C5050: An Efficient Framework for Author Identification Using Deep Learning

Ahmed Anwar
Department of Computer science,
Faculty of Computers
and Information, Menofia
University, Shebin El Kom, Egypt
ahmed.elasfr@gmail.com

Arabi E.Keshk
Department of Computer science,
Faculty of Computers
and Information, Menofia
University, Shebin El Kom, Egypt
arabi.keshk@ci.menofia.edu.eg

Eman M.Mohamed
Department of Computer science,
Faculty of Computers
and Information, Menofia
University, Shebin El Kom, Egypt
eman.mohamed@ci.menofia.edu.eg

***Abstract*— Author identification aims to uncover the individualsresponsible for creating texts, and it is a burgeoning field ofresearch with diverse applications in literary analysis, cybersecurity, forensics, and social media investigations. The primary goal of this paper is to perform an analysis on author identification. We introduce two main elements within this study. The initial element utilizes six machine learning (ML) techniques: Decision Trees (DT), Logistic Regression (LR), k Nearest Neighbors (K-NN), Random Forests (RF), Support Vector Machines (SVM), and Naive Bayes (NB), with the application of the TF-IDF method for feature extraction. The second part involves the experimentation with two variations of Deep Learning (DL) models—specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)—employing word embedding for the input vector. To validate our approach, we conducted an experimental study using the Reuters 50_50 dataset,employing two learning modes: Hold-out and 10-fold cross validation. The obtained results, measured in terms of Accuracy (ACC), Precision (PREC), Recall (REC), and F1-score (F1), demonstrate the superior performance of DL techniques when employing a 10-fold cross-validation strategy compared to the current state-of-the-art methods. The experiments detailed in this paper showcase the efficacy of our proposed DL models, yielding the best results for author identification.**

Keywords: Author Identification; Machine Learning; Deep Learning; Classification.

I. INTRODUCTION

In today's digital era, the exponential growth of textual data has created a need for efficient methods to identify authors of various written works. Author identification plays a crucial role in areas such as forensic linguistics, plagiarism detection, and content analysis. Traditional approaches often rely on manual analysis, which is time consuming and subject to human biases. As a result, researchers have turned to automated methods, particularly those leveraging machine learning algorithms, to tackle this challenging task.

Author identification is the process of determining the authorship of a particular document or piece of text. It is an important task in various fields, such as law enforcement, journalism, and literary studies. In recent years, machine learning algorithms have been increasingly used for author identification due to their ability to handle large amounts of data and identify patterns.

However, the accuracy of machine learning algorithms for author identification depends on various factors, such as the size of the training data, the feature selection, and the choice of algorithm. In this context, an efficient framework for authoridentification using an optimized machine learning algorithm can significantly improve the accuracy and efficiency of the identification process.

Such a framework typically involves several steps, including data pre-processing, feature extraction, feature selection, and classification. The data pre-processing step involves cleaning and normalizing the data to ensure consistency and accuracy. Feature extraction involves identifying the relevant featuresor characteristics of the text that can be used to distinguishone author from another. Feature selection involves selecting the most relevant features based on their importance and correlation with the target variable. Finally, the classification step involves using a machine learning algorithm to classify the text based on the selected features.

An optimized machine learning algorithm can significantly improve the accuracy and efficiency of author identification by selecting the best algorithm for the given dataset and problem. This can be achieved by evaluating different algorithms and selecting the one that performs the best based on variousmetrics, such as accuracy, precision, and recall.

This paper presents an efficient framework for author identification using an optimized machine learning algorithm. Our framework aims to accurately identify the author of a given text based on their unique writing style and linguistic patterns. By combining advanced natural language processing techniques with state-of-the-art machine learning algorithms, we achieve improved accuracy and scalability in the author identification process.

In order to optimize the performance of our framework, we employ advanced techniques such as hyper parameter tuning, feature selection, and ensemble learning. These methods enable us to fine-tune the model and improve its generalization capabilities, ultimately enhancing the accuracy and robustnessof author identification.

To evaluate the effectiveness of our framework, we conductextensive experiments on a diverse set of textual data from various genres, languages, and authors. We compare our approachagainst existing methods and demonstrate its superiority in terms of accuracy and efficiency. In conclusion, our proposed framework offers an efficient and optimized solution for author identification using machine

learning algorithms. By combining advanced natural language processing techniques, careful algorithm selection, and rigorous optimization, we achieve accurate and scalable author identification. This research contributes to the field of text analysis, providing valuable insights and practical applicationsfor areas such as forensic linguistics, plagiarism detection, andcontent analysis.

The main contributions of this paper are summarized as follows:

- Designing a new framework for author identification using different machine learning and Deep Learning algorithms,
- Introducing the wrapper feature selection using machine and Deep learning algorithms for author identification,

Testing the performance of optimized different machine learning and Deep Learning algorithms over author identification benchmark dataset (C5050).

The remainder of this paper is divided into the following sections: The second section contains a brief reference to a related work. Section.3 summarizes the framework for the proposed method for Author Identification problem. Section 4 discusses the data used, some metrics, and the conclusions drawn from the findings. We conclude our paper in Section 6 with a few observations and the raising of new issues.

II. RELATED WORK

The field of author identification is not unexplored; a lot of researchers investigated the field and came up with satisfactory results. We will mention some of the papers we read and analyzed to assist us in our research

A. On Author Attribution

For a while now, investigations on author attribution have been ongoing. In their early efforts, Mosteller and Wallace distributed 30 function words, including papers on Federalist Papers and conjunctions, prepositions, and references to the original writers [1]. Using novels by six English authors and their Spanish translations, Bogdanova and Lazaridou experimented with cross-language authorship attribution. Eventually, they suggested that machine translation may be utilized as a starting point for this endeavor [2] . Zhao et al. produced outstanding results using Kullback-Leibler [3] split with Dirichletaveraging on AP, Gutenberg, and Reuters-21578 corpora, in contrast to Nasir et al [4]. Semi-supervised anomalyidentification method. Character and term sequence kernelsfor authorship identification of short texts were examined by Sanderson and Guenter [5] To compare the performances, andtwo Markov chain techniques were used. Bozkurt I. N. et al.

[6] used cytometry and features like Vocabulary Diversity, Bag of Words, and Frequency of Word forms (article, pronoun, conjunction) to identify the writing features of five Milliyet columnists. Jonathan H. Clark [7] attempted authorship identification using synonym-based features for their experiment in 2007. Compared to the advances made in authorship attribution studies for English and German works, such study for Bangla has still not reached a high standard. There are just three major research works in Bangla. Das and Mitra examined authorship identification and worked with a data setof 36 documents and three authors. In addition to uni-gramand bi-gram characteristics, a probabilistic classifier method was employed. The uni-gram produced 90% accuracy, whereas the bi-gram produced an astounding 100% accuracy. However, their data set was limited, and the writers had vastly varied writing styles, making classification easier [8]. Chakraborty conducted a ten-fold cross validation on three groups, showing that SVM classifiers can achieve a maximum accuracy of 84%[9].

In addition to these three publications, Shanta Phani sought to attribute three authors using machine learning methods, much like Suprabhat Dass. Jana investigated the influence of Sister Nivedita on Jagadish Chandra Bose's writings, but no classification was performed. P. Das, R. Tasmim, and S. Ismail researched four contemporary Bangladeshi authors utilizing characteristics such as word frequency, word and sentence length, type-token ratio, number of prepositions and pronouns, etc [10] . Hossain and Rahman created a voting system with numerous features categorized by Cosine similarity, attaining an accuracy of 90.67%. Pal, Siddika, and Ismail achieved an accuracy of 90.74% using an SVM classifier on a single featurebased on the work of six authors. None of these publications satisfied the 90% accuracy threshold [11] . Except for Phani, Lahiri, and Biswas's [12] work using multilayered perceptrons,we did not discover much work with neural networks and none with LSTM or convolutional neural networks, or word embedding.

B. On Word Embeddings

Tripodi et al. examined the performance of the CBOW and Skip-gram techniques for the Italian language by adjusting the hyperparameter values [13]. Vine et al. [14] explored the use of unsupervised characteristics produced from word embedding methodologies and discovered that the usage of word embedding enhances the performance of concept extraction methods. In 2017, Haixia Liu [15] performed citation semantic analysis using Word2Vec and discovered that word embeddings effectively distinguish positive and negative citations. Santos et al [16]. Using a convolutional neural network with word embedding as its feature led to the conclusion that both FastText and Word2Vec outperform baseline models such as the SVM Algorithm, Random Forest, Logical Regression, etc. Rudkowsky et al. discovered that word embedding have all the potential to outperform current bag-of-words methodologies in the social sciences field of sentiment research[17]. Joulin et al [18]. discovered that the FastText classifier's accuracy is comparable to that of deep learning classifiers, while its training and assessment times are faster [17]. Also, other related work in [19, 20, 21, 22, 23, 22, 24, 25,26, 27, 28, 29, 30, 31, 27, 32, 33, 31, 34, 35, 36, 37, 38,39, 40] has been proposed in recent years to address machine learning and its application in different fields

III. PROPOSED METHODOLOGY

"Fig. 1" illustrates the proposed framework for author identification from Reuter's news as following steps:

- Acquiring Data
- Data Preprocessing
- Techniques for Feature Extraction
- Data Splitting
- Tuning Parameters for both Machine Learning (ML) and Deep Learning (DL)
- Classification based on ML and the proposed DL
- Outcome Prediction and Evaluation Metrics.

A. Acquiring Data

There are several datasets available for author identification and attribution tasks. Researchers and practitioners often curate their own datasets specific to their research goals or utilize domain specific datasets for authorship analysis in specialized fields, such as legal documents or social media posts. In this paper we used the most common author identification dataset namely Reuters 50_50

The Reuters 50_50 dataset is a widely used benchmark dataset in the field of author identification and attribution. It was created by the Reuters news agency and consists ofa collection of news articles written by 50 different authors. Each author contributed 50 articles to the dataset, resulting ina total of 2,500 documents.

The purpose of the Reuters 50_50 dataset is to provide researchers and practitioners with a standardized dataset for evaluating and developing author identification and attribution algorithms. The dataset covers a wide range of topics, including politics, sports, business, and entertainment, ensuring a diverse representation of writing styles and content.

The articles in the dataset are typically of moderate length, ranging from a few paragraphs to a few pages. They are written in English and follow the journalistic style commonly foundin news articles. The dataset includes both factual reporting and opinion pieces, reflecting the varied nature of journalistic writing.

Each document in the Reuters_50_50 dataset is labeled with the corresponding author's identity, allowing researchers to train and test their algorithms on a supervised learning task. This enables the development of models that can accurately identify the author of a given text based on stylistic and linguistic features.

Researchers can extract various features from the dataset, including word frequencies, syntactic patterns, and other linguistic characteristics, to build models for author identification. The dataset has been extensively used in the development and evaluation of machine learning algorithms, such as supervised classifiers, deep learning models, and natural language processing techniques. By providing a standardized and well labeled dataset, the Reuters_50_50 dataset serves as a valuable resource for researchers and practitioners working on author identification and attribution tasks. It facilitates the development of robust algorithms and helps advance the understanding of writing style analysis, authorship attribution, and related areas of research.

B. Data preprocessing

Data preprocessing is an essential step in preparing the Reuters_50_50 dataset for author identification tasks. It involves transforming and cleaning the raw data to ensure its suitability for analysis and model training. Here's a descriptionof the data preprocessing steps typically applied to the Reuters_50_50 dataset:

- Text Cleaning: The first step is to clean the text by removing any irrelevant information or noise that may interfere with the analysis. This includes removing HTML tags, special characters, punctuation marks, and numbers. Additionally, any metadata or header information specific to the Reuters dataset can be extracted and stored separately for future reference.
- Tokenization: Tokenization involves splitting the text into individual words or tokens. This step helps create a structured text representation for further analysis. Common approaches for tokenization include using whitespace as a delimiter or employing more advanced techniques such as natural language processing (NLP) libraries or regular expressions.
- Stop Word Removal: Stop words are commonly used words in a language (e.g., "the," "and," "is") that do not carry significant meaning and can be safely removed from the text. Stop word removal helps reduce noise in the dataset and can improve the efficiency of subsequent analysis steps.
- Stemming or Lemmatization: Stemming and lemmatization are techniques used to reduce words to their root or base form. Stemming involves removing prefixes and suffixes from words, while lemmatization maps words to their dictionary form. These techniques help in reducing the dimensionality of the dataset and capturing the core semantic meaning of words.
- Data Balancing: Depending on the distribution of documents across authors in the Reuters 50_50 dataset, it may be necessary to address the class imbalance. Class imbalance occurs when there is a significant difference in the number of documents per author. Techniques such as oversampling, undersampling, or generating synthetic samples can be employed to balance the dataset and avoid biases in the model training process. These data preprocessing steps help transform the raw Reuters_50_50 dataset into a structured and cleaned representation suitable for author identification and attribution tasks. They enable researchers to extract meaningful features from the text and build robust models that can accurately identify the authors of unseen documents (bullet list)

C. Techniques for Feature Extraction

Feature extraction plays a crucial role in author identification tasks using the Reuters_50_50 dataset. It involves transforming the preprocessed text into a numerical representation that captures relevant information about the writing style and content of each document. Here's a description of common feature extraction techniques used for author identificationwith the Reuters_50_50 dataset:

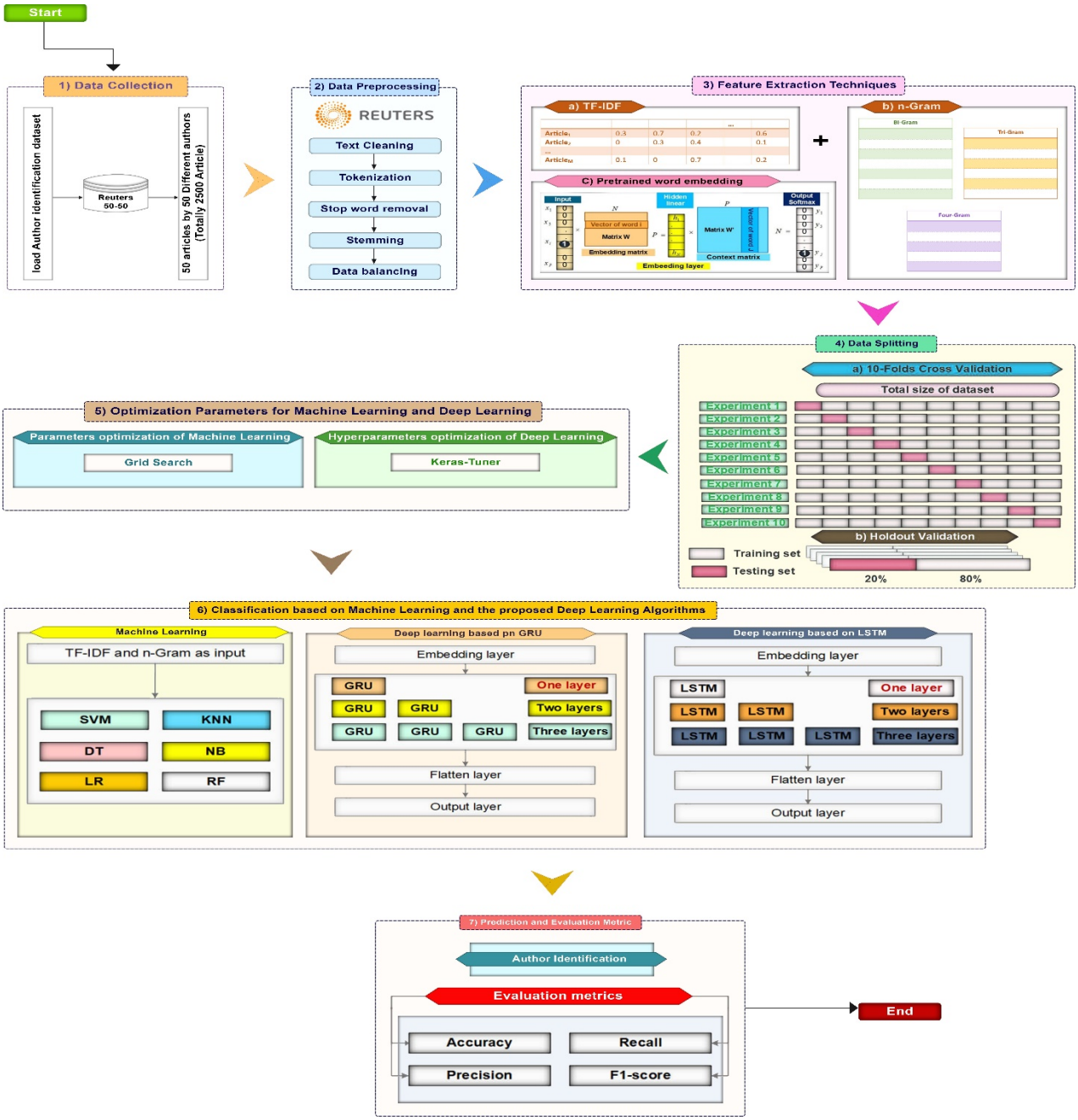


Figure 1: the proposed framework for Author identification problem

- Bag-of-word (BOW) Representation: The bag-of-words model represents each document as a vector of word frequencies. It disregards the order and context of words but captures their presence in the text. The BOW representation can be constructed using simple word counts or more sophisticated approaches like term frequency-inverse document frequency (TF-IDF), which considers the importance of words based on their frequency in the document and across the dataset.
- N-grams: are consecutive sequences of N words in the text. By extracting N-grams, such as uni-grams (single words), bigrams (two-word sequences), or trigrams (three-word sequences), it is possible to capture more contextual information about the writing style. N-grams can be counted or represented using TF-IDF weights.
- Syntactic Features**:** capture the structural characteristics of the text. These features include sentence length, average word length, part-of-speech (POS) tags, and syntactic parse tree-based features. For example, the frequency of specific POS tags (e.g., nouns,verbs) or syntactic patterns (e.g., noun phrases, verb phrases) can be calculated to represent the writing style.
- Readability Measures**:** quantifythe complexity of the text and can provide insights into the writing style of authors. Measures like Flesch-Kincaid Grade Level, Gunning Fog Index, and Coleman-Liau Index estimate the difficulty of reading the text basedon factors such as sentence length and word complexity.
- Lexical features: focus on the vocabulary used by authors. These features include word frequencies,

vocabulary richness (e.g., number of unique words), average word frequency (e.g., based on external corpora like the Brown Corpus), and lexical diversity measures (e.g., Type-Token Ratio).

- Stylometric features: capture various aspects of writing style, such as punctuation usage, capitalization patterns, sentence structure, and word usagepatterns. These features can be calculated using statistical measures like mean, standard deviation, or entropy of specific linguistic characteristics.
- Semantic Features (Optional): Depending on the specific objectives of the author identification task, semantic features can be incorporated. These features involve leveraging techniques like word embedding (e.g., Word2Vec, GloVe) or pre-trained language models (e.g., BERT, GPT) to capture the semantic meaning of words and phrases in the text.

These feature extraction techniques help convert the preprocessed text of the Reuters_50_50 dataset into numerical representations that capture various aspects of the writing style and content. By incorporating these features into author identification models, researchers can build effective models that leverage the distinctive patterns exhibited by different authors.

D. Data Splitting

During this stage, the dataset is partitioned into the training set and the testing set through the utilization of two methods: a holdout approach with an 80% allocation to training and 20% to testing, and a 10-Folds cross-validation (CV) technique. The core principle behind 10-Fold CV involves splitting the dataset into ten segments or folds, with nine of them employed for training and one for testing in a cyclical manner. This process of data division is repeated ten times, as denoted by "k" (k = 10)

E. Tuning Parameters for both Machine Learning (ML) and Deep Learning (DL)

To optimize hyperparameters for machine learning methods applied to author identification tasks using the Reuters_50_50 dataset. the following techniques are applied to optimize hyperparameters for machine learning.

- Grid search: involves exhaustively searching through a predefined grid of hyperparameter values. For each combination of hyperparameters, you train and evaluate the model on a validation set using cross-validation. Grid search helps identify the best hyperparameter values based on the highest validation performance.
- Random search: randomly samples hyperparameter values from predefined ranges. Instead of systematically exploring all possible combinations likegrid search, it randomly selects combinations for evaluation. Random search is less computationally expensive than grid search and has been shown to be effective in finding good hyperparameter values.
- Bayesian optimization: is a sequential model-based optimization approach. It uses asurrogate model to approximate the performance of different hyperparameter settings and updates this model iteratively based on the evaluation results. Bayesian optimization intelligently selects the next set of hyperparameters to evaluate, aiming to find the optimal values with fewer iterations compared to grid or random search.
- Automated Hyperparameter Tuning Libraries: There are several libraries and frameworks available that automate the process of hyperparameter tuning. Examples include scikit-learn's GridSearchCV and Randomized-SearchCV, Optuna, and Keras Tuner. These libraries provide convenient interfaces to define hyperparameter search spaces, perform cross-validation, and optimize hyperparameters using various algorithms.
- Early Stopping: Early stopping is a technique used during training to prevent overfitting and find an optimal number of training epochs. It monitors the model's performance on a validation set during training and stops training when the performance starts to degrade. By selecting the best model based on validation performance, you can avoid overfitting and determine an optimal stopping point. It's important to note that hyperparameter optimization is an iterative process that requires experimentation and evaluation

F. Classification based on ML and the proposed DL

The mentioned datasets were passed into 6 different Machine Learning algorithms which were Logistic Regression, Gradient Boosting, K Nearest Neighbor (k-NN), Random Forest, Decision Tree, and Naive Bayes. For each of the algorithms there were statistics generated, these statistics were: Accuracy, Recall, Precision, and Specificity. Afterwards, the results were charted and compared. The results, charts, and the discussion of the results can be found later in the paper

- Naive Bayes (NB) is a probabilistic machine learning algorithm that employs Bayes' theorem. Maximum Posterior decision laws executed within a Bayesian system are used to generate classifications.
- K-Nearest Neighbor (KNN) is one of the most effectiveand fundamental classification techniques. It makes no assumptions about data and is usable for classification purposes in situations where very little or no prior knowledge of the distribution of data is available. This algorithm is utilized to locate The value of the founddata points in it is assigned to the training set data points that are closest to the target-valued data point.
- Logistic Regression (LR) serves as a predictive technique used for analysis, functioning on the foundation of probability principles.
- Support Vector Machine (SVM) is an algorithm aimed at discovering the optimal hyperplane that effectively separates classes, thus enabling classification.
- Decision Tree (DT) is predominantly employed in supervised machine learning. It consists of nodes and branches, where nodes symbolize attribute tests, branches signify outcomes of these tests, and leaf

nodes denote class assignments.

- Random Forest (RF) employs an ensemble of decision trees to enhance adaptability. This approach enhances the efficacy of decision trees by utilizing multiple trees simultaneously

G. The proposed DL models:

Figure 1. Depicts the proposed deep neural network architecture. The developed models are applied to Reuter_50_50 dataset to identify author. It employs a word embedding matrix as input to an embedding layer, which is preceded by an embedding layer, followed by hidden layers consisting of Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), a flattening layer, and finally an output layer. Using the word2vec model, word embedding vectors representing article news are generated. Each DL model's embedding layer, concealed layer, and output layer are defined as follows:

- Long Short-Term Memory network (LSTM): The initial among the suggested Deep Learning (DL) models is constructed upon the LSTM architecture a form of recurrent neural network (RNN). This architecture is engineered to effectively handle time series data and address challenges of long-term dependencies. Its performance surpasses that of a conventional RNN. While a standard RNN includes a solitary layer (tanh), an LSTM introduces three interconnected gates: the Forget gate, Update gate, and gates for cell updating and output calculate from the input sequence Suppose $x = x_1, x_2, x_3....x_s$ to the output sequence, by iterative calculating $x = 1, ...s$ .The activation unit in the network of S. The definitions of input gate i_t forgetting gate f_t, and output gate o_t in LSTM network are shown in the following Equations

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1}) \quad (1)$$

$$ft = \sigma(Wfxxt + Wfhht-1) \quad (2)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1}) \quad (3)$$

$$\hat{h}_t = W_{hx}x_t + W_{hh}h_{t-1} \quad (4)$$

$$C_t = tanh(X^*_t U_c + H^*_{t-1} W_c) \quad (5)$$

Within the equation, the weight matrix is denoted as "w," while "sigma" signifies a sigmoid function. The symbols "i," "f," "o," and "c" correspond to the input gate, forgetting gate, output gate, and storage cell, respectively. In the context of the LSTM, variables such as "ht," "it," "ft," and "ot" are present. The symbol "σ" represents the activation function employed for both input and output in the cell unit, typically referred to as tanh.

- The Gated Recurrent Unit (GRU): is the second model proposed in our framework, it is the most recent form of Recurrent Neural Networks (RNNs) and is very similar to LSTM. GRU combines the forget and input layers into a single update gate, which is the primary distinction between LSTM and GRU. The GRU has eliminated the cell state and is now transferring information using the concealed state. There are just two gates, a reset gate and an update gate.

H. Outcome Prediction and Evaluation Metrics

Four standard performance metrics; Accuracy (Ac), Precision (Pr), Recall (Re), and F1-score (F1) are used to evaluate the performanceof the proposed models. They are calculated as follows:

$$Ac = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

$$Pr = \frac{TP}{TP+FP} \quad (7)$$

$$Re = \frac{TP}{TP+FN} \quad (8)$$

$$F = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (9)$$

IV. RESULTS AND DISCUSSIONS

TABLE I. DEEP LEARNING OPTIMAL HYPERPARAMETERS FOR AUTHOR IDENTIFICATION DATASET ROUTERS5050 (C5050)

Datasets	**Routers5050 dataset**	
Models	***Neurons Num***	***Dropout***
LSTM one layer	250	0.4
LSTM two layers	[350,310]	[0.2,0.1]
LSTM three layers	[320,440,70]	[0.6,0.5,0.4]
GRU one layer	370	0.2
GRU two layers	[450,50]	[0.5,0.1]
GRU three layers	[330,480,230]	[0.3,0.7,0.2]

A. HYPERPARAMETERS OPTIMIZATION METHODS FOR THE PROPOSED DL MODELS

During this phase, the Keras-tuner library is utilized to determine the optimal hyperparameters for the hidden layers and dropout layers [58] for Author identification in long short-term memory (LSTM), and gated recurrent units (GRUs). Each model utilized between one and three concealed layers. Inaddition, a dropout layer was utilized in conjunction with a concealed layer, and the ReLU activation function and Adam optimizer were incorporated into the output layer. Using the Keras Tuner library, the total number of neurons/layer and dropout rate for each data set of the proposed models are optimized, as shown in Table II.

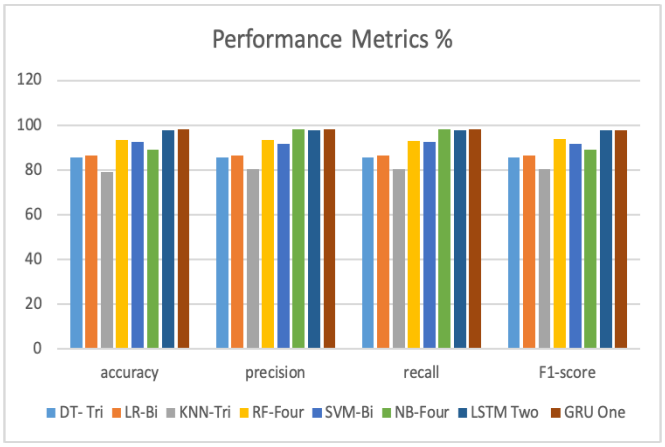
B. EXPERIMENTAL RESULTS AND DISCUSSION

To assess the effectiveness of both ML and DL models in author identification, we conducted an evaluation using the Reuters_50_50 dataset employing two distinct learning strategies: Holdout validation with an 80% training and 20% testing split, as well as a 10-Folds cross-validation approach. It's important to highlight that for ML methods (DT, KNN, LR, RF, SVM, and NB), feature extraction is accomplished

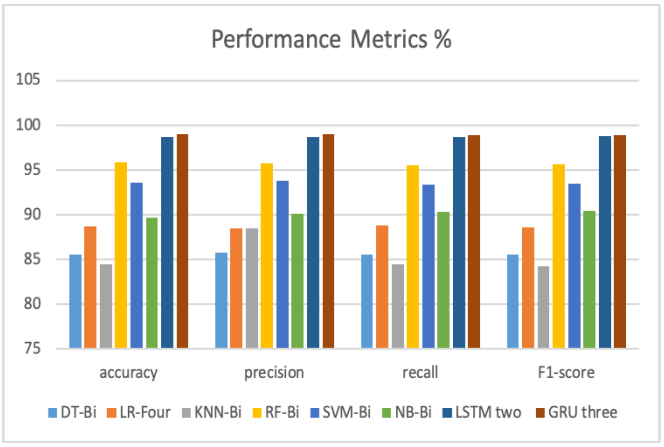
using TF-IDF across various scenarios, including Bi-Gram, Tri-Gram, and Four-Gram. On the other hand, Deep Learning models utilized an embedding layer generated by AraVec

TABLE II. Deep Learning Optimal Hyperparameters for Author Identification Dataset Routers5050 (C5050)

Type	Classifiers	Extraction methods	Holdout Validation (80% -20%)				10-Folds Cross Validation			
			ACC	PREC	REC	F1	ACC	PREC	REC	F1
Machine Learning	DT	TF-IDF (Bi-Gram)	82.34	83.11	83.43	83.19	**85.54±0.77**	**85.78±0.65**	**85.56±0.87**	**85.56±0.86**
		TF-IDF (Tri-Gram)	**85.45**	**85.56**	**85.54**	**85.45**	85.55±0.76	85.65±0.65	86.55±0.65	86.55±0.65
		TF-IDF (Four-Gram)	83.43	83.33	83.43	83.43	84.44±0.65	84.55±0.65	84.54±0.56	84.55±0.55
	KNN	TF-IDF (Bi-Gram)	70.20	71.22	69.20	70.22	**84.43±1.66**	**88.45±1.33**	**84.43±1.66**	**84.22±1.77**
		TF-IDF (Tri-Gram)	**79.23**	**80.34**	**80.44**	**80.43**	83.22±1.73	83.65±1.12	83.34±1.43	84.12±1.66
		TF-IDF (Four-Gram)	75.32	76.43	76.45	76.43	82.22±1.65	82.76±1.22	82.34±1.78	82.65±1.99
	LR	TF-IDF (Bi-Gram)	**86.22**	**86.34**	**86.34**	**86.32**	87.65±0.45	88.43±0.45	88.54±0.45	88.23±0.43
		TF-IDF (Tri-Gram)	86.11	85.54	85.87	85.67	87.45±0.54	87.54±0.52	87.23±0.43	87.55±0.34
		TF-IDF (Four-Gram)	85.34	85.34	85.44	85.55	**88.65±0.55**	**88.45±0.34**	**88.77±0.54**	**88.55±0.66**
	RF	TF-IDF (Bi-Gram)	93.22	93.65	93.45	93.23	**95.87±0.76**	**95.76±0.45**	**95.55±0.56**	**95.66±0.77**
		TF-IDF (Tri-Gram)	93.12	93.11	93.21	93.11	94.55±0.52	94.0±0.58	94.65±0.55	94.92±0.56
		TF-IDF (Four-Gram)	**93.43**	**93.18**	**93.01**	**93.98**	93.91±0.65	94.34±0.62	94.33±0.60	94.33±0.61
	SVM	TF-IDF (Bi-Gram)	**92.45**	**91.41**	**92.34**	**91.66**	**93.55±0.77**	**93.76±0.56**	**93.34±0.77**	**93.45±0.77**
		TF-IDF (Tri-Gram)	91.23	91.33	91.34	91.34	91.55±0.66	91.76±0.77	91.55±0.66	91.34±0.55
		TF-IDF (Four-Gram)	91.77	91.54	91.34	91.34	91.86±0.55	91.33±0.56	91.87±0.61	91.93±0.62
	NB	TF-IDF (Bi-Gram)	87.77	87.99	87.76	88.11	**89.65±0.66**	**90.11±0.54**	**90.32±0.65**	**90.45±0.54**
		TF-IDF (Tri-Gram)	88.55	88.43	88.43	88.67	87.99±0.90	87.98±0.91	87.55±0.90	87.78±0.88
		TF-IDF (Four-Gram)	**89.03**	**89.01**	**89.03**	**89.22**	87.55±1.32	87.44±1.43	87.44±1.23	87.23±1.22
Deep Learning	LSTM (1 layer)	Embedding layer	95.88	95.67	95.65	95.83	96.99±0.65	96.66±0.34	96.55±0.44	96.65±0.29
	LSTM (2 layers)	Embedding layer	**97.65**	**97.66**	**97.87**	**97.77**	**98.66±0.21**	**98.65±0.32**	**98.67±0.33**	**98.77±0.22**
	LSTM (3 layers)	Embedding layer	96.65	96.65	96.65	96.65	98.70±0.14	98.72±0.12	98.72±0.22	98.72±0.11
	GRU (1 layer)	Embedding layer	**97.99**	**97.98**	**97.98**	**97.73**	98.11±0.10	98.12±0.23	98.21±0.14	98.12±0.08
	GRU (2 layers)	Embedding layer	96.65	96.77	96.88	96.87	97.87±0.10	97.66±0.23	97.76±0.14	97.91±0.08
	GRU (3 layers)	Embedding layer	96.96	96.99	96.99	96.87	**98.99±0.03**	**98.96±0.02**	**98.89±0.00**	**98.89±0.05**



(a) The hold out performance results



(b) The cross validation performance results

Figure 2: The performance metrics for C50_50 Dataset using ML and DL methods

(pre-trained word embedding) as input for the LSTM and GRU models.

- Hold out Validation

This section presents the performance outcomes of both ML and DL models using holdout validation on the C50_50 dataset. Table II presents the metrics values for Accuracy (ACC), Recall (REC), Precision (PREC), and F1-score (F1). Upon analyzing the results of the ML methods, it becomes evident that RF based on TF-IDF (Four-Gram) emerges as the most proficient classifier when compared to its counterparts, boasting an ACC of 93.43%, REC of 93.01%, PREC of 93.18%, and F1 of 93.93%. This success can be attributed to RF's inherent ability to autonomously select features. Additionally, the KNN classifier using Tri-Gram features attains the lowest performance metrics (ACC of 79.23%, PREC of 80.34%, REC of 80.44% and F1 of 80.43%). Due to KNN's nature as a sluggish learner, this behavior is obvious and logical. Consequently, the classification assignment is accomplished solely through the computation of Euclidean distance, which has a detrimental effect on performance inthe case of a high-dimensional space of representation.

– 10-Folds Cross Validation

In the context of 10-Folds Cross Validation, this section aims to illustrate the outcomes of splitting the C50_50 dataset through the use of 10-Folds cross-validation. Both ML and DL techniques were employed for this purpose, and the results are presented in Table II. When examining the results of 10-Folds cross-validation in terms of accuracy, recall, precision, and F1-score using fundamental ML methods, it becomes apparent that RF based on TF-IDF with Bi-Gram features achieves a notable performance enhancement. This is reflected in its metrics of 95.87% accuracy, 95.67% precision, 95.55% recall, and 95.66% F1-score. The construction of multiple trees based on distinct subsets of features enables the improvement of the classification rate, resulting in improved results from RF. Additionally, dividing into multiple folds increases the reparability between various classes of authors. Due to the Euclidean distance used to distinguish between distinct

authors, when applying 10-Folds cross-validation, K-Nearest Neighbors (KNN) continues to perform as the least effective classifier for the C50_50 dataset. On the other hand, concerning DL models utilizing the same cross-validation approach, the GRU model with three layers consistently demonstrates superior performance across all metrics. Specifically, the results for GRU are an accuracy of 98.99%, precision of 98.96%, recall and F1-score are 98.89% respectively.

It is noteworthy to emphasize that the weakest DL model, LSTM, still surpasses the best performing ML method, RF by margins of 2.79% in accuracy, 3.12% in recall, 2.81% in precision, and 3.11% in F1-score.

The results of Deep Learning (DL) based on GRU (3 layers)with embedding layer provide an effective combination for the50_50 dataset, which represents a real challenge.

C. Graphical analysis

Regarding the results of DL on the C50_50 dataset with hold-out validation, GRU (1 layer) achieves an improvement of 4.56% in accuracy, 4.97% in recall, 4.8% in precision, and 3.75% in F1-score. It's important to emphasize that even the least favorable outcomes of the DL model, particularly the LSTM with two layers, surpass the performance of the ML method's top model relying on RF. This improvement can be attributed to two key factors: Firstly, the utilization of word2vec as a pre-trained neural network for feature extraction; secondly, the incorporation of memory mechanisms inherent to LSTM models.

Figure 2 provides a concise overview of the optimum metric values, encompassing ACC, PREC, REC, and F1, derived from both ML methods and DL models for the C50_50 dataset. This evaluation considers two distinct splitting strategies, namely holdout and 10-Folds cross-validation. Notably, GRU (Three Layers) demonstrates superior performance across all categories, surpassing both ML and DL algorithms in terms of accuracy, recall, precision, and F1 for both splitting strategies.

Moreover, it's noteworthy that the deep learning model centered around GRU with three layers consistently outperforms other DL variations as well as fundamental ML methods in the context of the C50_50 dataset, regardless of the splitting strategy employed.

V. CONCLUSION AND FUTURE WORK

Authorship identification is an essential task which is permits us to identify the most probable author of articles, news or messages. Authorship identification can be used for duties like identifying anonymous authors, detecting plagiarism, and locating ghost writers. In this undertaking, we approachedthis issue from a variety of perspectives, utilizing various deep learning models and datasets. In addition, the extraction of features is realized using TF-IDF based on Bi-Gram, Three-Gram and Four-Gram. Regarding the (C50_50 dataset), the best testing results are achieved by the deep learning model GRU employing both learning strategies (Hold out and 10-Folds). The performance measurement results are highly significant in terms of accuracy, precision, recall, and F1 scorebecause they exceed the respective benchmarks of 97% and 98% for Hold-out and 10-Folds. The most accurate classifiers in machine learning are RF with 93.43% for Hold out and RF with 95.78% based on 10 folds cross-validation.

In our future work, we will intend to use ensample learning in conjunction with two distinct feature extraction methods.

References

[1] F. Mosteller and D. L. Wallace, “Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers,” *Journal of the American Statistical Association*, vol. 58, no. 302, pp. 275–309, 1963.

[2] D. Bogdanova and A. Lazaridou, “Cross-language authorship attribution.” in *LREC*. Citeseer, 2014, pp. 2015–2020.

[3] Y. Zhao, J. Zobel, and P. Vines, “Using relative entropy for authorship attribution,” in *Information Retrieval Technology: Third Asia Information Retrieval Symposium, AIRS 2006, Singapore, October 16-18, 2006. Proceedings 3*. Springer, 2006, pp. 92–105.

[4] J. A. Nasir, N. Görnitz, and U. Brefeld, “An off-the-shelf approach to authorship attribution,” in *Proceedingsof COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 895–904.

[5] C. Sanderson and S. Guenter, “Short text authorship attribution via sequence kernels, markov chains and authorunmasking: An investigation,” in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 2006, pp. 482–491.

[6] M. Fabien, E. Villatoro-Tello, P. Motlicek, and S. Parida, “Bertaa: Bert fine-tuning for authorship attribution,” in *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, 2020, pp. 127– 137.

[7] J. H. Clark and C. J. Hannon, “A classifier system for author recognition using synonym-based features,” in *MICAI 2007: Advances in Artificial Intelligence: 6th Mexican International Conference on Artificial Intelligence, Aguascalientes, Mexico, November 4-10, 2007. Proceedings 6*. Springer, 2007, pp. 839–849.

[8] A. Dhar, H. Mukherjee, S. Sen, M. O. Sk, A. Biswas, T. Gonçalves, and K. Roy, “Author identification from literary articles with visual features: A case study with bangla documents,” *Future Internet*, vol. 14, no. 10, p. 272, 2022.

[9] T. Kumar, S. Gowtham, and U. K. Chakraborty, “Comparing word embeddings on authorship identification,” in *Applied Soft Computing*. Apple Academic Press, 2022, pp. 177–194.

[10] S. Das and P. Mitra, “Author identification in bengali literary works,” in *Pattern Recognition and Machine*

Intelligence: 4th International Conference, PReMI 2011, Moscow, Russia, June 27-July 1, 2011. Proceedings 4. Springer, 2011, pp. 220–226.

[11] M. T. Hossain, M. M. Rahman, S. Ismail, and M. S. Islam, "A stylometric analysis on bengali literature for authorship attribution," in *2017 20th International Conference of Computer and Information Technology (ICCIT)*. IEEE, 2017, pp. 1–5.

[12] U. Pal, A. S. Nipu, and S. Ismail, "A machine learning approach for stylometric analysis of bangla literature," in *2017 20th International Conference of Computer and Information Technology (ICCIT)*. IEEE, 2017, pp. 1–5.

[13] R. Tripodi and S. L. Pira, "Analysis of Italian word embeddings," *arXiv preprint arXiv:1707.08783*, 2017.

[14] L. De Vine, M. Kholghi, G. Zuccon, L. Sitbon, and A. Nguyen, "Analysis of word embeddings and sequencefeatures for clinical information extraction," in *Australasian Language Technology Association Workshop 2015: Proceedings of the Workshop*. Australasian Language Technology Association (ALTA), 2015, pp. 21–30.

[15] H. Liu, "Sentiment analysis of citations using word2vec,"*arXiv preprint arXiv:1704.00177*, 2017.

[16] A. Abbasi, A. R. Javed, F. Iqbal, Z. Jalil, T. R. Gadekallu, and N. Kryvinska, "Authorship identification using ensemble learning," *Scientific reports*, vol. 12, no. 1, p. 9537, 2022.

[17] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich, and M. Sedlmair, "More than bags of words: Sentiment analysis with word embeddings," *Communication Methods and Measures*, vol. 12, no. 23, pp. 140– 157, 2018.

[18] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[19] N. Neggaz, "Swarming behavior of harris hawks optimizer for arabic opinion mining."

[20] D. S. AbdElminaam, N. Neggaz, I. A. E. Gomaa, F. H. Ismail, and A. Elsawy, "Aommpa: Arabic opinion mining using marine predators algorithm based feature selection," in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. IEEE, 2021, pp. 395–402.

[21] D. S. Abdelminaam, N. Neggaz, I. A. E. Gomaa, F. H. Ismail, and A. A. Elsawy, "Arabic dialects: An efficient framework for Arabic dialects opinion mining on twitter using optimized deep neural networks," *IEEE Access*, vol. 9, pp. 97 079–97 099, 2021.

[22] D. S. AbdElminaam, I. A.E. Ahmed, and F. Sakr, "Scbiot: Smart cane for blinds using iot," in *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. IEEE, 2022, pp. 371–377.

[23] M. A. Ali, R. Orban, R. Rajammal Ramasamy, S. Muthusamy, S. Subramani, K. Sekar, F. Rajeena PP, I. A. E. Gomaa, L. Abulaigh, and D. S. A. Elminaam, "A novel method for survival prediction of hepatocellular carcinoma using feature-selection techniques," *Applied Sciences*, vol. 12, no. 13, p. 6427, 2022.

[24] D. S. AbdElminaam, E. H. Houssein, M. Said, D. Oliva, and A. Nabil, "An efficient heap-based optimizer for parameters identification of modified photovoltaic models," *Ain Shams Engineering Journal*, vol. 13, no. 5, p. 101728, 2022.

[25] D. S. Abd Elminaam, S. A. Ibrahim, E. H. Houssein, and S. M. Elsayed, "An efficient chaotic gradient-based optimizer for feature selection," *IEEE Access*, vol. 10,pp. 9271–9286, 2022.

[26] S. A. Ibraheem, "Hmfc: Hybrid Modlem-fuzzy Classifier for liver diseases diagnose.", 2019

[27] K. K. Patro, J. P. Allam, B. C. Neelapu, R. Tadeusiewicz, U. R. Acharya, M. Hammad, O. Yildirim, and P. Pławiak, "Application of kronecker convolutions in deep learning technique for automated detection of kidney stones with coronal CT images," *Information Sciences*, vol. 640, p. 119005, 2023.

[28] A. J. Prakash, "Capsule network for the identificationof individuals using quantized ecg signal images," *IEEE Sensors Letters*, vol. 6, no. 8, pp. 1–4, 2022.

[29] M. Hammad, S. A. Chelloug, R. Alkanhel, A. J. Prakash, A. Muthanna, I. A. Elgendy, and P. Pławiak, "Automated detection of myocardial infarction and heart conduction disorders based on feature selection and a deep learning model," *Sensors*, vol. 22, no. 17, p. 6503, 2022.

[30] B. Venkata Phanikrishna, A. Jaya Prakash, and C. Suchismitha, "Deep review of machine learning techniques on detection of drowsiness using eeg signal," *IETE Journal of Research*, pp. 1–16, 2021.

[31] K. R. Pedada, B. Rao, K. K. Patro, J. P. Allam, M. M. Jamjoom, and N. A. Samee, "A novel approach for brain tumour detection using deep learning based technique," *Biomedical Signal Processing and Control*, vol. 82, p. 104549, 2023.

[32] A. J. Prakash, K. K. Patro, S. Saunak, P. Sasmal, P. L. Kumari, and T. Geetamma, "A new approach of transparent and explainable artificial intelligence technique for patient-specific ecg beat classification," *IEEE SensorsLetters*, 2023.

[33] K. K. Patro, J. P. Allam, M. Hammad, R. Tadeusiewicz, and P. Pławiak, "Scovnet: A skip connection-based feature union deep learning technique with statistical approach analysis for the detection of covid-19," *Biocybernetics and Biomedical Engineering*, vol. 43, no. 1, pp. 352–368, 2023.

[34] A. J. Prakash, K. K. Patro, M. Hammad, R. Tadeusiewicz, and P. Pławiak, "Baed: A secured biometric authentication system using ECG signal based on deep learning techniques," *Biocybernetics and Biomedical Engineering*,vol. 42, no. 4, pp. 1081–1093, 2022.

[35] A. J. Prakash, K. K. Patro, S. Samantray, P. Pławiak, and M. Hammad, "A deep learning technique for

biometric authentication using ECG beat template matching," *Information*, vol. 14, no. 2, p. 65, 2023.

[36] K. K. Patro, A. J. Prakash, S. Samantray, J. Pławiak, R. Tadeusiewicz, and P. Pławiak, "A hybrid approachof a deep learning technique for real–time ECG beat detection," *International journal of applied mathematics and computer science*, vol. 32, no. 3, pp. 455–465, 2022.

[37] K. K. Patro, A. Jaya Prakash, M. Jayamanmadha Rao, and P. Rajesh Kumar, "An efficient optimized feature se lection with machine learning approach for ECG biometricrecognition," *IETE Journal of Research*, vol. 68, no. 4, pp. 2743–2754, 2022.

[38] K. K. Patro, S. P. R. Reddi, S. E. Khalelulla, P. Rajesh Kumar, and K. Shankar, "ECG data optimization for biometric human recognition using statistical distributed machine learning algorithm," *The Journal of Supercomputing*, vol. 76, pp. 858–875, 2020.

[39] K. K. Patro and P. R. Kumar, "Machine learning classification approaches for biometric recognition system using ECG signals." *Journal of Engineering Science & Technology Review*, vol. 10, no. 6, 2017.

[40] Kiran Kumar Patro, P. Rajesh Kumar, "Effective feature extraction of ECG for biometric application," *Procedia computer science*, vol. 115, pp. 296–306, 2017.