Received 10 July 2023, accepted 21 July 2023, date of publication 28 July 2023, date of current version 4 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3299565

RESEARCH ARTICLE

An Empirical Study on Authorship Verification for Low Resource Language Using Hyper-Tuned CNN Approach

TALHA FAROOQ KHAN, WAHEED ANWAR, HUMERA ARSHAD, AND SYED NASEEM ABBAS
Department of Computer Science, Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan

Corresponding author: Waheed Anwar (waheed@iub.edu.pk)

ABSTRACT Authorship verification is a crucial process employed to determine the authorship of a given text by analyzing distinct aspects of the writer's style, such as vocabulary, syntax, and punctuation. This process has gained significant research attention in various domains, including intellectual property rights, plagiarism detection, cybercrime investigations, copyright infringement, and forensics. While extensive studies have been conducted on multiple languages worldwide, encompassing Western European languages like Italian and Spanish, as well as Asian languages such as Bengali and Chinese, the investigation of authorship verification in Urdu has been comparatively limited, despite its status as a prominent South Asian language. This limitation can be attributed to the intricate and distinctive morphology of Urdu, which necessitates specific methodologies that cannot be directly applied in the same manner as other languages. To bridge this gap, we propose an innovative approach for authorship verification in Urdu, leveraging Convolutional Neural Networks (CNNs) with three distinct hyper-tuned parameters: ADAM, SGD, and RMSProp. To facilitate the development of this approach, we have curated a new corpus called UAVC-22, specifically tailored for Urdu authorship verification. This corpus offers enhanced robustness in terms of authors' classes and unique words. We have developed 9 authorship verification models, utilizing three different text embedding techniques, namely Word2Vec, GloVe, and FastText, we have performed a comparative analysis with traditional machine learning models such as Support Vector Machines (SVM) and Random Forest to assess the superiority and efficacy of the CNN-based approach. The optimized CNN-ADAM model with FastText achieved the highest accuracy of 98% for the Urdu dataset UAVC-22.

INDEX TERMS Authorship verification, low resource language, natural language processing, deep learning.

I. INTRODUCTION

With the global proliferation of digital documents, the risk of identity theft has significantly increased. One prevalent form of identity theft is the "Email Scam," where scammers impersonate company owners or managers and deceive employees into carrying out fraudulent activities, such as money transfers. Another form of identity abuse occurs through the dissemination of bogus reviews. To combat these emerging threats, the analysis of writing styles has become an effective means of differentiating between authentic authors and imposters.

In such a scenario, we can compare the writing style of writers to identify whether the email is written by the true Author A or by any imposter B. Writing style depends on many features which reflect an individual's identity. In other words, each individual has a unique form of the language they speak or write, known as the idiolect. This idiolect is characterized by unusual word choices in printed and digital documents. In general, the Internet possesses a tremendous quantity of data that expands exponentially daily. Such a rapid expansion is accompanied by issues, such as stolen or misidentified information. In order to resolve these issues, authorship analysis methods are introduced.

Authorship analysis is a scientific discipline dedicated to exploring the relationship between writers and their works.

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

It operates on the assumption that an author's identity can be discerned through their distinctive style features, including vocabulary, word usage, syntax, and stylistic qualities. In recent years, authorship analysis has gained significant recognition and has been the focus of extensive research. Computational and statistical methods have been developed to identify the authorship of text writings based on writing style, encompassing aspects such as word choice, punctuation usage, unique grammatical errors, and even the utilization of emoticons in contemporary digital texts.

In recent years, authorship analysis has been recognized as an important topic in the field of research. A lot of previous research has been done on computational and statistical methods for identifying the authorship of text writings based on writing style, including word choice, punctuation usage, unique grammatical errors, and in more contemporary digital texts, the use of emoticons. One thriving community where these computational methods for authorship analysis are created and assessed is PAN. PAN is an international organization which offers annual scientific assessments for digital text forensics and Stylometry. PAN has organized a variety of shared projects over the years, which has helped to advance the field of authorship identification. Some important tasks include:

- AUTHORSHIP VERIFICATION: assesses whether a new document is written by a particular author using a set of previous works. (PAN shared task in 2013,2014,2015,2020,2021,2022-2023)
- STYLE CHANGE DETECTION : (PAN shared task in 2017,2018,2019,2020,2021,2022,2023
- AUTHORSHIP ATTRIBUTION Determine the document's author from a list of candidates. (PAN shared task in 2011-2012, 2018,2019)
- AUTHORSHIP PROFILING identifies an author profile by compiling characteristics gleaned from the author's published works. (PAN shared task in 2013,2014,2015,2016,2017,2018

Although authorship analysis has been the subject of numerous studies in the past and due to the availability of authorship corpora, feature extractors, and classification approaches, authorship verification is well-established research. The authorship verification problem requires determining whether or not two different documents, one of which is unknown (DU) and the other of which is known (DA), were written by the same author (A).

Authorship verification is a well-established research area for high-resource languages (such as English and other European languages), but due to the lack of linguistic resources and methods, it is a difficult assignment for a low-resource language like Urdu. However, due to structural divergences and changes in regional dialects, feature extraction and classification methods developed for high-resource languages cannot be easily applied to low-resource languages. This drew our attention to the fact that there is a lack of research on this issue, especially for the Urdu language.

There has been a lot of research into authorship verification over the years [1]. Stylometry traits are used to analyze individual writers' writing styles in this subject. Syntactic, lexical, structural, and content-specific features are all examples of Stylometric attributes [2]. Because the total number of features, especially when utilizing Stylometry, can exceed hundreds, feature selection must be accomplished before classification. Naive Bayes, decision trees, Markov chains, support vector machines, logistic regression, and neural networks are the most often used classifiers in Stylometry-based authorship verification models [3]. The text must be prepared and features extracted using these antiquated procedures. This necessitates a significant investment in computer power [4].

This study presents a deep learning technique for authorship verification on the Urdu corpus. The deep learning approach reflects the semantic properties of the text. It provides automatic feature selection, best classification model selection during training, and alleviation of model overfitting and underfitting issues. The contribution to this study are:

- Development of a new authorship verification Urdu Corpus containing 6,000 documents of 15 well-known authors with 400 articles per author. Which is more extensive than existing corpora.
- Generation of nine embedding models for Urdu authorship verification using a combination of three embedding techniques (Word2Vec, GloVe and FastText).
- Development of a novel architecture for the authorship verification problem based on the Convolutional Neural Network (CNN) model with hyper-tuned parameters and the inclusion of a Discriminator and Generator at the fully connected layer with a sigmoid function.
- It is the first-ever study for Urdu Text using a Hyper-tuned Convolutional Neural Network based on Generators and Discriminators to the best of our knowledge.

The rest of the paper is divided into Related Work (Section II), Corpus (Section III), Methodology (Section IV), Results and Discussion (Section V), and Conclusions (Section VI).

II. RELATED WORK

The research on authorship categorization has made significant headway in low-resource languages such as Arabic, Latin, and Bangali, in addition to high-resource languages such as English and other European languages. These languages include English and other European languages. Authorship classification in high-resource languages and authorship classification in low-resource languages are the two categories that may be derived from the prior study on authorship classification.

A. HIGH-RESOURCE LANGUAGE AUTHORSHIP CLASSIFICATION

Tweedle et al. [5] used the neural network technique in conjunction with stylometry to determine who the authors of

English literature were. An additional study conducted by Ruder and colleagues [6] revealed a character-level and multi-channel CNN model for the purpose of authorship identification in English texts. In addition, Rocha et al. [7] studied various machine learning techniques that are appropriate for use with small sample sets. A backpropagation-based particle swarm technique was utilized by Yeang et al. [8] to identify the author based on English source code. They analyzed lexical, structural, and syntactic feature metrics in order to determine the authors of 2,022 Java files, and they were successful in doing so with a 91.06% accuracy rate. In addition, Alsulami et al. [9] used Long Short-Term Memory (LSTM) for source code authorship classification using 200 source files from 10 different programmers, and they reached an accuracy of 85.00%.The correctness of computer-generated English text was assessed by Enrique et al. [10] by using a tried-and-true authorship identification method. For authorship classification in English literature, Koppel et al. [11] used simplistic similarity-based techniques and got a precision of 93.20% for 1,000 writers. With an accuracy of 99.86%, Kabala [12] created a computer method for authorship classification in a medieval Latin corpus using Bray-Curtis distance and logistic regression. For the authorship classification of source code in three programming languages, Zafar et al. [13] suggested a character-level CNN model incorporating keywords and stylistic factors and attained 84.94% accuracy. In [14] authors investigate the transferability of syntactic knowledge across languages using the multilingual BERT (mBERT) language model. They demonstrate that mBERT can effectively transfer syntactic information between English, Italian, and French, specifically focusing on the null-subject phenomenon in Italian. The results indicate that mBERT can accurately reconstruct dependency parse trees without language-specific training. This study highlights the practical implications of transferring syntactic knowledge and suggests further exploration of other linguistic phenomena and psycholinguistic paradigms.

B. LOW-RESOURCE LANGUAGE AUTHORSHIP CLASSIFICATION

By utilizing co-author information, [15] multi-authorship classification technique was able to classify 1,360 text documents with 76.92% accuracy. An F1 score of 92% was achieved for 6,000 text pieces in the Urdu language using a Latent Dirichlet Allocation (LDA) based text attribution approach [16]. Agun et al.'s [17] statistically based system for authorship text attribution was developed, and it was tested using datasets they had created themselves. Using 1,000 programmer source codes, Ullah et al.'s [18] elaborate scale attribution system is operational. On a small number of programming languages, this system achieved 99.00% accuracy by combining the TF-IDF feature with deep CNN learning. Al-Sarem et al.'s [19] ensemble technique-based Arabic authorship attribution system made use of a number of stylometric variables and was tested against datasets of self-created Fatwas. With the help of word n-grams, LDA, and a sqrt similarity approach, Anwar et al. [16] were able to attribute authorship to 6,000 Urdu newspaper articles and attain a 92% F1-score. An authorship classification method based on stylometric features and machine learning was created by Neocleous et al. [20]. Their research showed that on 27 essays, SVM and DT classifiers performed the best.

Support vector machines (SVM) beat other classifiers in a comparison of various machine learning techniques for Bengali authorship recognition reported by Chakraborty et al. [21]. In their assessment of various methods for identifying Bengali authorship, Tamboli et al. [22] discovered that n-gram-based characteristics had a 90% accuracy rate. In order to identify authors, Hossain et al. [23] used a stylometry and voting-based classification model, which resulted in an accuracy of 90.67% for a corpus of 700 blog posts. Anisuzzaman et al. [24] used the Naive Bayes technique to identify Bengali writers from a dataset of 107,380 words. On a dataset of 20 Bengali bloggers, Pal et al.'s [25] suggested Bengali authorship classification model, employing SVM and Naive Bayes, achieved accuracies of 90.74% (SVM) and 86.21% (Naive Bayes). Another study [26] classified the authorship of Bengali poetry using a multi-class SVM, with semantic and stylistic factors producing an accuracy of 92%. On a dataset of 3125 passages, Islam et al.'s [26] random forest algorithm successfully identified authors from Bengali literature with a 96% accuracy rate, outperforming Naive Bayes (62%) and decision tree (85%) classifiers. The use of a character-level convolutional neural network (CNN) for the attribution of Bengali authorship was suggested by Khatun et al. [27], but it was discovered that the performance of such a network deteriorated with an increase in the number of authors and sample texts. Phani et al. [28] presented a method for determining the authorship of Bengali text that combined n-gram features with information gain approaches (feature ranking). However, this strategy was only able to achieve an accuracy of 95-99% when applied to 3,000 Bengali text documents and three Bengali writers. M.Moshiul et al.'s [29] developed optimized CNN with GloVe model and get highest accuracy of 98.6% for Bangla text classification. Table 1 provides a concise overview of the most significant facets of more contemporary methods of authorship classification.

In their investigation, [30]explored the use of bilingual lexicons, specifically cross-lingual word embeddings (CLWEs), to enhance language models with limited textual training data. While CLWEs showed promising results in improving models across multiple languages, their effectiveness was limited in low-resource environments such as Yongning Na due to challenges related to tonal systems, polysemy, and lexicon size. In [31] authors propose a language-independent feature set for accurate cross-lingual authorship identification. Their method partitions documents into fragments, achieving 96.66% accuracy on a multilingual corpus. Their solution outperforms existing methods without relying on external resources.

TABLE 1. Comparative analysis of previous state-of-the-art techniques.

Reference	Approach Used	Language	Accuracy
Tweedie, Singh, and Holmes 1996 [5]	Stylometric	English	-
Ruder, Ghaffari, and Breslin 2016 [6]	CNN	English	-
Rocha et al. 2017 [7]	Ensemble	English	-
Yang et al. 2017 [8]	Back propagation	Java code	91%
Alsulami et al. 2017 [9]	LSTM	Programming	85%
Koppel, Schler, and Argamon 2011 [11]	Naïve Similarity	English	93%
Zafar et al. 2020 [13]	Char-CNN + KNN	Programming	85%
Al-Sarem et al. 2020 [19]	Stylometry	Arabic	99%
Anwar et al. 2018 [16]	LDA + Sqrt	Urdu	92%
Neocleous and Loizides 2021 [20]	SVM,DT,KNN	English	93%
Chakraborty 2012 [21]	SVM + Stylometry	Bengali	-
M. T. Hossain et al. 2018 [23]	Stylometry + Voting	Bengali	90%
Anisuzzaman and Salam 2018 [24]	Ngram	Bengali	90%

It is absolutely necessary for there to be accessible relevant corpora in order for natural language processing systems to be developed. When it comes to the creation of authorship classification algorithms for Urdu, the absence of an author dataset is a significant challenge that must be overcome. Furthermore, the language that is being utilized plays a significant role in the process of determining dominant characteristics. When attempting to extract syntactic and semantic features from language corpora, it is common practice to make use of embedding models. However, since the characteristics of each language are unique from those of the others, embedding models that were created for one language cannot readily be extended to another language because of the differences in the attributes of each language. In order to find a solution to these problems, the research presented here advises developing an embedding model that is tailored particularly to the requirements of the Urdu language.

III. CORPUS

The dataset used by Waheed et al. [29], a renowned Urdu columnist, has made significant contributions to authorship attribution in the Urdu language. In order to conduct authorship verification, we adopted a similar approach to create an Urdu dataset, following the technique employed by Waheed et al. [29]. To establish a benchmark corpus, we extensively examined various websites and blogs. One crucial criterion for data extraction was that the content should be available in a digital text format rather than in JPEG format. After thorough analysis, we identified a particular list of websites that proved to be an ideal source due to their extensive collection of documents and diverse authors.

http://www.express.pk
http://www.dunya.com.pk
http://www.dunyapakistan.com
http://www.nawaiwaqt.com.pk

A. ETHICAL CONSIDERATIONS AND DATA COLLECTION REQUIREMENTS

In our study, we took careful considerations regarding the ethical and legal aspects of using someone's data for research purposes. The data we utilized was already available publicly, implying that implicit consent was provided by the authors. Moreover, we went the extra mile by contacting all the authors individually to obtain explicit permission to use their columns in our study.

Given the vast number of authors with online columns, we focused solely on columns published in newspapers within the scope of our research. To gain a comprehensive understanding of an author's writing style, it was crucial to have a substantial writing sample. Therefore, we established a minimum requirement of 400 articles for an author to be included in the candidate's list. Additionally, we set a minimum article length of 100 words for inclusion in the corpus. No specific constraints were imposed on the collection of columns in terms of topics, gender, or age. The collected columns encompass a wide range of topics, representing whatever the authors had published. It was imperative for our research to maintain a diversified and realistic nature, ensuring that the collected data remains unbiased and free from any inherent biases.

B. DATA COLLECTION

Our study focused on collecting data from Urdu newspapers to create a benchmark corpus for author verification in the Urdu language. To accomplish this, we proposed two approaches for the data collection process: a manual approach and an automatic approach.

In the manual approach, we compiled a list of regular Urdu columnists by examining mainstream Urdu newspapers in Pakistan, including Express, Nawa-e-waqat, Dunyapakistan, and Dunya. We reached out to these columnists via telephone and email, requesting them to share their columns for our research. Unfortunately, only 26% of the correspondents responded. It's worth noting that the majority of the data we received through this method was in JPG image format, rather than digital text. Additionally, some data was in page file format. To convert the collected images into a usable format, we utilized image processing and Optical Character Recognition (OCR) software. However, the output generated by these methods was not satisfactory, as it failed to produce an exact copy of the original text. In the automatic approach, we developed custom scripts in PHP to collect data from leading newspaper websites and blogs. This automated process allowed us to extract digital text directly from these sources.

To initiate the data collection process, we adopted a semi-automatic approach by manually browsing through the authors' columns and storing their respective URLs, column titles, column contents, and access times in a database. Over a period of ten working days, we successfully collected eight hundred columns. However, this approach proved to be time-consuming and labor-intensive.

To expedite the data collection procedure, we developed webpage scraping scripts using PHP language for each newspaper, as the webpage structures varied among them. This transition to the automatic approach allowed for a more streamlined and efficient process. The list of newspapers used for data collection in the semi-automatic and automatic approaches remained the same as in the manual approach, except for the exclusion of Jang newspaper, as all of its online data was in JPG image format. In the first step of the automatic approach, we compiled a URL list containing links to the column repositories of authors whose columns were available online on their respective newspaper websites. This list was prepared using a semi-automatic procedure, where we extensively explored the websites to identify the columnists and their column URLs, which were then stored in the database.

Next, we developed a web crawler and web scraper in PHP language to automatically extract the relevant data from these URLs. The data extraction process consisted of two steps. In the first step, the crawler extracted all the URLs associated with a specific author, and in the second step, the webpage scraper utilized these URLs to extract the complete column contents. Initially, we collected a substantial number of documents, totaling over 21,918, from these newspaper websites.

It is important to note that the columns were downloaded in the exact form in which they were initially published in the newspapers. No additional content was added or deleted from the data. Additionally, to ensure meaningful and analyzable content, we set a minimum article length requirement of 100 words for inclusion in the corpus. This criterion was established to facilitate the extraction of relevant stylistic and content-based features from the documents. There was no specific limit on the maximum number of words in an article, as more information about writing style and word structuring contributes to the training of a better model and yields more accurate predictions and results. However, we ensured that articles were at least 85 words long to extract relevant content-based and stylistic elements.

In order to maintain a balanced dataset, we selected 6,000 columns contributed by 15 authors, with each author providing 400 columns. We named this corpus UAVC-22. Further information, including the names of all the columnists, the total word count in their columns, and the average number of words per column, can be found in Table 2.

TABLE 2. Distribution of 6,000 UAVC-2022 documents.

Sr. No	Name	Articles	Words	Avg. words
1	I.Ansari	400	158309	396
2	A.A Khanzada	400	484256	1211
3	K.I Ullah	400	345903	865
4	Dr M.A Niazi	400	471024	1178
5	Dr T.A Khan	400	526201	1316
6	A.Q Hassan	400	418265	1046
7	H.U Rashid	400	534802	1337
8	I.A Arif	400	571798	1430
9	A.U Ghalib	400	474673	1187
10	J.Chaudhary	400	676141	1690
11	N.Naji	400	590991	1478
12	K.Nadeem	400	511401	1279
13	N.Raza	400	268674	672
14	Z.Hina	400	603032	1508
15	Q.Nizami	400	501933	1255

C. DATA PREPROCESSING

The data that is gathered by web crawling frequently contains a wide variety of unnecessary letters, symbols, and mathematical formulas that are unable to be transformed into UTF-8 format. For this reason, each text file that is gathered needs to go through a series of specialized pre-processing processes before it can be used. These steps are as follows:

- Eliminating all alphabets and digits that are not used in Urdu.
- Eliminating regular expressions and symbols by replacing them with a single blank space in order to remove them.
- Removing HTML elements, hashtags, and URLs, as well as any punctuation and white space that is not necessary.
- A single new line is used in place of several new lines.
- Removing any redundant text.

IV. METHODOLOGY

A. MODEL DESCRIPTION

We have proposed a novel architecture based on the Convolutional Neural Network (CNN) model with hyper tuned parameters and the inclusion of a Discriminator and Generator

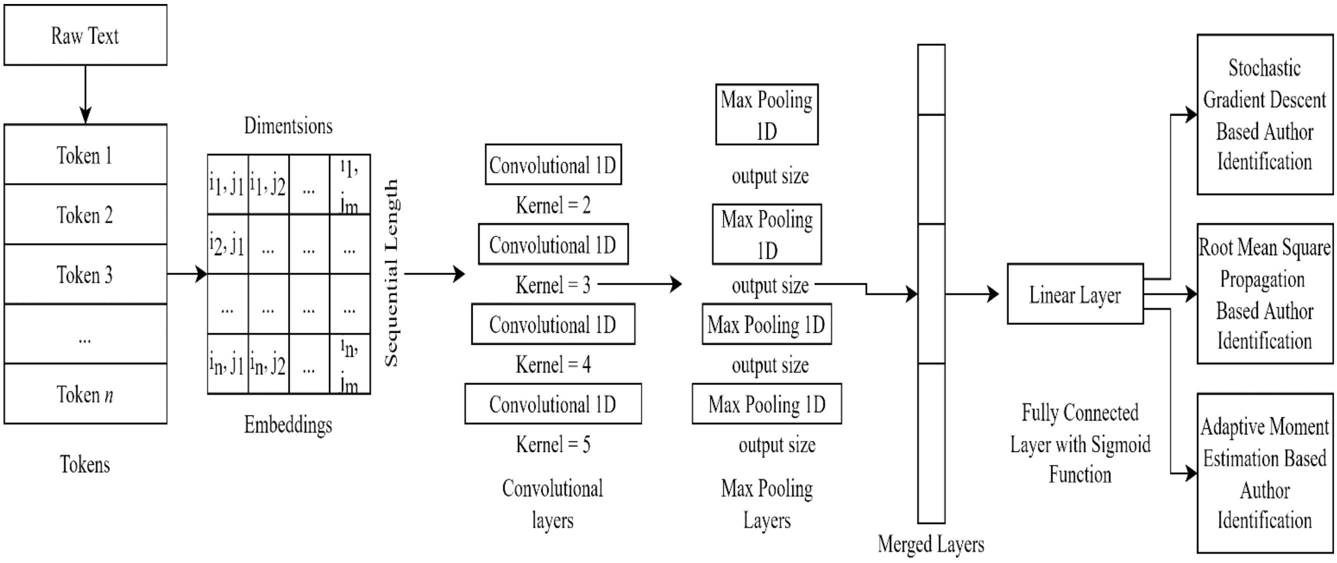

FIGURE 1. Complete convolutional neural network (CNN) model for author verification.

at the fully connected layer with a sigmoid function, as shown in Figure 1, which is based on Discriminator and Generator.

Using a hyper-tuned- Convolutional Neural Network (CNN), we produce a wide range of synthetic texts of varying lengths and quality to study different authors' Stylometric analyses of texts. Thus, we may automatically construct various controlled texts without supervision. To produce author verification labels {0/1} (text related to that author {0} or not {1}) using Convolutional Neural Networks (CNN), some issues must be solved. Because we use the Urdu Corpus for this study, texts are discrete. There is no way for the gradient to flow from the Discriminator to the generator in a differentiable sampling step. This section is divided into two parts. In section (a), we show the proposed hyper-tuned model of Convolutional Neural Networks (CNN) by optimizing the Adaptive Moment (ADAM), Stochastic Gradient Descent (SGD), and Root Mean Square Propagation (RMSProp). While in section (b), we are developing the pairs of generators and discriminators at the fully connected layer of Convolutional Neural Networks (CNN) to produce positive synthetic data, negative synthetic data, and class labels. We have hyper-tuned the discriminator and generator functions using Equation (4). The following are the paper's significant contributions:

(i) For the generation of generic, diverse, and high-quality Stylometric text with various class labels, we present a unique semi-supervised framework called hyper tuned-CNN-based Author Verification Model.
(ii) We propose a new penalty-based multi-objective function for each CNN generator output text with various class labels.
(iii) The results of extensive testing on the Urdu dataset show that our approach is effective and superior.

B. HYPER-TUNED CONVOLUTIONAL NEURAL NETWORK

A convolutional neural network (CNN) is a specialized type of artificial neural network that excels at analyzing and interpreting data, particularly in the domains of computer vision and natural language processing. While 2D CNNs are widely used for image data, 1D CNNs are specifically designed to handle sequential data such as text and 1D signals.

In the context of text classification, 1D CNNs leverage filters of varying sizes and shapes to effectively reduce the dimensionality of input matrices. This is especially beneficial for distributed and discrete word embeddings commonly used in text analysis. When implementing a CNN model for text, convolutions are applied across all channels of the input to capture meaningful dependencies within the text data. Following convolutions, pooling operations and additional convolutional layers are often employed to create a hierarchical structure of feature extractors. The extracted features are then passed through the network, usually as a reshaped vector represented by a single row.

In our work, we aimed to optimize the CNN model by incorporating three different optimizers. However, in order to provide a more comprehensive explanation, additional details about the specific optimizers and their implementations are necessary.

Furthermore, we performed a hyper parameter tuning process to identify the best combination of hyper parameters for our task. The specific hyper parameters that were tuned include the number of filters in the convolutional layers for different features (lexical_filters, syntactic_filters, and structural_filters), the number of units in the dense layer (dense_units), and the learning rate of the optimizer (learning_rate).

To conduct the hyper parameter search, we employed the Random Search class from the Keras Tuner library, which enables a randomized exploration of the hyper parameter space. The tuner utilized the build_model function, which not only defines the model architecture but also incorporates the hyper parameters of interest. Our objective during the search was to maximize the accuracy on the validation dataset. The tuner performed a predetermined number of trials (max_trials), with each trial training the model using a specific set of hyper parameters.

Upon completing the hyper parameter search, we determined the best hyper parameters by utilizing the get_best_hyperparameters method. In our model, we obtained only the best set of hyper parameters. Subsequently, we constructed the final model using these optimal hyper parameters, and it was trained on the training data using the specified batch size and number of epochs.

1) ROOT MEAN SQUARE PROPAGATION (RMSPROP)

This propagation was invented by Geoffrey Hinton; with a moving average squared gradient, Root Mean Square Propagation (RMSProp) tries to resolve the dramatically reduced learning rates for Adagrad. The Root Mean Square Propagation (RMSProp) study rate will be automatically updated for each parameter. Root Mean Square Propagation (RMSProp) divides the average learning rate between squared gradients by their exponential decay. Below, Equation (1) shows the calculation of Root Mean Square Propagation (RMSProp) in the CNN hyper-tuned model:

$$\theta_{t+1} = \theta_t - \frac{n}{\sqrt{(1-y)\, g_{t-1}^2 + y\left[g\,(t-1)^2\right] + \in}} \tag{1}$$

n is the decay term that takes 0 to 1 in value. gt moves an average gradient of squared

2) ADAM – ADAPTIVE MOMENT ESTIMATION

Adaptive Moment (ADAM) is another method that calculates each parameter's adaptive learning rate from the estimates of the first and second instants. Adagrad's significantly lower learning rates are also reduced. ADAM can be seen as an Adagrad combination, which works well in sparse gradients and Root Mean Square Propagation (RMSProp), both online and non-stationary. The ADAM algorithm updates the first and second moment's exponential moveable gradient averages (mt) and squared gradient (vt).

The exponential decay rates of these movements are controlled by the hyper-parameters of $\beta 1$, $\beta 2$ $\beta 1$ [0,1], as shown below in Equation (2).

$$\begin{aligned} m_t + v_t = {} & [\beta_{1+mt-1} + (1-\beta_1)_{g_t}] \\ & + [\beta_{2+mt-1} + (1-\beta_2)_{g_t} 2] \end{aligned} \tag{2}$$

Moving averages are initialized as zero, which leads to instant estimates of zero in the first steps. This initialization partition can be counteracted, and biased estimates may be achieved. The updated parameters of the adaptive moment in the Convolutional Neural Networks (CNN) model are shown in equations (3) and (4):

$$*m_t + *v_t = \left[\frac{m_t}{1-\beta_1^t}\right] + \left[\frac{v_t}{1-\beta_2^t}\right] \tag{3}$$

$$\theta_{t+1} = \theta_t - \frac{n{*}m_t}{\sqrt{*v_t + \in}} \tag{4}$$

3) STOCHASTIC GRADIENT DESCENT (SGD)

Stochastic Gradient Descent (SGD) only calculates on a small subset or random selection of data instances instead of computations on the entire dataset, which is redundant and inefficient. ADAM is essentially an algorithm to optimize stochastic objective functions through gradients.

C. PAIRS OF GENERATORS AND DISCRIMINATORS

Three sets of generators (G_p, G_n, and G_l) and discriminators (D_p, D_n, D_l) make up the proposed Hypertuned- Convolutional Neural Networks (CNN) based model at a fully connected layer with a sigmoid function. They are in charge of generating positive synthetic data (p), negative synthetic data (n), and class labels (l). These three pairs are mathematically defined in Equations (5), (6), and (7):

The diagram of the proposed model may be seen in Figure 2,

$$\begin{aligned} & \min_{Gp} max_{Dp} \mathrm{V}\,(D, G) \\ & \quad = exp_{(x)}^{x\sim p} \log_{10}(\mathrm{Dp}\,(\mathrm{x})) \\ & \qquad + exp_{(z)}^{z\sim p} log_{10}(1 - Dp(Gp(z))) \end{aligned} \tag{5}$$

$$\begin{aligned} & \min_{Gn} max_{Dn} \mathrm{V}\,(D, G) \\ & \quad = exp_{(x)}^{x\sim n} \log_{10}(\mathrm{Dn}\,(\mathrm{x})) \\ & \qquad + exp_{(z)}^{z\sim n} \log_{10}(1 - \mathrm{Dn}\,(\mathrm{Gn}\,(\mathrm{z}))) \end{aligned} \tag{6}$$

$$\begin{aligned} & \min_{Gn,Gp,Gl} max_{Dl} \mathrm{V}\,(D, G) \\ & \quad = exp_{(x)}^{x\sim l} \pi \log_{10}(\mathrm{Dl}\,(\mathrm{x})) \\ & \qquad + exp_{(z)}^{z\sim l} \pi \log_{10}(1 - \mathrm{Dl}\,(\mathrm{Gl}\,(\mathrm{z}))) \end{aligned} \tag{7}$$

This model uses three different types of generators (Gp, Gn, and Gl) and three different discriminators (D_p, D_n, D_l) at fully connected layers of the Convolutional Neural Networks (CNN) model. Generating positive synthetic data is the responsibility of G_p, and D_p makes a distinction between naturally occurring and artificially created positive text synthetic data. For negative data, G_n and D_n both have a comparable role. G_l and D_l take the data created by G_p and G_n as input and produce a class label of author verification(0/1) (text related to that author [0] or not [1]); D_l serves as a discriminator for Generator G_l to determine which class it belongs to. Generator Loss is the evaluation parameter to evaluate the proposed model. The generator makes an effort for this function L (G_l) as efficient as possible. It means that it seeks the maximum output of the discriminator D(x) from its negative instances. When evaluating these functions, the critic's output is denoted by Equation (8):

$$\mathrm{L}\,(G_l) = \pi_p \left[D_l(G_l \log_{10} G_z x^p\right)] + \pi_n \left[D_l(G_l \log_{10} G_z x^n\right)] \tag{8}$$

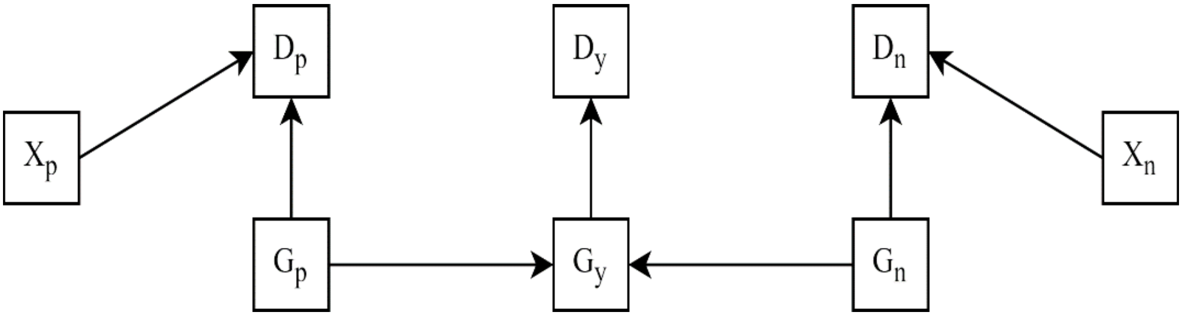

FIGURE 2. Diagram of proposed pairs of discriminator and generator in the proposed model.

In Equation (8), L (G_l), the Loss of the Generator π_p is the probability of data having a positive instance while π_n Is the probability of data having a negative instance. Attaining the equilibrium condition is critical for a neural network system. First, we must determine the optimized and hyper-tuned discriminator settings (see section (a) to arrive at an equilibrium condition. To find the generator's minimization conditions, use the Discriminator's optimum conditions as input by using the hyper parameters of Adaptive Moment (ADAM), Stochastic Gradient Descent (SGD), and Root Mean Square Propagation (RMSProp). Considering that the generators (G_p, G_n, and G_y) are fixed, and p and n are the probability of positive and negative claims in the dataset. As a result, when the system is in equilibrium, the distribution of positive and negative generated data, and both will follow Equations (9) and (10), respectively.

$$p\left[G\left(p_x\right)\right]=p\left[P\left(x\right)\right] \tag{9}$$

$$p\left[G\left(n_x\right)\right]=p\left[N\left(x\right)\right] \tag{10}$$

After hyper tuning the parameters of Equations (5), (6), and (7) using Equations (1) and (2), the optimal Discriminator D^H functions are (11)-(13), shown at the bottom of the page.

Minimum value of Generators can be obtained when the following Equations (14)-(16) are satisfied:

$$p\left[G\left(p_x\right)\right]=pG\left[P\left(x\right)\right]*m_t+*v_t \tag{14}$$

$$p\left[G\left(n_x\right)\right]=pG\left[N\left(x\right)\right]*m_t+*v_t \tag{15}$$

$$p\left[G\left(l_x\right)\right]=\pi pG\left[P\left(x\right)\right]+\pi nG\left[N\left(x\right)\right]*m_t+*v_t \tag{16}$$

Generator learning and discriminator learning are two opposing learning objectives that can be applied to the framework. Generating texts with the ith Stylometric type may fool a discriminator, which is why the ith generator, G_i, is used. It seeks to minimize our proposed penalty-based objective. Instead, our multi-class classification objective is to identify as feasible between phoney texts (texts made by generators) and authentic texts with k different attitude types. With no loss of generality, we allowed a hyper-tuned- Convolutional Neural Network (CNN) to generate two writing styles by setting k to 2. (Positive and negative for class label 0 or 1).

By forcing each generator to develop sentimental texts that are distinct from texts generated by others, our multi-class classification aim helps to increase the sentiment accuracy of the generated texts. To begin with, the best ith generator can figure out the style distribution of authentic texts from authors. The Discriminator's main purpose is to identify differences between the groups of authors. This study makes use of the Urdu-based dataset. The dataset contains the author ID, Name, and Text:

D. SIMILARITY OF TEXT

It is a technique to find the similarity and dissimilarities among the text [16], [32]. In this method, a matrix is formed. The matrix is used to identify the level of similarity; the product with less distance on a matrix to another product has more similarity, while the products with more distance have less similarity. Similarity is an important property of the products. It is used to classify the products. [33] The classification is then further used to get the right recommendation. Cosine Similarity is the measurement of cosine angles between two vectors. It is the value concerning the origin of the angle. The cos (θ) of degree 1 is 0. The cos1 means products are similar. When the value of cos 90^0, the result is 0,

$$D^H\mathrm{p}\left(\mathrm{x}\right)=\frac{p\left[G\left(p_x\right)\right]}{p\left[G\left(p_x\right)\right]+p\left[P\left(x\right)\right]}(\theta_t-\frac{n*m_t}{\sqrt{*v_t+\in}}) \tag{11}$$

$$D^H\mathrm{n}\left(\mathrm{x}\right)=\frac{p\left[G\left(n_x\right)\right]}{p\left[G\left(n_x\right)\right]+p\left[N\left(x\right)\right]}(\theta_t-\frac{n*m_t}{\sqrt{*v_t+\in}}) \tag{12}$$

$$D^H\mathrm{l}\left(\mathrm{x}\right)=\frac{p\left[G\left(l_x\right)\right]}{p\left[G\left(l_x\right)\right]+(\pi_p\left[D_l(G_l\log G_z x^p)\right]+\pi_n\left[D_l(G_l\log G_z x^n)\right](\theta_t-\frac{n*m_t}{\sqrt{*v_t+\in}}))} \tag{13}$$

which means the products are not similar. The following Equation defines it.

$$\vec{q}.\vec{d} = \left\| \xrightarrow[q]{} \right\| . \left\| \xrightarrow[d]{} \right\| . \cos\theta \tag{17}$$

$$Sim\,(q.d) = \cos\theta = \frac{\vec{q}.\vec{d}}{\left\| \xrightarrow[q]{} \right\| . \left\| \xrightarrow[d]{} \right\|} \tag{18}$$

For better performance accuracy formula for the novel proposed model evaluation.

1) ACCURACY

Classification Accuracy is what we usually mean when we use the term accuracy. It is the number of correct predictions to the total number of input samples, (19), and (20), shown at the bottom of the page.

V. RESULTS AND DISCUSSION

In this section, we present the results and discuss the findings of our authorship verification study conducted on Urdu text using a Convolutional Neural Network (CNN) model. The study aimed to investigate the impact of different embedding techniques, namely Word2Vec, GloVe, and FastText, on the performance of the CNN models. Additionally, we compared the results obtained with three optimization algorithms, namely ADAM, SGD, and RMSProp. Furthermore, to provide a comprehensive analysis, we also conducted experiments using traditional machine learning models, namely Support Vector Machines (SVM) and Random Forest, for authorship verification. The following subsections provide a detailed analysis of the results, followed by a discussion of the implications and insights gained from the study.

A. CNN-ADAM WITH Word2Vec

With ADAM optimization and the Word2Vec embedding technique, the CNN model was able to verify the authorship of Urdu text with a 93% accuracy rate. Word2Vec's reliance on co-occurrence patterns for word representation may limit its capacity to capture the subtleties unique to Urdu, albeit having slightly lower accuracy than other algorithms.

B. CNN-ADAM WITH GloVe

The GloVe embedding method with CNN-ADAM produced an accuracy of 96% for identifying the author of Urdu text. With the use of global word co-occurrence statistics, GloVe is better able to understand semantic and syntactic features, which improves its ability to spot authorship patterns.

C. CNN-ADAM WITH FastText

When used in conjunction with the FastText embedding method, the CNN-ADAM model verified Urdu text's authorship with an astounding 98% accuracy. FastText excels at reliably recognizing authorship patterns thanks to the addition of subword information that enables it to grasp the morphological and syntactic intricacies unique to Urdu.

The comparative findings of the CNN-ADAM models that made use of the Word2Vec, GloVe, and FastText embedding strategies are displayed in figure 3, which can be seen here.

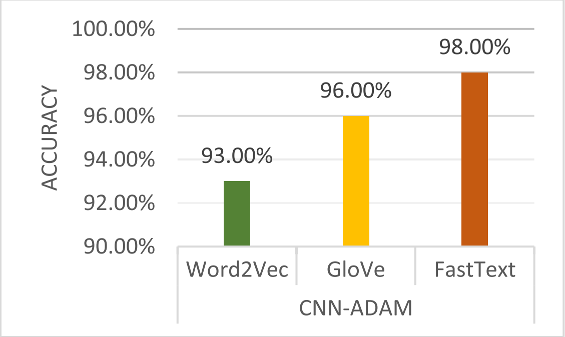

FIGURE 3. Comparative results of CNN-ADAM with Word2Vec, GloVe, and FastText.

D. CNN-SGD WITH Word2Vec

In terms of authorship verification for Urdu text, the CNN model with SGD optimization and Word2Vec embedding approach achieved an accuracy of 94%. SGD offers an effective alternative to CNN-ADAM with Word2Vec for authorship verification tasks due to its use of stochastic gradient descent and capacity for handling big datasets.

E. CNN-SGD WITH GloVe

The GloVe embedding method with CNN-SGD produced an accuracy of 96% for identifying the author of Urdu text. GloVe uses global word co-occurrence statistics to provide a semantic and syntactic knowledge of Urdu text, which enhances its performance in properly identifying authorship patterns.

F. CNN-SGD WITH FastText

An accuracy of 96% was attained when evaluating the authorship of Urdu text using the CNN-SGD model and the FastText embedding method. Its performance in identifying authorship patterns is improved by FastText's addition of subword information, which helps it to capture the morphological and syntactic nuances unique to Urdu.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total number of a prediction made}} \tag{19}$$

$$\text{Accuracy} = \frac{\text{TruePositives} + \text{TrueNegatives}}{\text{TruePositives} + \text{TrueNegatives} + \text{FalsePositives} + \text{FalseNegatives}} \tag{20}$$

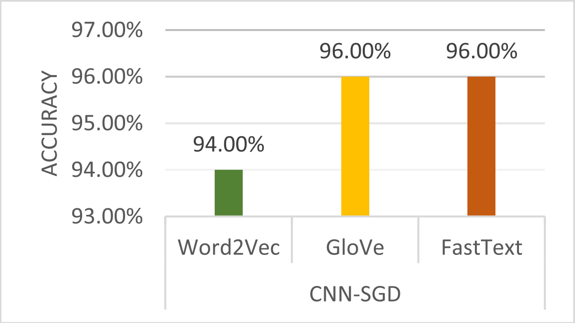

FIGURE 4. Comparative results of CNN-SGD with Word2Vec, GloVe, and FastText.

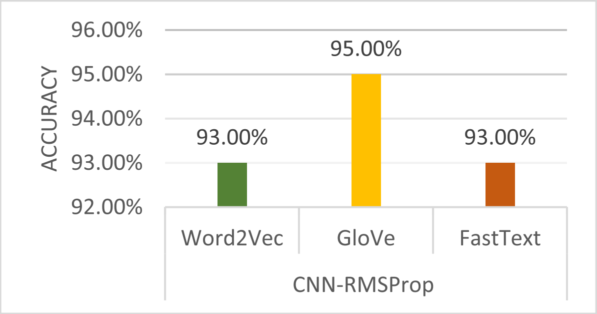

FIGURE 5. Comparative results of CNN-RMSProp with Word2Vec, GloVe, and FastText.

Figure 4 shows the comparison of the outcomes of the CNN-SGD models using the Word2Vec, GloVe, and FastText embedding methods.

G. CNN-RMSProp WITH Word2Vec

An accuracy of 93% was attained in the authorship verification of Urdu text using the CNN model with RMSProp optimization and Word2Vec embedding approach. Word2Vec's reliance on co-occurrence patterns, which is similar to the results achieved with CNN-ADAM, may limit its capacity to capture the distinctive properties of Urdu, yielding slightly poorer accuracy in comparison to other systems.

H. CNN-RMSProp WITH GloVe

The accuracy of authorship verification of Urdu text increased to 95% when the GloVe embedding technique was combined with CNN-RMSProp. GloVe's application of global word co-occurrence data enables it to provide a semantic and syntactic comprehension of Urdu text, which helps to the success of the system in identifying authorship patterns.

I. CNN-RMSProp WITH FastText

When paired with the FastText embedding technique, the CNN-RMSProp model was able to reach an accuracy of 93% in the verification of the authorship of Urdu text. Although the accuracy gained with CNN-ADAM and CNN-SGD with FastText was slightly lower than it was with FastText, the integration of subword information into FastText allows it to capture the morphological and syntactic nuances that are unique to Urdu.

The comparative outcomes of the CNN-RMSProp models using the Word2Vec, GloVe, and FastText embedding strategies are shown in figure 5.

J. SUPPORT VECTOR MACHINES

In addition to the CNN models, we also conducted experiments using Support Vector Machines (SVM) for authorship verification in Urdu. The SVM model achieved an accuracy of 94%, further demonstrating the effectiveness of the proposed approach.

K. RANDOM FOREST

We also explored the use of Random Forest algorithm for Urdu text analysis. The Random Forest model yielded an accuracy of 92%, further validating the efficacy of our proposed approach.

Authorship patterns in Urdu text can be identified using CNN models trained with a variety of embedding techniques and optimization algorithms. A comparative comparison of these results has shed light on the effectiveness of these models in this regard. Figure 6 shows the comparative outcomes of all the models, demonstrating their individual accuracy levels and emphasizing the variations in performance.

The CNN-ADAM optimization algorithm outperformed the other two optimization algorithms consistently across all embedding methods. With Word2Vec, GloVe, and FastText, it obtained accuracy levels of 93%, 96%, and 98%, respectively. With accuracies of 94%, 96%, and 96% for Word2Vec, GloVe, and FastText, respectively, CNN-SGD demonstrated competitive results. In contrast, CNN-RMSProp produced significantly lower accuracy results for the corresponding embedding strategies, 93%, 95%, and 93%. Additionally, we conducted experiments with Support Vector Machines (SVM) and Random Forest. The SVM model achieved an accuracy of 94%, while the Random Forest model achieved an accuracy of 92%. These results further validate the superior performance of the CNN-ADAM model with FastText embeddings for authorship verification in Urdu.

With a 98% accuracy rate, CNN-ADAM with FastText emerged as the most successful method for authorship verification in Urdu text when taking the entire dataset into account. FastText's higher performance is a result of the integration of subword information, which helps it to recognize the subtle syntactic and morphological variations unique to Urdu. With an accuracy rate of 96%, GloVe's semantic and syntactic comprehension also produced remarkable results. However, Word2Vec showed somewhat poorer accuracy than all other optimization algorithms when it only considered co-occurrence patterns.

The significance of choosing the right embedding method and optimization algorithm for authorship verification tasks in Urdu literature is highlighted by these findings. FastText

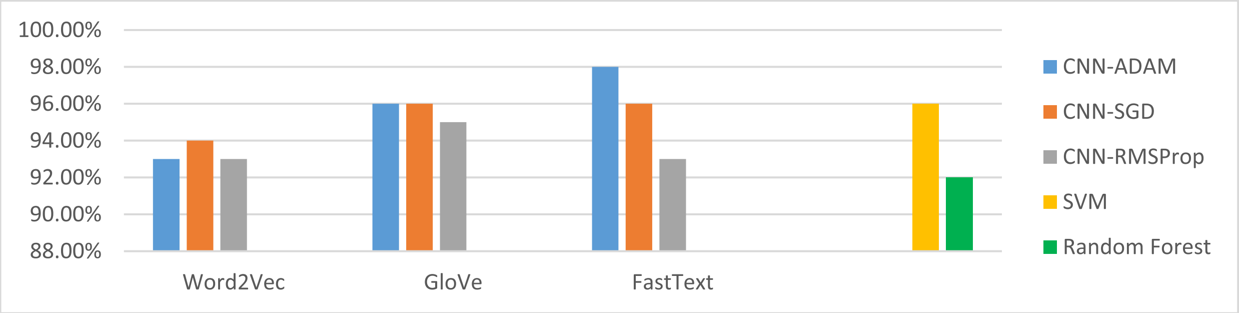

FIGURE 6. Comparative results of CNN-ADAM, CNN-SGD, and CNN-RMSProp with Word2Vec, GloVe, and FastText, along with support vector machines (SVM) and random forest.

TABLE 3. Comparative statistical measures of all models.

Model	Precision %	Recall %	F1 Score %	Accuracy %
CNN-ADAM with Word2Vec	94	92	93.00	93.41
CNN-ADAM with Glove	94	98	96.00	96.36
CNN-ADAM with FastText	98	97	98.00	98.32
CNN-SGD with Word2Vec	93	97	95.00	94.22
CNN-SGD with Glove	95	98	96.00	96.34
CNN-SGD with FastText	97	97	97.00	96.43
CNN- RMSProp with Word2Vec	93	89	90.00	93.12
CNN- RMSProp with Glove	90	91	91.00	95.36
CNN- RMSProp with FastText	90	95	93.00	93.39
SVM	94	97	94.00	94.21
Random Forest	90	91	92.00	92.19

can considerably increase the accuracy of such systems when combined with CNN-ADAM, making it possible to identify authorship patterns with confidence. These findings have important ramifications for practical applications such as forensic linguistics, plagiarism detection, and security software where establishing authorship is essential.

In addition to the findings and implications discussed above, it is important to acknowledge and address the limitations and challenges associated with the proposed CNN-based approach for Urdu authorship verification. The limited availability of Urdu language resources, including labeled datasets and pre-trained word embeddings, poses a significant challenge compared to widely studied languages like English. The scarcity of resources can hinder the development and training of robust language models, potentially impacting the accuracy and generalization of the proposed approach. Furthermore, the complex morphology and syntax of Urdu, along with the variation in writing styles among Urdu authors, present inherent difficulties in accurately capturing the nuances of authorship solely based on textual features.

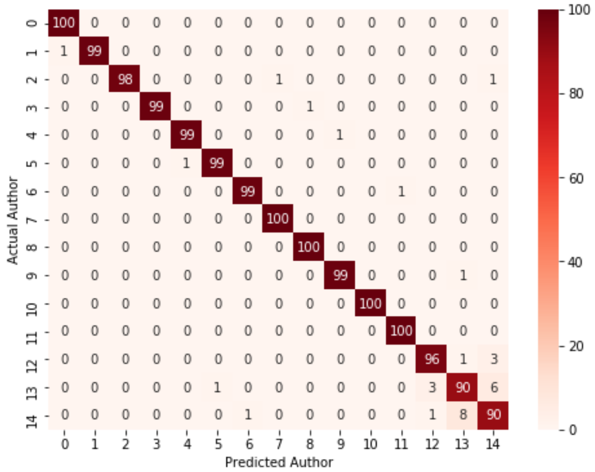

FIGURE 7. Confusion matrix for CNN-ADAM with FastText.

Exploring additional techniques such as recurrent neural networks or attention mechanisms could help address these limitations and improve the model's performance in capturing the unique writing styles and idiosyncrasies specific to individual authors.

Additionally, the size and diversity of the dataset used for training and evaluation play a crucial role in the model's generalization capabilities. Ensuring a representative and sufficiently large dataset that encompasses diverse writing styles and authors is essential to mitigate the challenges associated with data scarcity and bias. Moreover, the cultural and contextual influences in Urdu language and literature may not be fully captured by the proposed CNN-based approach alone. Incorporating additional features or linguistic knowledge specific to Urdu literature and culture could enhance the model's performance and enable a more comprehensive analysis of authorship. Future research should focus on developing comprehensive datasets, designing architectures that better capture the unique linguistic characteristics of Urdu, and incorporating domain-specific knowledge to address these limitations. By addressing these challenges, the CNN-based approach for Urdu authorship verification can be improved and provide valuable insights for text analysis and attribution tasks in Urdu literature.

In addition to Urdu authorship verification, the proposed CNN-based methodology shows potential for application in other low-resource languages or domains. To achieve this, factors such as the availability of labelled datasets and pre-trained word embeddings specific to the target language or domain should be considered. Linguistic characteristics unique to the language or domain may require adaptations to the architecture or inclusion of additional linguistic features. The feasibility of adapting the methodology to different datasets and leveraging domain-specific knowledge should also be assessed. While our study focused on Urdu, future research can explore the transferability of the CNN-based approach to diverse linguistic contexts, conducting comparative studies to gain insights into its effectiveness.

VI. CONCLUSION

In this study, our objective was to conduct authorship verification of Urdu text using CNN models with various embedding techniques and optimization algorithms. The comparative analysis of the results revealed the effectiveness of CNN-ADAM across all embedding techniques, with FastText achieving the highest accuracy of 98%. This finding underscores the importance of selecting appropriate techniques for authorship verification in Urdu text and highlights the value of incorporating subword information through FastText. Our study contributes valuable insights to the field of authorship verification in Urdu text, with practical implications for forensic linguistics, plagiarism detection, and security applications.

Moving forward, further research can be conducted to explore additional factors that may enhance the performance and robustness of authorship verification systems in Urdu text. This may include investigating the impact of different feature representations, exploring the use of ensemble methods, or considering the influence of document length on verification accuracy. By addressing these avenues, we can continue to advance the field of authorship verification and its applications in Urdu text analysis.

In conclusion, our study demonstrates the efficacy of CNN models with FastText embedding in authorship verification of Urdu text. The findings contribute to our understanding of authorship verification in low-resource languages and provide practical insights for real-world applications. With ongoing research and development, we can further improve the accuracy and reliability of authorship verification systems in Urdu and other similar languages.

REFERENCES

[1] H. van Halteren, "Linguistic profiling for author recognition and verification," Assoc. Comput. Linguistics, East Stroudsburg, PA, USA, Tech. Rep., 2004, p. 199. [Online]. Available: https://hdl.handle.net/2066/61127, doi: 10.3115/1218955.1218981.

[2] L. Tawalbeh, F. Muheidat, M. Tawalbeh, and M. Quwaider, "IoT privacy and security: Challenges and solutions," *Appl. Sci.*, vol. 10, no. 12, pp. 1–17, 2020, doi: 10.3390/APP10124102.

[3] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC and its potential for DDoS attacks: A comprehensive measurement study," in *Proc. Conf. Internet Meas. Conf.*, Nov. 2014, pp. 449–460, doi: 10.1145/2663716.2663731.

[4] M. Carlos-Mancilla, E. López-Mellado, and M. Siller, "Wireless sensor networks formation: Approaches and techniques," *J. Sensors*, vol. 2016, pp. 1–18, Mar. 2016, doi: 10.1155/2016/2081902.

[5] 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, doi: 10.1007/BF00054024.

[6] S. Ruder, P. Ghaffari, and J. G. Breslin, "Character-level and multi-channel convolutional neural networks for large-scale authorship attribution," 2016, *arXiv:1609.06686*.

[7] 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, doi: 10.1109/TIFS.2016.2603960.

[8] 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, no. 11, Nov. 2017, Art. no. e0187204, doi: 10.1371/journal.pone.0187204.

[9] B. Alsulami, E. Dauber, R. Harang, S. Mancoridis, and R. Greenstadt, "Source code authorship attribution using long short-term memory based networks," in *Proc. Eur. Symp. Res. Comput. Secur.*, vol. 10492, 2017, pp. 65–82, doi: 10.1007/978-3-319-66402-6_6.

[10] E. Manjavacas, J. De Gussem, W. Daelemans, and M. Kestemont, "Assessing the stylistic properties of neurally generated text in authorship attribution," in *Proc. Workshop Stylistic Variation*, 2017, pp. 116–125, doi: 10.18653/v1/w17-4914.

[11] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *Lang. Resour. Eval.*, vol. 45, no. 1, pp. 83–94, Mar. 2011, doi: 10.1007/s10579-009-9111-2.

[12] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014, doi: 10.1109/JSTARS.2014.2329330.

[13] 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, 2020, doi: 10.1109/ACCESS.2020.3034932.

[14] R. Guarasci, S. Silvestri, G. D. Pietro, H. Fujita, and M. Esposito, "BERT syntactic transfer: A computational experiment on Italian, French and English languages," *Comput. Speech Lang.*, vol. 71, pp. 1–19, Apr. 2021, doi: 10.1016/j.csl.2021.101261.

[15] 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, 2020, doi: 10.1109/ACCESS.2020.2967449.

[16] 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, 2019, doi: 10.1109/ACCESS.2018.2885011.

[17] 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, 2019, doi: 10.1109/ACCESS.2019.2930536.

[18] 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, 2019, doi: 10.1109/ACCESS.2019.2943639.

[19] 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, 2020, doi: 10.1109/ACCESS.2020.2964952.

[20] 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, 2021, doi: 10.1109/ACCESS.2020.3047583.

[21] T. Chakraborty, "Authorship identification in Bengali literature: A comparative analysis," 2012, *arXiv:1208.6268*.

[22] 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, doi: 10.5120/13566-1375.

[23] M. T. Hossain, M. M. Rahman, S. Ismail, and M. S. Islam, "A stylometric analysis on Bengali literature for authorship attribution," in *Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2017, pp. 1–5, doi: 10.1109/ICCITECHN.2017.8281768.

[24] D. M. Anisuzzaman and A. Salam, "Authorship attribution for Bengali language using the fusion of N-Gram and Naive Bayes algorithms," *Int. J. Inf. Technol. Comput. Sci.*, vol. 10, no. 10, pp. 11–21, Oct. 2018, doi: 10.5815/ijitcs.2018.10.02.

[25] U. Pal, A. S. Nipu, and S. Ismail, "A machine learning approach for stylometric analysis of Bangla literature," in *Proc. 20th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2017, pp. 1–5, doi: 10.1109/ICCITECHN.2017.8281800.

[26] D. S. Sharma, "Automated analysis of Bangla poetry for classification and poet identification," in *Proc. 12th Int. Conf. Natural Lang. Process.* Perth, WA, Australia, Dec. 2015, pp. 247–253. [Online]. Available: https://aclanthology.org/W15-5937

[27] A. Khatun, A. Rahman, M. S. Islam, and Marium-E-Jannat, "Authorship attribution in Bangla literature using character-level CNN," in *Proc. 22nd Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2019, pp. 18–20, doi: 10.1109/ICCIT48885.2019.9038560.

[28] D. S. Sharma, R. Sangal, S. Proc, S. Phani, S. Lahiri, and A. Biswas, "Authorship attribution in Bengali language," in *Proc. 12th Int. Conf. Natural Lang. Process.*, 2012, pp. 100–105. [Online]. Available: http://www.isical.ac.in/

[29] Md. R. Hossain, M. M. Hoque, M. A. A. Dewan, N. Siddique, M. N. Islam, and I. H. Sarker, "Authorship classification in a resource constraint language using convolutional neural networks," *IEEE Access*, vol. 9, pp. 100319–100338, 2021, doi: 10.1109/ACCESS.2021.3095967.

[30] O. Adams, A. Makarucha, G. Neubig, S. Bird, and T. Cohn, "Cross-lingual word embeddings for low-resource language modeling," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, Long Papers*, vol. 1, 2017, pp. 937–947, doi: 10.18653/v1/e17-1088.

[31] R. Sarwar, Q. Li, T. Rakthanmanon, and S. Nutanong, "A scalable framework for cross-lingual authorship identification," *Inf. Sci.*, vol. 465, pp. 323–339, Oct. 2018, doi: 10.1016/j.ins.2018.07.009.

[32] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004, doi: 10.1145/963770.963772.

[33] A. Agarwal and M. Chauhan, "Similarity measures used in recommender systems: A study," *Int. J. Eng. Technol. Sci. Res.*, vol. 4, no. 6, pp. 619–626, 2017. [Online]. Available: www.ijetsr.com

TALHA FAROOQ KHAN is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science (DCS), The Islamia University of Bahawalpur, Punjab, Pakistan. His current research interests include text mining, web mining, machine learning, and deep learning.

WAHEED ANWAR received the Ph.D. degree in computer science from The Islamia University of Bahawalpur, Pakistan, in 2019. He is currently an Assistant Professor with the Department of Computer Science, The Islamia University of Bahawalpur. In addition, he is also a sun certified java programmer SCJP2. He has published 15 articles in well-reputed peer-refereed journals. His accumulative impact factor is 33C. He has over 15 years of teaching and research and development experience. His current research interests include text mining, web mining, machine learning, and deep learning.

HUMERA ARSHAD received the master's degree in information technology from the National University of Science and Technology (NUST), Pakistan, and the Ph.D. degree from the School of Computer Science, University Sains Malaysia. She joined the Faculty of Computer Sciences and IT, in 2004. She is currently an Associate Professor and the Chairperson of the Department of Computer Sciences, The Islamia University of Bahawalpur, Pakistan. Her research interests include digital and social media forensics, information security, online social networks, cybersecurity, intrusion detection, reverse engineering, and semantic web.

SYED NASEEM ABBAS is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science and Information Technology (DCS), The Islamia University of Bahawalpur, Punjab, Pakistan. His current research interests include text mining, web mining, machine learning, and deep learning.