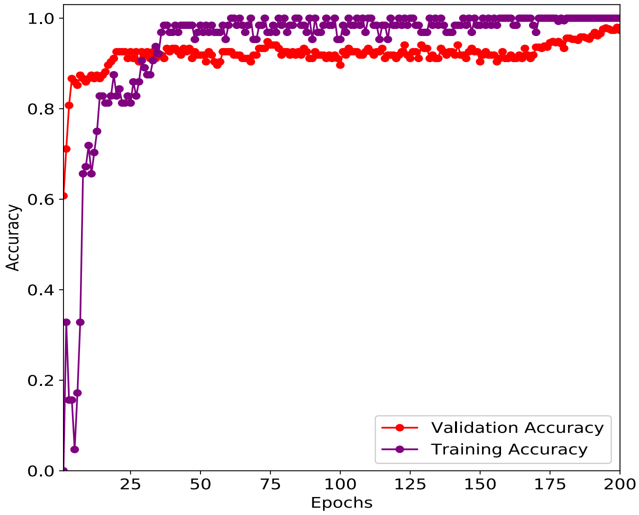


FIGURE 7. **Training and validation accuracy vs epochs.**

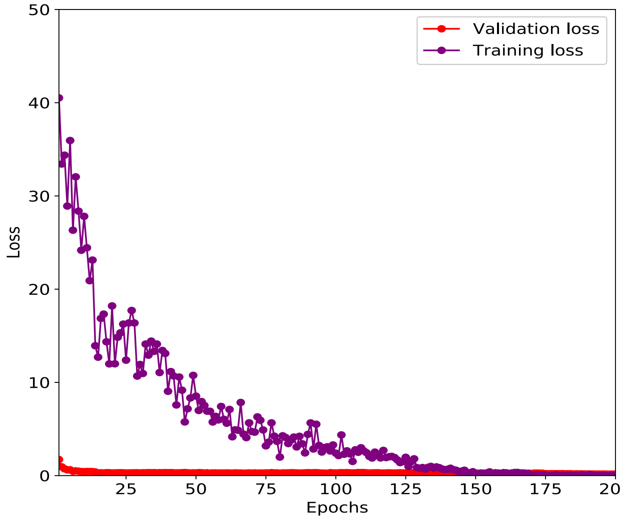


FIGURE 8. **Training and validation loss vs epochs.**

from FastText and three from Word2Vec embeddings based on the highest Pearson and Spearman correlation scores. Table 8 shows the best nine intrinsic evaluation results of three embedding techniques for 100 semantic and syntactic word pairs.

The maximum semantic word similarity in terms of Pearson (62.02%) and Spearman (63.66%) correlation scores has been achieved for the GloVe model with an embedding dimension (ED) of 250 and 13 windows size. In syntactic similarity, maximum Pearson correlation score (71.09%) is obtained from the Glove model with ED of 200 and 15 contextual window, whereas the highest Spearman correlation score (72.30%) is achieved for the GloVe with 250 ED and 13 window. The FastText model has achieved the second-best performance with Pearson and Spearman correlation for semantic and syntactic similarity evaluations. It is observed that the Word2Vec technique has achieved the lowest performance. Thus, it reveals that the GloVe embedding technique with 250 embedding dimension and 13 contextual windows size performed the best than Word2Vec and FastText.

B. TRAINING PHASE EVALUATION

Figure 7 shows the evaluation results of the training and validation phases. Initially, the training accuracy starts from 0.13 at the first epoch. The accuracy varied from 0.2 to 0.7 for the epochs $2-20$. After the 20 epoch, the accuracy values rise sharply. Accuracy varied from 0.75 (27 epoch) to 1.00 (40 epoch). The training accuracy is stable between epoch numbers $180-200$ with the highest value of 1.00. On the other hand, the validation accuracy starts from 0.65 at epoch number 1 (Fig. 7). This accuracy is almost stable between 25 and 175 epoch and reached the highest values (0.97) between 180 to 200 epoch. Loss value or error minimization is the primary concern for better classifier model generation. Fig. 8 exhibits the validation and training losses in different epochs.

Figure 8 shows that the validation loss value starts from 1.56, and after a few epochs, the loss values slow down to 0.05. The validation loss mostly minimizes after epoch number 25 and continues up to 150. The straight-line part (curve in red) indicates that the loss value is stable, and no more chance to minimize the validation loss value. After the 162 epoch, the loss value settled to the global minimum loss value (0.0045). The training loss begins at 43.00, which is a logarithmic loss value. Loss value step-downs quickly with the increase of epoch numbers. After completing an epoch, the trained model adjusted the weight values and reduced the training loss. From 50 to 125 epoch, the loss values are gradually reduced and stable between 150 to 200 epoch with minimum errors.

C. TESTING PHASE EVALUATION

Intrinsic evaluation selects the top *nine* best performing embedding models. The authorship classification task is evaluated on BACC-18, BAAD16 and LD. To investigate the Bengali authorship classification task performance, we implemented 36 models using the combinations of 9 embedding models and four classification techniques (SVM, SGD, CNN & LSTM). In particular, 36 models are generated from four classification techniques where SVM and SGD contributed to 18 models ([9 (embedding model) x 2 (SVM & SGD)]), 9 models contributed to CNN-based models ([9 (embedding model) x 1 (CNN)]) and LSTM-based models contributed to 9 models. Among 36 models, Table 9 illustrates 18 top-performing models, which are selected based on different classification techniques. The CNN + GloVe model achieved the highest accuracy of 86.98% (for BACC-18), 91.08% (for BAAD16), and 96.67% (for LD) in 100 EM. In the 200 ED, BACC-18 test datasets obtained the highest of 91.07% accuracy using GloVe+CNN 200 ED and 92.96% accuracy for BAAD16 test datasets, whereas Fasttext+CNN gained the maximum of 98.53% accuracy for LD test datasets. With the 250 embedding dimension, the GloVe+CNN obtained the highest accuracy of 93.45% (for BACC-18), 95.02% (for BAAD16) and 98.67%

TABLE 8. Summary of the intrinsic evaluation results for best-performing embedding techniques based on Spearman ($\hat{\rho}$) and Pearson ($\hat{r}$) correlations.

			Semantic (%)		Syntactic (%)	
Model	ED	Window	$S_{s\hat{\rho}}$	$S_{s\hat{r}}$	$S_{y\hat{\rho}}$	$S_{y\hat{r}}$
GloVe	**250**	**13**	**62.02**	**63.66**	68.12	**72.30**
	200	**15**	56.33	57.77	**71.09**	72.01
	100	10	53.20	55.07	67.10	69.12
FastText	250	13	57.11	58.19	44.34	46.31
	200	15	53.64	53.38	39.47	38.31
	100	10	56.78	55.18	37.78	36.03
Word2Vec	250	13	44.31	45.10	31.78	30.51
	200	15	44.31	45.10	31.78	30.51
	100	10	44.31	45.10	31.78	30.51

(for LD). The analysis revealed that the proposed approach (GloVe+CNN) achieved the best performance in all datasets (BACC-18: 93.45%, BAAD16: 95.02% & LD: 98.67%) with 250 ED. Due to the superiority of the Glove+CNN model, the subsequent analysis is performed with the 250 feature dimension only.

Table 10 shows the statistical summary of the GloVe+CNN model concerning 250 embedding dimension. Author wise classification performance denotes by Precision, Recall and Micro F1-score. Results indicate that for BACC-18, author 10 obtained the highest precision (99.00%) and F1-score (98.00%) whereas author 13 achieved the lowest recall (62.00%) and F1-score (75.00%). In the BAAD16 dataset, the highest recall (99.00%) and F1-score (97%) are achieved for author 07. For the author 21, the approach obtained the highest precision, whereas author 03 achieved the lowest recall (64.00%) and F1-score (75.00%). On LD, a maximum of 100.00% performance measures (precision, recall and Micro F1-score) are obtained for the one author (01) whereas a lowest of 98.00% is achieved for authors 02 and 10 respectively. However, the average accuracy of the proposed method on the LD dataset is 98.67%. The LD dataset contains only three authors with a small training/testing set where a huge stylometric feature variations are observed between author 01 and the other two authors (02 & 10). Usually, most of the literature has been written by author 01 in Shadhu-bhasha whereas authors 02 and 10 written in Cholito-bhasha. These stylists variation carry out a huge impact in feature representation and thus author 01 achieved a better accuracy than others. The proposed GloVe+CNN architecture can detect this distinguishable features of the author's (01) text very well, which helps to predict all the texts in the test set correctly.

Figure 9 summarizes the performance of the proposed model (GloVe+CNN with 250 EM) on three datasets. The bar denotes the average precision, recall and Micro F1-score for BACC-18, BAAD16 and LD. The proposed model performed slightly better in LD than BACC-18 and BAAD16 (e.g., precision values increased 9.00% and 5.00% compared to BACC-18 and BAAD16). Due to the smaller number of authors (only three), LD has achieved a slightly better performance than BACC-18 (18 authors) and BAAD16 (16 authors). The maximum 7.6% precision, 13.43% recall and 7.54% Micro F1-score error are obtained on BACC-18 datasets. The minimum 1.15% precision, recall and Micro F1-score are achieved on LD.

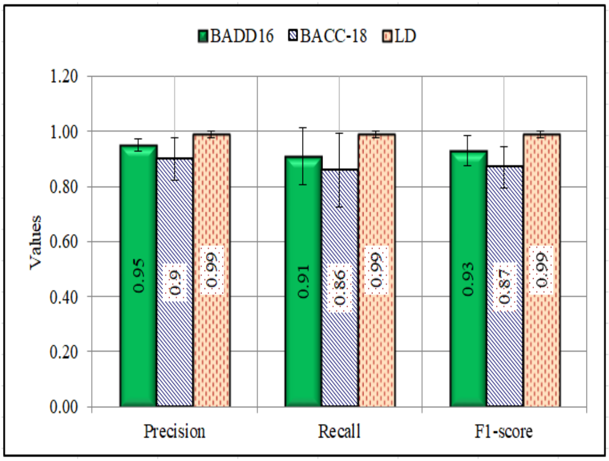


FIGURE 9. Performance of GloVe+CNN model on BACC-18, BAAD16 and LD. The error bar indicates the standard error.

D. COMPARISON WITH PREVIOUS APPROACHES

Table 11 shows the performance comparison concerning the proposed system's accuracy with the previously developed methods ([6], [30]), [1]) on BACC-18, BAAD16 and LD datasets.

Results indicate that the proposed method achieved the highest accuracy of 98.67% than presented in [6] (97.6%), [1] (94.66%) and [30] (98%) for LD. Concerning BAAD16, the proposed technique achieved the maximum accuracy of 95.2%, which is better than Khatun *et al.* [6] (79.44%), Lahiri *et al.* [1] (68%) and Phani *et al.* [30] (65.17%). Moreover, in the case of BACC-18, the proposed method obtained the highest accuracy of 93.46%, whereas the method in [6] obtained 76.82% and [1] gained 64.21%. Thus, for three datasets, the proposed method achieved the highest accuracy and outperformed the existing techniques.

E. COMPARISON WITH TRANSFORMER-BASED TECHNIQUES

In recent years, transformer-based approaches have gained increased attention among NLP researchers for the various text classification task. This work carried out a set of new experiments using transformer-based techniques on the

TABLE 9. Macro-F1 (M-F1), weighted-F1 (W-F1) and Accuracy (A) measures on three datasets.

		BACC-18 (%)			BAAD16 (%)			LD (%)		
Model	ED	M-F1	W-F1	A	M-F1	W-F1	A	M-F1	W-F1	A
GloVe+CNN	**100**	**81.00**	**87.00**	**86.98**	**87.00**	**91.00**	**91.08**	**97.00**	**96.00**	**96.67**
GloVe+SVM		61.00	64.00	64.21	64.00	68.00	68.37	94.00	95.00	94.66
Fasttext+CNN		80.00	85.00	85.04	85.00	89.00	88.79	97.00	96.00	96.67
Fasttext+SVM		62.00	64.00	64.10	65.00	70.00	69.88	95.00	95.00	94.67
Fasttext+SGD		63.00	67.00	66.95	65.00	71.00	71.17	95.00	95.00	94.67
Word2Vec+CNN		78.00	82.00	81.89	83.00	86.00	86.03	96.00	96.00	96.67
Word2Vec+SVM		58.00	61.00	60.97	62.00	67.00	67.06	93.00	93.00	93.33
Word2Vec+SGD		59.00	60.00	60.02	64.00	69.00	69.19	95.00	95.00	94.67
GloVe+SGD [68]		62.00	66.00	65.89	65.00	70.00	70.15	95.00	95.00	94.66
GloVe+CNN	**200**	**84.00**	**91.00**	**91.07**	**89.00**	**93.00**	**92.96**	**98.00**	**98.00**	97.60
Fasttext+CNN		82.00	90.00	89.97	88.00	92.00	92.03	98.00	**99.00**	**98.53**
Word2Vec+CNN		80.00	88.00	88.06	86.00	91.00	90.74	98.00	98.00	97.33
GloVe+CNN	**250**	**87.00**	**93.00**	**93.45**	**93.00**	**95.00**	**95.02**	**99.00**	**99.00**	**98.67**
GloVe+LSTM		85.00	90.00	89.90	89.00	92.00	92.02	97.00	98.00	98.13
Fattext+CNN		86.00	91.00	90.94	92.00	94.00	94.11	**99.00**	**99.00**	98.40
Fasttext+LSTM		85.00	91.00	91.09	90.00	93.00	93.12	98.00	98.00	98.13
Word2Vec+CNN		83.00	89.00	89.20	89.00	91.00	91.16	98.00	98.00	97.33
Word2Vec+LSTM		82.00	89.00	89.09	90.00	91.00	90.93	98.00	98.00	97.60

TABLE 10. Statistical measures summary of GloVe+CNN with 250 ED on three datasets).

Datasets	Author code	Precision(%)	Recall(%)	Micro F1-score (%)	support
BACC-18	01	98.00	94.00	96.00	179
	17	97.00	82.00	89.00	76
	19	78.00	90.00	84.00	20
	03	90.00	76.00	83.00	84
	02	94.00	97.00	96.00	720
	08	92.00	65.00	76.00	34
	09	90.00	95.00	92.00	57
	10	**99.00**	97.00	**98.00**	461
	11	91.00	91.00	91.00	47
	12	90.00	97.00	93.00	196
	18	**72.00**	**100.00**	84.00	42
	24	98.00	70.00	82.00	61
	05	84.00	70.00	77.00	61
	15	95.00	98.00	96.00	608
	16	81.00	**100.00**	90.00	22
	14	84.00	72.00	78.00	58
	07	84.00	98.00	90.00	97
	13	95.00	**62.00**	**75.00**	32
BAAD16	06	97.00	94.00	95.00	220
	01	93.00	88.00	90.00	112
	07	96.00	**99.00**	**97.00**	906
	04	95.00	78.00	86.00	93
	03	96.00	**61.00**	**75.00**	44
	22	98.00	95.00	96.00	95
	02	**91.00**	92.00	92.00	252
	23	95.00	94.00	95.00	210
	08	97.00	98.00	**97.00**	282
	25	94.00	92.00	93.00	177
	10	93.00	98.00	95.00	261
	11	94.00	98.00	96.00	169
	05	94.00	98.00	96.00	393
	20	94.00	92.00	93.00	155
	14	95.00	95.00	95.00	186
	21	**100.00**	78.00	88.00	37
LD	**01**	**100.00**	**100.00**	**100.00**	250
	02	**98.00**	**98.00**	**98.00**	250
	10	**98.00**	**98.00**	**98.00**	250

TABLE 11. Performance comparison of authorship attribution.

	Accuracy (%)		
Model	BAAD16	BACC-18	LD
Fasttext (Char)+CNN [6]	79.44	76.82	97.60
Bigrams-TF+NB [30]	65.17	62.44	98.00
GloVe+SVM [1]	68.00	64.21	94.66
Proposed (GloVe+CNN)	**95.02**	**93.45**	**98.67**

TABLE 12. Performance of transformer models in authorship classification.

Datasets	Techniques	Accuracy (%)
BACC-18	M-BERT	92.46
	Distil-BERT	91.06
	Proposed (GloVe+CNN)	**93.45**
BAAD16	M-BERT	94.09
	Distil-BERT	93.54
	Proposed (GloVe+CNN)	**95.02**
LD	M-BERT	98.00
	Distil-BERT	97.33
	Proposed (GloVe+CNN)	**98.67**

three corpora: BACC-18, BAAD16 and LD. Although many variants of transformer-based techniques are available, this work investigated two popular methods (m-BERT [51] and Distil-BERT [52]) for the authorship classification task. The pre-trained m-BERT and Distil-BERT hugging face models[7][8] are used to evaluate the performance of the authorship classification. All of the hyperparameters are used as default. Table 12 shows the performance of m-BERT and Distil-BERT models including the proposed model on the three datasets.

Results indicate that m-BERT achieved a higher accuracy of 1.4% than Distil-BERT for the BACC-18 dataset due to the better semantic feature representation at word label with a lower 40% trainable parameters.[9]

On the other hand, the m-BERT model achieved a lower accuracy than the proposed method for most cases.

[7] https://huggingface.co/bert-base-multilingual-cased,

[8] https://huggingface.co/transformers/model_doc/distilbert.html

[9] https://huggingface.co/transformers/model_doc/distilbert.html jsdgkjd-kaka

Specifically, the m-BERT model showed an approximately 0.99% lower accuracy compared to the proposed technique. Bengali author's texts usually are available in two morphological variants, such as Sadhu-bhasha and Cholito-bhasha. These variations are not considered in the pre-train transformer-based models (i.e., m-BERT, Distil-BERT). Thus, the pre-trained models suffers from handling the author's text written in Shadhu-bhasha form (such as authors 01 and 10 usually written in Shadhu-bhasha). However, the developed corpus contained authors' texts of both forms. The previous study also revealed that the pre-trained transformer-based models do not perform well for the low-resource language [67], including the Bengali, due to its limited unique words, out-of-vocabulary problem and incapability to handle Bengali morphological variants. Although further investigations with other transformer-based techniques (such as Bangla-BERT, XLM and ELECTRA) are deemed necessary, the preliminary results revealed that the proposed model outperformed other techniques, including transfer-based techniques (m-BERT and Distil-BERT).

F. ERROR ANALYSIS

It is evident from Tables 9 and 10 that GloVe+CNN is the best performing model to classify text documents for the Bengali language. A detail error analysis is carried out using the confusion matrix (CM) to investigate more insights into the individual author's class performance.

Tables 13, 14 and 15 illustrate the confusion matrix for BACC-18, BAAD16 and LD datasets respectively. The diagonal cells of CM denote the correctly classified numbers. The dark pink cells represent the maximum incorrectly classified numbers, whereas the light pink indicates the multiple incorrectly classified number for a particular author class.

Concerning BACC-18, among 32 texts of author 13, 20 texts correctly classified and 12 texts are incorrectly classified (e.g., maximum error) whereas all texts of author 18 and author 16 are correctly classified (e.g., minimum error). In BAAD16, among 906 texts of author 07, there are only 13 texts incorrectly classified, which is the minimum misclassified class. Among 44 texts of author 03, 16 texts are classified incorrectly, indicating the maximum misclassification rate of BAAD16. Concerning LD, among 250 texts of author 02 and 10, five texts are incorrectly classified. On the other hand, all texts (250) of author 01 classified correctly.

The confusion matrix concluded that a minimum error rate (0.00%) is achieved for the authors 18 and 16 on BACC-18 corpus. On the other hand, on the BAAD16 corpus, a minimum of 1.00% error rate is obtained for the author 07 whereas the author class 01 gained the lowest (0.00%) error rate on LD corpus.

Fig. 10 shows the classifier performance with actual and predicted labels on four sample texts. The 'Yes' in the remarks column indicates that the proposed approach could correctly classify, whereas the word 'No' represents the classification failure. Input 1 considered from BACC-18, and LD, which are correctly predicted by the proposed approach (GloVe+CNN). However, GloVe+LSTM, Word2Vec+LSTM, and GloVe+SVM models have failed to classify input 1. Inputs 2 and 3 are taken from LD and BADD datasets where all models have failed to predict the actual author class. The proposed method (GloVe+CNN) and LSTM+GloVe can correctly predict the input text 4 whereas Word2Vec+LSTM and GloVe+SVM cannot predict. The error analysis concludes that the proposed approach has some limitations, such as if the text's size is too short or too long, it cannot predict the correct label. The document level similarity between semantics and syntactic is also responsible for the misclassification.

G. DISCUSSION

The proposed technique of deep neural networks is based on a series of empirical investigations over a period of time involving extensive experimental evaluations (Sections VII-A and VII-C). The analysis of these investigations led to the final development of the GloVe+CNN model for the authorship classification task in Bengali, which achieved the highest performance compared to the other combinations.

Results revealed that the GloVe model achieved the better semantic and syntactic performance according to intrinsic evaluations (Table 8). The Word2Vec and FastText embeddings cannot carry out the useful syntactic feature due to a shortage of word-word co-occurrence information, whereas the GloVe technique overcomes these limitations. The proposed classifier (Glove + CNN) is compared with other classifiers, including SVM, SGD, and LSTM, with three embedding techniques. The classification results shown in Table 9 indicate that the statistical classification techniques (SVM or SGD) failed to perform better due to the lack of local and global features and many classes. The SVM and SGD have achieved good performance for LD datasets, but when the number of authors increased in BACC-18 and BAAD16, the classification performance fell abruptly about 30%−35%. Thus, the number of authors has an immense impact on authorship classification task that can not be overcome by the SVM and SGD techniques.

It is evident that the proposed model achieves a higher accuracy if BACC-18 is used for the word embedding. Empirical investigation showed that the overall classification performance (i.e., accuracy) increases from 93.45% to 95.62% on the BACC-18 dataset if the same corpus (i.e., BACC-18) is used for embedding and classification models. This improvement of accuracy occurs due to the joint distribution of data in embedding and classification models. However, the classification performance will reduce for an exact author text which lies outside of BACC-18. We further investigated the accuracy of the proposed model with author 02 data of the BAAD16 corpus for better insight. Using the same corpus, author 02 achieved an accuracy of 86.11% on the BAAD16 test set, whereas the same author 02 obtained 95.62% accuracy on the BACC-18 test set. On the other hand, as regards to the different corpora, author 02 obtained 92.46%

TABLE 13. Confusion matrix of GloVe+CNN model with 250 ED for BACC-18.

CM	01	17	19	03	02	08	09	10	11	12	18	24	05	15	16	14	07	13
01	169	0	0	2	7	0	1	0	0	0	0	0	0	0	0	0	0	0
17	0	62	0	1	0	0	0	0	0	0	0	0	1	9	0	1	2	0
19	0	0	18	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
03	1	0	0	64	5	2	0	0	2	2	3	0	0	3	1	0	1	0
02	0	0	0	0	697	0	0	6	0	0	0	1	0	14	0	0	2	0
08	0	0	0	0	0	22	0	0	0	0	2	0	3	0	0	2	5	0
09	1	0	0	0	1	0	54	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	8	0	2	449	0	0	0	0	0	2	0	0	0	0
11	0	0	0	0	1	0	0	0	43	0	1	0	0	0	0	1	1	0
12	0	0	0	0	2	0	0	0	0	190	0	0	2	0	0	2	0	0
18	0	0	0	0	0	0	0	0	0	0	42	0	0	0	0	0	0	0
24	0	0	0	3	3	0	1	0	1	5	2	43	0	2	0	0	0	1
05	0	1	1	0	0	0	0	0	0	10	3	0	43	0	0	0	3	0
15	2	0	1	0	8	0	0	0	0	0	0	0	1	593	1	0	2	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0
14	0	1	1	0	2	0	1	0	1	2	5	0	0	0	1	42	2	0
07	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	95	0
13	0	0	2	1	3	0	0	0	0	3	0	0	1	0	2	0	0	20

TABLE 14. Confusion matrix of GloVe+CNN (ED=250) model for BAAD16.

CM	06	01	07	04	03	22	02	23	08	25	10	11	05	20	14	21
06	207	0	12	0	0	0	0	0	0	0	0	0	1	0	0	0
01	0	98	0	0	1	0	7	0	0	4	2	0	0	0	0	0
07	5	1	893	0	0	0	1	0	1	0	0	0	3	0	2	0
04	0	0	4	73	0	0	2	1	3	1	3	0	1	5	0	0
03	1	0	2	1	27	0	2	2	0	0	5	0	1	2	1	0
22	0	0	1	0	0	90	0	0	1	1	0	0	0	1	1	0
02	1	1	2	2	0	0	233	0	0	3	6	1	2	1	0	0
23	0	1	1	0	0	1	4	198	0	0	0	1	3	0	1	0
08	0	0	3	0	0	0	0	1	275	0	0	0	3	0	0	0
25	0	3	2	0	0	0	1	0	1	162	1	6	0	0	1	0
10	0	1	0	1	0	0	3	0	0	0	256	0	0	0	0	0
11	0	0	1	0	0	0	0	1	0	0	0	166	1	0	0	0
05	0	0	2	0	0	1	1	1	1	0	0	0	386	0	1	0
20	0	0	1	0	0	0	1	3	0	1	3	1	2	143	0	0
14	0	0	2	0	0	0	0	1	0	1	0	0	5	0	177	0
21	0	0	3	0	0	0	0	0	1	0	0	1	1	0	2	29

TABLE 15. Confusion matrix of GloVe+CNN (ED=250) model for LD.

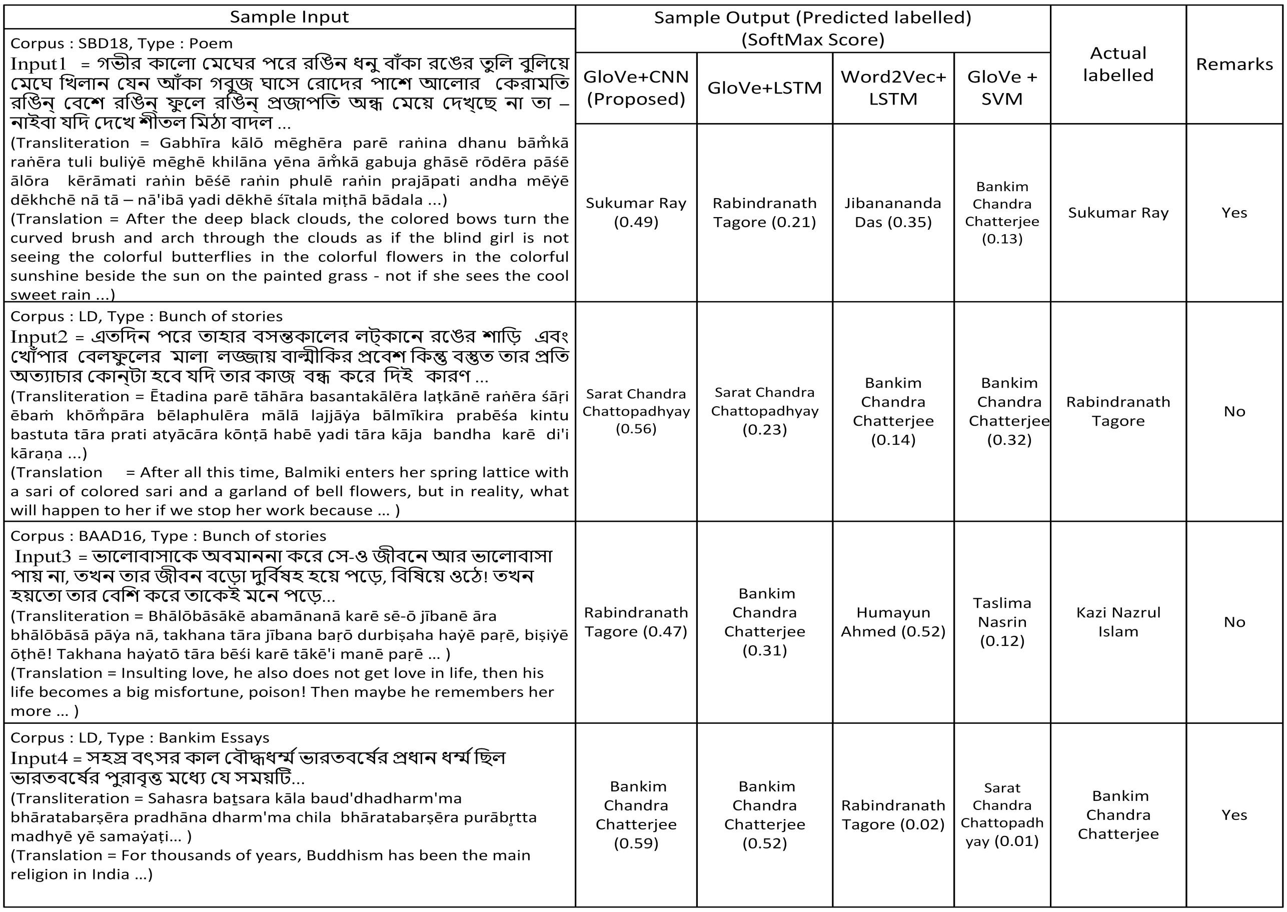
CM	01	02	10
01	250	0	0
02	0	245	5
10	0	5	245

accuracy on the BAAD16 test set and 93.45% accuracy for the BACC-18 test set. Therefore, based on the previous survey [34], the proposed research uses different corpora for the embedding model generation and classification task to ensure diverse data distributions.

The confusion matrix deliberates a good statistic about the within authors misclassification. Error analysis shows that there are a large number of misclassification between the authors when both authors are writing in similar topics and same decades (such as author 02 and 15). The author wise classification performance (e.g. Precision, Recall and Micro F1-score) shown in Table 10 indicates that the highest recall value obtains for authors 01, 16 and 18 due to theirs distinguished writing styles and topics. Therefore, the analysis revealed that Bengali authorship attribution performance depends on the number of authors, their writing styles, topics, types of embedding models, model hyperparameters, and training samples.

The 100 fake texts with a category labelled as "Others (Code: 00)" for further investigation of the proposed model. The maximum expected value forces to select the corresponding author name. In these scenarios, a fake text can also select the author name among the existing 18 authors. So, the fake text is likely to increase the misclassification rate. As a solution for such an erroneous this situation, rank-based expected value [68] with a threshold is proposed to resolve this problem. Now the 100 expected-value scores are calculated using Eq. 6 for a given author (a) and its corresponding feature values (Z).

The expected function returns a total of 18 author scores and the maximum expected value corresponding the index is defined for the author name. From these expected values, it selects a value such that the false positive and true-positive rate is the maximum. The selected value is called the threshold value. Any expected value that is larger than the threshold value will belong to one of the 18 authors, and any value below the threshold value belongs to the 00 category. The



Sample Input	Sample Output (Predicted labelled) (SoftMax Score)				Actual labelled	Remarks
	GloVe+CNN (Proposed)	GloVe+LSTM	Word2Vec+ LSTM	GloVe + SVM		
Corpus : SBD18, Type : Poem Input1 = গভীর কালো মেঘের পরে রঙিন ধনু বাঁকা রঙের তুলি বুলিয়ে মেঘে খিলান যেন আঁকা গবুজ ঘাসে রোদের পাশে আলোর কেরামতি রঙিন্ বেশে রঙিন্ ফুলে রঙিন্ প্রজাপতি অন্ধ মেয়ে দেখ্ছে না তা – নাইবা যদি দেখে শীতল মিঠা বাদল ... (Transliteration = Gabhīra kālō mēghēra parē raṅina dhanu bām̐kā raṅēra tuli buliẏē mēghē khilāna yēna ām̐kā gabuja ghāsē rōdēra pāśē ālōra kērāmati raṅin bēśē raṅin phulē raṅin prajāpati andha mēẏē dēkhchē nā tā – nā'ibā yadi dēkhē śītala miṭhā bādala ...) (Translation = After the deep black clouds, the colored bows turn the curved brush and arch through the clouds as if the blind girl is not seeing the colorful butterflies in the colorful flowers in the colorful sunshine beside the sun on the painted grass - not if she sees the cool sweet rain ...)	Sukumar Ray (0.49)	Rabindranath Tagore (0.21)	Jibanananda Das (0.35)	Bankim Chandra Chatterjee (0.13)	Sukumar Ray	Yes
Corpus : LD, Type : Bunch of stories Input2 = এতদিন পরে তাহার বসন্তকালের লট্কানে রঙের শাড়ি এবং খোঁপার বেলফুলের মালা লজ্জায় বাল্মীকির প্রবেশ কিন্তু বস্তুত তার প্রতি অত্যাচার কোন্টা হবে যদি তার কাজ বন্ধ করে দিই কারণ ... (Transliteration = Ētadina parē tāhāra basantakālēra laṭkānē raṅēra śāṛi ēbaṁ khōm̐pāra bēlaphulēra mālā lajjāẏa bālmīkira prabēśa kintu bastuta tāra prati atyācāra kōnṭā habē yadi tāra kāja bandha karē di'i kāraṇa ...) (Translation = After all this time, Balmiki enters her spring lattice with a sari of colored sari and a garland of bell flowers, but in reality, what will happen to her if we stop her work because ...)	Sarat Chandra Chattopadhyay (0.56)	Sarat Chandra Chattopadhyay (0.23)	Bankim Chandra Chatterjee (0.14)	Bankim Chandra Chatterjee (0.32)	Rabindranath Tagore	No
Corpus : BAAD16, Type : Bunch of stories Input3 = ভালোবাসাকে অবমাননা করে সে-ও জীবনে আর ভালোবাসা পায় না, তখন তার জীবন বড়ো দুর্বিষহ হয়ে পড়ে, বিষিয়ে ওঠে! তখন হয়তো তার বেশি করে তাকেই মনে পড়ে... (Transliteration = Bhālōbāsākē abamānanā karē sē-ō jībanē āra bhālōbāsā pāẏa nā, takhana tāra jībana baṛō durbiṣaha haẏē paṛē, biṣiẏē ōṭhē! Takhana haẏatō tāra bēśi karē tākē'i manē paṛē ...) (Translation = Insulting love, he also does not get love in life, then his life becomes a big misfortune, poison! Then maybe he remembers her more ...)	Rabindranath Tagore (0.47)	Bankim Chandra Chatterjee (0.31)	Humayun Ahmed (0.52)	Taslima Nasrin (0.12)	Kazi Nazrul Islam	No
Corpus : LD, Type : Bankim Essays Input4 = সহস্র বৎসর কাল বৌদ্ধধর্ম্ম ভারতবর্ষের প্রধান ধর্ম্ম ছিল ভারতবর্ষের পুরাবৃত্ত মধ্যে যে সময়টি... (Transliteration = Sahasra baṯsara kāla baud'dhadharm'ma bhāratabarṣēra pradhāna dharm'ma chila bhāratabarṣēra purābr̥tta madhyē yē samaẏaṭi...) (Translation = For thousands of years, Buddhism has been the main religion in India ...)	Bankim Chandra Chatterjee (0.59)	Bankim Chandra Chatterjee (0.52)	Rabindranath Tagore (0.02)	Sarat Chandra Chattopadhyay (0.01)	Bankim Chandra Chatterjee	Yes

FIGURE 10. **Sample input/output with model performance.**

thresholding technique will likely reduce the misclassification rate of the fake text. In our work, 100 fake texts are considered for the experimental purpose only. The threshold value changes accordingly if the number of fake text documents is varied. In particular, empirical investigation revealed the best threshold values of 0.008113 for 100 fake texts.

Availability of benchmark corpus is a crucial constituent in developing any automatic text classification approach. Bengali is considered a resource-poor languages due to the unavailability of the benchmark dataset and other related resources. Thus, to develop an automatic Bengali authorship classification approach, this research had to develop two corpora: (i) an embedding corpus (EC) (ii) authorship classification corpus (BACC 18) with 18 author classes. Usually, Bengali author's text available in two morphological variants: Sadhu-bhasha and Cholito-bhasha. Thus, to develop an automatic text classification approach in Bengali for real-world applications, these variants should consider while building the embedding and training models. Transformer-based models (such as m-BERT, distil-BERT, Bangla-BERT) pre-trained with 104 mono lingual datasets, including Bengali Wikipedia. However, Wikipedia dataset considered only one variant (i.e. Cholito-bhasha) of Bengali text. On the other hand, the developed corpora considered both forms of Bengali text. For example, the author 01 has written in Shadhu-bhasha form while the author 07 written in Cholito-bhasha form. Thus, the research community and language industries working on developing Bengali language processing tools can use the developed corpora and proposed technique for text classification[10] Although the proposed technique performed better than the transformer-based approaches, a detailed investigations with other transformer-based techniques (such as Bangla-BERT, XLM, ELECTRA) should be carried out in future for generalization of the authorship classification task performance in Bengali.

VIII. CONCLUSION

This paper introduced an authorship classification technique for a language with little to no resources (like Bengali). To perform the classification task, an embedding corpus has developed. Three embedding techniques (Word2Vec, FastText and GloVe) with CNN, LSTM, SVM and SGD classifiers have been investigated on three datasets. Results revealed that CNN with the GloVe model outperformed the

[10] Code available at https://github.com/mrhossain/Bengali-Authorship-Classification.

other classification techniques on BACC-18, BAAD16 and LD datasets. The proposed model achieved the best results compared to the existing techniques. Future improvements may include meta-embedding (e.g., addition, concatenation and dynamic) and context-based feature extraction techniques (e.g., BERT, XLM, GPT-2). Moreover, the proposed approach can be explored with multi-domain (e.g., Facebook, Blogs, Twitter, Quora and Stack-overflow), and code mixed datasets for improved performance.

REFERENCES

[1] S. Phani, S. Lahiri, and A. Biswas, "A supervised learning approach for authorship attribution of Bengali literary texts," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 16, no. 4, pp. 1–15, Sep. 2017.

[2] T. Chakraborty, "Authorship identication using stylometry analysis in Bengali literature," 2012, *arXiv:1208.6268*. [Online]. Available: https://arxiv.org/abs/1208.6268

[3] K. Luyckx, W. Daelemans, and E. Vanhoutte, "Stylogenetics: Clustering-based stylistic analysis of literary corpora," in *Proc. LREC*, Genoa, Italy, 2006, pp. 30–35.

[4] M. T. Hossain, M. M. Rahman, S. Ismail, and M. S. Islam, "A stylometric analysis on Bengali literature for authorship attribution," in *Proc. ICCIT*, Dhaka, Bangladesh, 2017, pp. 1–5.

[5] P. Das, R. Tasmim, and S. Ismail, "An experimental study of stylometry in Bangla literature," in *Proc. EICT*, Cox's Bazar, Bangladesh, 2015, pp. 575–580.

[6] A. Khatun, A. Rahman, M. S. Islam, and M. E. Jannat, "Authorship attribution in Bangla literature using character-level CNN," in *Proc. ICCIT*, Dhaka, Bangladesh, 2019, pp. 1–5.

[7] S. Ruder, P. Ghaffari, and J. G. Breslin, "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," *arXiv:1208.6268*. [Online]. Available: https://arxiv.org/abs/1208.6268

[8] N. Islam, M. M. Hoque, and M. R. Hossain, "Automatic authorship detection from Bengali text using stylometric approach," in *Proc. ICCIT*, Dhaka, Bangladesh, 2017, pp. 1–6.

[9] D. M. Anisuzzaman, B. EngineeringDhaka, and A. Salam, "Authorship attribution for Bengali language using the fusion of N-gram and Naïve Bayes algorithms," *Int. J. Inf. Technol. Comput. Sci.*, vol. 10, no. 10, pp. 11–21, Oct. 2018.

[10] U. Pal, A. S. Nipu, and S. Ismail, "A machine learning approach for stylometric analysis of Bangla literature," in *Proc. ICCIT*, Dhaka, Bangladesh, 2017, pp. 1–5.

[11] G. Rakshit, A. Ghosh, P. Bhattacharyya, and G. Haffari, "Automated analysis of Bangla poetry for classification and poet identification," in *Proc. ICON*, Trivandrum, India, 2015, pp. 247–253.

[12] M. ShaukatTamboli and R. S. Prasad, "Authorship analysis and identification techniques: A review," *Int. J. Comput. Appl.*, vol. 77, no. 16, pp. 11–15, Sep. 2013.

[13] E. Manjavacas, J. D. Gussem, W. Daelemans, and M. Kestemont, "Assessing the stylistic properties of neurally generated text in authorship attribution," in *Proc. Style-Var*, Copenhagen, Denmark, 2017, pp. 116–125.

[14] M. R. Hossain and M. M. Hoque, "Automatic Bengali document categorization based on deep convolution nets," in *Proc. ERCICA*, Bangalore, India, 2019, pp. 513–525.

[15] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1746–1751.

[16] M. R. Hossain and M. M. Hoque, "Semantic meaning based Bengali web text categorization using deep convolutional and recurrent neural networks (DCRNNs)," in *Proc. ICIoTCT*. Patna, India: Springer, 2021, pp. 494–505.

[17] F. J. Tweedie, S. Singh, and D. I. Holmes, "Neural network applications in stylometry: The federalist papers," *Comput. Humanities*, vol. 30, no. 1, pp. 1–10, 1996.

[18] A. Rocha, W. J. Scheirer, C. W. Forstall, T. Cavalcante, A. Theophilo, B. Shen, A. R. B. Carvalho, and E. Stamatatos, "Authorship attribution for social media forensics," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 5–33, Jan. 2017.

[19] X. Yang, G. Xu, Q. Li, Y. Guo, and M. Zhang, "Authorship attribution of source code by using back propagation neural network based on particle swarm optimization," *PLoS ONE*, vol. 12, pp. 1–18, Nov. 2017.

[20] A. Bander, D. Edwin, H. Richard, M. Spiros, and G. Rachel, "Source code authorship attribution using long short-term memory based networks," in *Proc. ESORICS*, Oslo, Norway, 2017, pp. 65–82.

[21] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Lang. Resour. Eval.*, vol. 45, no. 1, pp. 83–94, Jan. 2010.

[22] J. Kabala, "Computational authorship attribution in medieval latin corpora: The case of the Monk of Lido (ca. 1101–08) and Gallus Anonymous (ca. 1113–17)," *Lang. Resour. Eval.*, vol. 54, no. 1, pp. 25–56, Oct. 2018.

[23] S. Zafar, M. U. Sarwar, S. Salem, and M. Z. Malik, "Language and obfuscation oblivious source code authorship attribution," *IEEE Access*, vol. 8, pp. 197581–197596, Oct. 2020.

[24] R. Sarwar, N. Urailertprasert, N. Vannaboot, C. Yu, T. Rakthanmanon, E. Chuangsuwanich, and S. Nutanong, "*CAG*: Stylometric authorship attribution of multi-author documents using a co-authorship graph," *IEEE Access*, vol. 8, pp. 18374–18393, Jan. 2020.

[25] W. Anwar, I. S. Bajwa, M. A. Choudhary, and S. Ramzan, "An empirical study on forensic analysis of urdu text using LDA-based authorship attribution," *IEEE Access*, vol. 7, pp. 3224–3234, Dec. 2019.

[26] H. V. Agun and O. Yilmazel, "Incorporating topic information in a global feature selection schema for authorship attribution," *IEEE Access*, vol. 7, pp. 98522–98529, Jul. 2019.

[27] F. Ullah, J. Wang, S. Jabbar, F. Al-Turjman, and M. Alazab, "Source code authorship attribution using hybrid approach of program dependence graph and deep learning model," *IEEE Access*, vol. 7, pp. 141987–141999, Sep. 2019.

[28] M. Al-Sarem, F. Saeed, A. Alsaeedi, W. Boulila, and T. Al-Hadhrami, "Ensemble methods for instance-based arabic language authorship attribution," *IEEE Access*, vol. 8, pp. 17331–17345, Jan. 2020.

[29] A. Neocleous and A. Loizides, "Machine learning and feature selection for authorship attribution: The case of mill, Taylor mill and Taylor, in the nineteenth century," *IEEE Access*, vol. 9, pp. 7143–7151, Dec. 2021.

[30] S. Phani, S. Lahiri, and A. Biswas, "Authorship attribution in Bengali language," in *Proc. ICON*, Trivandrum, India, 2015, pp. 100–105.

[31] S. Lahiri, S. Phani, and A. Biswas, "A supervised authorship attribution framework for Bengali language," *arXiv:1607.05650*. [Online]. Available: https://arxiv.org/abs/1607.05650

[32] S. Phani, S. Lahiri, and A. Biswas, "A machine learning approach for authorship attribution for Bengali blogs," in *Proc. IALP*, Tainan, Taiwan, 2016, pp. 271–274.

[33] J. Santos, B. Consoli, and R. Vieira, "Word embedding evaluation in downstream tasks and semantic analogies," in *Proc. LREC*, Marseille, France, 2020, pp. 4828–4834.

[34] B. Wang, A. Wang, F. Chen, Y. Wang, and C.-C.-J. Kuo, "Evaluating word embedding models: Methods and experimental results," *APSIPA Trans. Signal Inf. Process.*, vol. 8, pp. 1–14, Jun. 2019.

[35] A. Khatun, A. Rahman, and M. S. Islam, "Authorship attribution dataset in Bangla," Shahjalal Univ. Sci. Technol., Sylhet, Bangladesh, Tech. Rep., Dec. 2020.

[36] NLTR. *Society for Natural Language Technology Research (SNLTR)*. Accessed: Jul. 12, 2018. [Online]. Available: https://www.nltr.org/

[37] Ebanglalibrary. *Bangla Book*. Accessed: Aug. 20, 2018. [Online]. Available: https://www.ebanglalibrary.com/

[38] GitHub. *Stylogenetics*. Accessed: Aug. 7, 2018. [Online]. Available: https://github.com/olee12/Stylogenetics

[39] Taslimanasrin. *Taslima Nasreen*. Accessed: Jan. 12, 2020. [Online]. Available: https://www.taslimanasrin.com/category/bangla-blogs/

[40] Banglapdfbooksblog Wordpress. *Bangla Pdf Books Blog*. Accessed: Jan. 14, 2020. [Online]. Available: https://banglapdfbooksblog.wordpress.com/

[41] Goodreads. *Jibanananda Das*. Accessed: Jan. 16, 2020. [Online]. Available: https://www.goodreads.com/author/quotes/638250.Jibanananda_Das

[42] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1532–1543.

[43] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, pp. 1–12, *arXiv:1301.3781*. [Online]. Available: https://arxiv.org/abs/1301.3781

[44] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[45] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, Sardinia, Italy, vol. 9, 2010, pp. 249–256.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[47] I. V. Tetko, D. J. Livingstone, and A. I. Luik, "Neural network studies. 1. Comparison of overfitting and overtraining," *J. Chem. Inf. Comput. Sci.*, vol. 35, no. 5, pp. 826–833, 1995.

[48] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, Atlanta, GA, USA, vol. 28, 2013, pp. 1–6.

[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[50] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747*. [Online]. Available: https://arxiv.org/abs/1609.04747

[51] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" in *Proc. ACL*, Florence, Italy, Jul. 2019, pp. 4996–5001. [Online]. Available: http://arxiv.org/abs/1609.04747

[52] K. Jain, A. Deshpande, K. Shridhar, F. Laumann, and A. Dash, "Indic-transformers: An analysis of transformer language models for Indian languages," *CoRR*, vol. abs/2011.02323, Nov. 2020. [Online]. Available: https://arxiv.org/abs/2011.02323

[53] D. J. C. MacKay, "Hyperparameters: Optimize, or integrate out?" in *Fundamental Theories of Physics: Their Clarification, Development and Application*, vol. 62, G. R. Heidbreder, Ed. Dordrecht, The Netherlands: Springer, 1996, pp. 43–60, doi: 10.1007/978-94-015-8729-7_2.

[54] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Exp.*, vol. 6, no. 4, pp. 312–315, Jan. 2020.

[55] W. S. Ahmed and A. A. A. Karim, "The impact of filter size and number of filters on classification accuracy in CNN," in *Proc. CSASE*, Duhok, Iraq, 2020, pp. 88–93.

[56] S. Obla, X. Gong, A. Aloufi, P. Hu, and D. Takabi, "Effective activation functions for homomorphic evaluation of deep neural networks," *IEEE Access*, vol. 8, pp. 153098–153112, Aug. 2020.

[57] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv:1505.00853*. [Online]. Available: https://arxiv.org/abs/1505.00853

[58] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," in *Proc. CCDC*, Shenyang, China, 2018, pp. 1836–1841.

[59] A. Darwish, D. Ezzat, and A. E. Hassanien, "An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis," *Swarm Evol. Comput.*, vol. 52, Feb. 2020, Art. no. 100616.

[60] M. R. Hossain and M. M. Hoque, "Towards Bengali word embedding: Corpus creation, intrinsic and extrinsic evaluations," in *Proc. ICON*, Patna, India, 2020, pp. 1–7.

[61] D. Banik, A. Ekbal, and P. Bhattacharyya, "Statistical machine translation based on weighted syntax–semantics," *Sādhanā*, vol. 45, pp. 1–12, Jul. 2020, doi: 10.1007/s12046-020-01427-w.

[62] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Embeddings for word sense disambiguation: An evaluation study," in *Proc. ACL*, Berlin, Germany, 2016, pp. 897–907. [Online]. Available: https://www.aclweb.org/anthology/P16-1085

[63] D. Gupta, S. Suman, and A. Ekbal, "Hierarchical deep multi-modal network for medical visual question answering," *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113993.

[64] Z. Vitalii, S. Aleksandar, S. April, and H. Nils, "Correlation coefficients and semantic textual similarity," in *Proc. NACAL*, Minneapolis, MN, USA, 2019, pp. 951–962.

[65] P. Kapil and A. Ekbal, "A deep neural network based multi-task learning approach to hate speech detection," *Knowl.-Based Syst.*, vol. 210, Dec. 2020, Art. no. 106458.

[66] M. R. Hossain and M. M. Hoque, "Automatic Bengali document categorization based on word embedding and statistical learning approaches," in *Proc. ICME*, Rajshahi, Bangladesh, 2018, pp. 1–6.

[67] G. Gambino and R. Pirrone, "Investigating embeddings for sentiment analysis in Italian," in *Proc. NLAI@AIIA*, Rende, Italy, vol. 2521, 2019, pp. 1–10.

[68] Y. Yang, "A study of thresholding strategies for text categorization," in *Proc. SIGIR*, New Orleans, LA, USA, 2001, pp. 137–145.

MD. RAJIB HOSSAIN (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science and engineering from the Chittagong University of Engineering and Technology (CUET), Chittagong, Bangladesh, in 2011 and 2018, respectively, where he is currently pursuing the Ph.D. degree in computer science and engineering (CSE). He is currently a Senior Software Engineer with the Research and Development Team, Tiger IT Bangladesh Ltd. He is working on a number of projects in the field of computer vision and machine/deep learning for several years. His research interests include natural language processing, deep learning, computer vision, and document classification. He is a member of IEB, Bangladesh.

MOHAMMED MOSHIUL HOQUE (Senior Member, IEEE) received the Ph.D. degree from the Department of Information and Computer Sciences, Saitama University, Japan, in 2012. He is currently a Distinguish Professor with the Department of Computer Science and Engineering (CSE), Chittagong University of Engineering and Technology (CUET). He is also serving as the Dean of the Faculty of Electronics and Communication Engineering (ECE), CUET, and the Director of the CUET Natural Language Processing Laboratory. He published more than 135 publications in several international journals, book chapters, and conferences. His research interests include human robot/computer interaction, computer vision, and natural language processing. He is a fellow of the Institute of Engineers, Bangladesh. He was served as the Award Coordinator, from 2016 to 2017, a Conference Coordinator, from 2017 to 2018, the Vice-Chair (Technical) of IEEE Bangladesh Section, from June 2019 to 2021, the Award Coordinator of IEEE Computer Society Bangladesh Chapter, from 2017 to 2018, an Educational Activity Coordinator, in 2018, the IEEE RAS, Bangladesh Chapter, and the Vice-Chair (Activity), from 2018 to 2019. He was also served as the TPC Chair for IEEE r10 HTC 2017, ECCE 2019, and ACMI 2021, the TPC Co-Chair for ICISET 2018, IEEE TenSymp 2020, and ICREST 2021, the Publication Chair for IEEE WIECON-ECE 2018/2019 and IEEE TenSymp 2020, and a TPC member for several international conferences. He is serving as the Chair of IEEE Bangladesh section for July 2021.

M. ALI AKBER DEWAN (Member, IEEE) received the B.Sc. degree in computer science and engineering from Khulna University, Bangladesh, in 2003, and the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2009. From 2003 to 2008, he was a Lecturer with the Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Bangladesh, where he was an Assistant Professor, in 2009. From 2009 to 2012, he was a Postdoctoral Researcher with Concordia University, Montreal, QC, Canada. From 2012 to 2014, he was a Research Associate with the École de Technologie Supérieure, Montreal. He is currently an Associate Professor with the School of Computing and Information Systems, Athabasca University, Athabasca, AB, Canada. He has published more than 70 articles in high impact journals and conference proceedings. His research interests include artificial intelligence, affective computing, computer vision, data mining, information visualization, machine learning, biometric recognition, medical image analysis, and health informatics. He has served as an editorial board member, a chair/co-chair, and a TPC member for several prestigious journals and conferences. He received the Dean's Award and the Excellent Research Achievement Award for his excellent academic performance and research achievements during his Ph.D. studies in South Korea.

NAZMUL SIDDIQUE (Senior Member, IEEE) received the Dipl.-Ing. degree in cybernetics from TU Dresden, Germany, the M.Sc. degree in computer science from BUET, Bangladesh, and the Ph.D. degree in intelligent control from the Department of Automatic Control and Systems Engineering, The University of Sheffield, U.K. He is currently with the School of Computing, Engineering and Intelligent Systems, Ulster University. He has published more than 170 research articles in the broad area of computational intelligence, vehicular communication, robotics, and cybernetics. He has authored and coauthored five books published by John Wiley, Springer, and Taylor & Francis. His research interests include robotics, cybernetics, computational intelligence, nature-inspired computing, stochastic systems, and vehicular communication. He is a fellow of the Higher Education Academy and a member of different committees of IEEE SMCS. He has guest edited eight special issues of reputed journals on cybernetic intelligence, computational intelligence, neural networks, and robotics. He has been involved in organizing many national and international conferences and co-edited seven conference proceedings. He is on the Editorial Board of the *Nature Scientific Research*, *Journal of Behavioural Robotics*, *Engineering Letters*, *International Journal of Machine Learning and Cybernetics*, *International Journal of Applied Pattern Recognition*, and *International Journal of Advances in Robotics Research*, and the Editorial Advisory Board of the *International Journal of Neural Systems*.

MD. NAZMUL ISLAM (Graduate Student Member, IEEE) received the B.Sc. degree in computer science and engineering from the Chittagong University of Engineering and Technology (CUET), Bangladesh, in 2017. He is currently working as a Software Engineer with Tiger IT Bangladesh Ltd and a part time postgraduate student of the Department of Computer Science and Engineering, CUET. He has been working on different kinds of iOS applications ranging from IoT device controlling apps to end-to-end encrypted (E2EE) chat applications for more than three years. His research interests include Bangla language processing (BPL) and statistical machine learning (SML). He is a member of IEB, Bangladesh.

IQBAL H. SARKER (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, Australia, in 2018. He is currently working as a Faculty Member with the Department of Computer Science and Engineering, Chittagong University of Engineering and Technology. He is one of the Research Founder of the International AIQT Foundation, Switzerland. His professional and research interests include data science, machine learning, AI-driven computing, NLP, cybersecurity intelligence, and the IoT-smart city technologies. He has published a number of high impact journals and conferences proceedings, such as *Journal of Network and Computer Applications* (Elsevier, USA), the *Internet of Things* (Elsevier), *Journal of Big Data* (Springer Nature, U.K.), *Mobile Network and Applications* (Springer, Netherlands), *The Computer Journal*, (Oxford University Press, U.K.), IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, and IEEE ACCESS, USA, and conferences, such as IEEE DSAA, Canada, IEEE Percom, Greece, ACM Ubicomp, USA and Germany, ACM Mobiquitous, Australia, and PAKDD, Australia. He is a member of ACM.