applied sciences

MDPI

*Article*

# Gender Recognition of Bangla Names Using Deep Learning Approaches

**Md. Humaun Kabir [1], Faruk Ahmad [2], Md. Al Mehedi Hasan [3] and Jungpil Shin [3,*]**

[1] Department of Computer Science and Engineering, Bangamata Sheikh Fojilatunnesa Mujib Science & Technology University, Jamalpur 2012, Bangladesh
[2] AI Engineer, Semantics Lab, Begum Rokeya University, Rangpur 5404, Bangladesh
[3] School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan
* Correspondence: jpshin@u-aizu.ac.jp; Tel.: +81-242-37-2704

**Abstract:** The name of individuals has a specific meaning and great significance. Individuals' names generally have substantial gender differences, and explicitly, Bengali names usually have a solid sexual identity. We can determine if a stranger is a man or a woman based on their name with remarkably suitable precision. In this research, we primarily conducted a thorough investigation into gender prediction based on a person's name using DL-based methods. While various techniques have been explored for the English language, there has been little progress in the Bengali language. We address this gap by presenting a large-scale experiment with 2030 Bangladeshi unique names. We used both convolutional neural network (CNN)- and recurrent neural network (RNN)-based deep learning methods to infer gender from the Bangladeshi names in the Bengali language. We presented the one-dimensional CNN (Conv1D), simple long short-term memory (LSTM), bidirectional LSTM, stacked LSTM, and combined Conv1D and stacked bidirectional LSTM-based models and evaluated the performance of each scheme using our own dataset. Experimental results are analyzed on the basis of accuracy, precision, recall, F1-score, ROC AUC score, and loss performance metrics. The performance evaluative results show that Conv1D outperforms with 91.18% accuracy, which is likely to improve as the size of the training data grows.

**Keywords:** NLP; gender prediction; gender classification; feature extraction; deep learning; LSTM; CNN

check for updates

**Citation:** Kabir, M.H.; Ahmad, F.; Hasan, M.A.M.; Shin, J. Gender Recognition of Bangla Names Using Deep Learning Approaches. *Appl. Sci.* **2023**, *13*, 522. https://doi.org/10.3390/app13010522

Academic Editor: Zeng Nianyin

Received: 28 October 2022
Revised: 24 December 2022
Accepted: 27 December 2022
Published: 30 December 2022

## 1. Introduction

Bengali, commonly known as Bangla, is a widely spoken language in Bangladesh as well as India's eastern regions (West Bengal and Kolkata). Bengali is the native language of more than 228 million people and ranks sixth in the world in terms of native speakers. Bengali speakers are 265 million worldwide, placing it sixth in terms of total speakers [1,2]. Nowadays, Bangla natural language processing (BNLP) has become a focal point in the area of NLP and AI for both academic and industrial researchers [3–6]. As a preprocessing step in many AI and NLP applications, gender recognition of names is required. It can help to improve the performance of applications such as coreference resolution (discover all phrases in a text that refer to the same item), machine translation (transforming a text from one natural language to another), automatic query replying, content-based ad, and automatic retrieval of specific information, etc. Person's names have important relevance in several sectors since they are an essential element of social groups. Many tasks, such as testing for fairness using machine learning algorithms for content recommendation, need knowledge of users' demographic information, which includes gender [7,8]. As a result, those registered users who do not reveal their gender must be inferred by using their information, such as name provided during the registration process. We proposed various approaches for automatic gender prediction from a person's name, described in detail in Sections 5 and 6. At present, many academic and commercial investigators have been studying and exploring the automated identification of human names [9–12]. It is known from the literature reviewing that many researchers of academic institutions and industries have been working on NLP using AI, ML, and DL-based algorithms, but to our knowledge, we did not find many research works on gender inference from names. The majority of previous works available in the literature have focused on English, ignoring other languages such as Bangla. The currently available software for determining the sexual identity of people uses dictionary lookup algorithms. Currently, there is no freely available system/method to determine the sexual identity from Bangla names for research, industrial, or other uses. Many attempts are also being made to make the Bangla language more accessible on the web and in technological areas. CNN- and LSTM-based classification approaches find use in a large number of NLP tasks and other machine learning applications specifically for text and time sequence data [13–16]. Our main contributions are Bangla name data collection, cleaning and preprocessing of collected data, preparation of a suitable dataset of Bangla names, an in-depth inspection of different word-level characteristics of Bangladeshi names in the Bengali language that differentiate between the two sexes, identification of the attributes that are very useful in classification, and presentation of a state-of-the-art gender identification technique based on CNN and LSTM. This research is mostly concerned with the gender recognition of Bengali names. In this paper, we built Conv1D, LSTM, bidirectional LSTM, stacked LSTM, and combined Conv1D and stacked bidirectional LSTM-based gender classifier and critically evaluated the performance of each classifier using our prepared own dataset through the analysis of the values of accuracy, precision, recall, F1-score, ROC AUC score, and loss metrics. The following is a breakdown of the paper's structure. The related works will be covered in the second section, details of dataset preparation are mentioned in the third section, the proposed framework for the development of the gender classifier is presented in Section 4, and the description of the DNN-based experimental models will be introduced in the fifth part. The sixth section of this article will discuss the experimental setup and procedure, and analysis of the outcomes of the developed DNN-based gender classifiers will be presented in Section 7. The performance comparison and discussion will be covered in Section 8, and the last portion will summarize the advantages and draw conclusions for future research.

## 2. Related Works

In this section, some relevant works are described briefly. The authors of [17] focused their study on gender identification of English names using different machine learning algorithms, including support vector classifier, naive Bayes classifier, decision tree classifier, and maximum entropy classifier. SVM classifiers and naive Bayes classifiers are two of the most effective ML-based models. Authors did related work in [18] to recognize the Chinese organization names automatically based on conditional random fields. Authors proposed several machine learning approaches such as LSTM and BERT to predict users' genders based on their first names and conducted their experiment on a very large real-world dataset of 21M unique first names in [19]. They found that genders can be classified very effectively using the composition of the name strings and also demonstrated that the performance could be further enhanced when utilizing both the first and the last names through the dual-LSTM algorithm. The authors of [20] looked at and used feed-forward and recurrent deep neural network models to classify gender based on the first name, including MLP, RNN, GRU, CNN, and Bi-LSTM. The models are trained and evaluated using a dataset of Brazilian names. To assess the models' performance, they looked at the accuracy, recall, precision, and confusion matrix. The findings show that gender prediction can be made using a feature extraction method that looks at names as a group of strings and that some models can correctly identify gender in more than 90% of cases. Authors proposed a method in [21] based on three types of features: word endings, character n-*g*, and dictionary of names, all of which are integrated with a linear supervised model to determine a person's gender based on his or her full name. They conducted a large-scale experiment on a dataset of 100,000 Russian full names from Facebook and found that the suggested simple and computationally efficient technique produces outstanding results, with an accuracy of up to 96%. In [22], the authors proposed a support vector machine (SVM)-based classification method for predicting the gender of Indian names. They discovered different characteristics based on morphological analysis and then proposed a unique technique of combining these traits with n-*g* suffixes, which outperforms the baseline approach significantly. The system achieved an F1 score of 94.9%, according to the authors of this article. In [23], the authors made an investigation on gender recognition for characters in text. Authors demonstrated that they could detect a character's gender based on the variations in the words when a man or a woman is mentioned in the text, besides the word characteristics of a character's name. A large number of significant words with gender differences, gender bias feature words, and gender bias personal appellations are obtained depending on different explanations of males or females in numerous perspectives, and finally, authors proved that gender bias feature words have a better description of different gender roles than gender bias personal appellations. Authors in [24] systematically compared the three most widely used DNNs- CNN, GRU, and LSTM- on a wide range of NLP tasks, including sentiment/relation classification, textual entailment, answer selection, Freebase question-relation matching, Freebase path query answering, and part of speech tagging. They discovered that for text classification tasks, CNNs and RNNs provide complimentary information and that which architecture performs better depends on how essential it is to interpret the entire sequence semantically. They also recommended that optimizing the hidden size and batch size is critical for excellent CNN and RNN performance. The authors also tested CNN and RNN for relation classification and found that CNN outperformed RNN. Both authors of [24] and [25] agreed that CNN should be preferred over GRU/ LSTM for lengthy sentence classification. In [26], authors introduced three neural network-based approaches for Twitter sentiment analysis. They evaluated three proposed methods based on different supervised classifiers: gated recurrent unit neural network (GRU), convolutional neural network (CNN), and support vector machine (SVM) classifier and demonstrated that GRU-based solution achieved the best macro-averaged F1-score and also has the best micro-averaged F1-score. Important similar works from the literature are summarized in Table 1.

**Table 1.** Summary of the literature review.

| Article | Origin of Names | Language | Processing Methods | Dataset and Results |
|---|---|---|---|---|
| Q. Shuai et al. [17] | Chinese | English | Naïve Bayes, support vector, decision tree, and maximum tree classifier-based models. | NLTK (Natural Language Toolkit) dataset with 7944 entries is used for English names. These models achieved around 72%, 73%, 71%, and 69% accuracy, respectively. |
| Y. Hu et al. [19] | USA | English | Character-based machine learning models such as LSTM and BERT. | A dataset of 21M unique first names from SSA and YAHOO data. LSTM and BERT-based different models obtained approximately 87% and 88% accuracy, respectively. |

**Table 1.** *Cont.*

| Article | Origin of Names | Language | Processing Methods | Dataset and Results |
|---|---|---|---|---|
| C. B. Rego et al. [20] | Brazilian | English | DNN-based models, including MLP, RNN, GRU, CNN, and Bi-LSTM. | The dataset consists of 100,787 Brazilian names, of which 54.82% are female names and 45.18% are male, based on 2010 CENSO data. CNN, Bi-LSTM, RNN, MLP, and GRU-based models achieved 92%, 95%, 93%, 86%, and 94% accuracy, respectively. |
| A. Panchenko et al. [21] | Russian | English | Statistical models based on one type of features: endings, character trigrams, and dictionary. | A dataset of 100,000 Russian full names from Facebook. Accuracy of up to 96% is obtained from a combined model: endings + 3-*g* + dictionary. |
| A. Tripathi et al. [22] | Indian | English | Support vector machine (SVM)-based different classifiers. | A dataset of 2000 Indian names with almost equal numbers of Gujarati, Punjabi, Bangla, Hindi, Urdu, Tamil, and Telugu names. The compiled training data contained 890 female and 1110 male names. Achieved maximum F1 score is 94.9%. |

## 3. Dataset Preparation

We prepared a dataset for NLP tasks containing 2030 data samples of Bengali names and corresponding gender, both for females and males. In our dataset, there are 1029 names of the male gender and 1001 names of the female gender. The data samples are collected from different sources such as Wikipedia, baby name suggestion websites, etc. Mainly we collected the names and corresponding gender of famous poets, singers, athletes, baby names, and other well-known persons in Bangladesh. Our dataset is publicly available in the Hugging Face (https://huggingface.co/datasets/faruk/bengali-names-vs-gender?fbclid=IwAR1lT_GGiWRquGAJ4tefNSUqQj8OuCeTCiaOY1t_79SbLdq8Pv94W4Fto7Y, accessed on 26 December 2022). Full documentation and dataset specifications are also provided in the repository. The dataset in CSV format has two columns, namely Name and Gender. We classified the data into two categories, which are Male and Female. Each row has two attributes. The first one is name, and the second one is gender. The name attribute is in utf-8 encoding, and the gender attribute has been signified by 0 and 1; 0 is used for male, and 1 is for female. We did not consider only diminutive names, all kinds of names that means short, medium, and large size names are available in our dataset containing one, two, and three string, respectively. We did not perform augmentation to our collection by multiplying the number of names. This whole dataset was prepared and cleaned manually for our work. Our prepared dataset is graphically depicted in Figure 1.

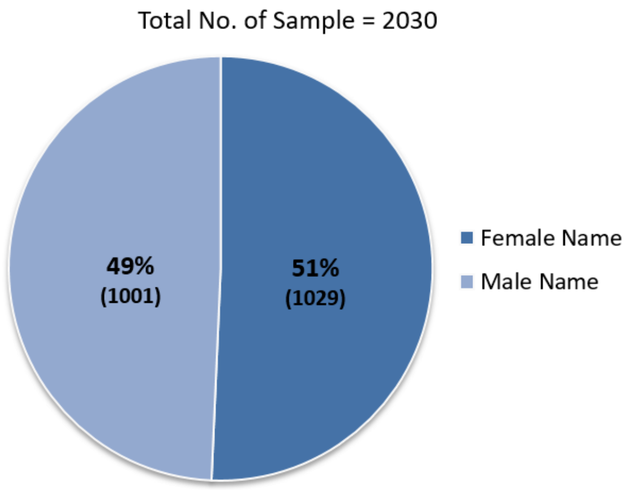


**Figure 1.** Dataset overview.

## 4. The Proposed Framework

The overall system architecture, including major components of the proposed method, is shown in Figure 2. The detail of each component is described in the following subsections:

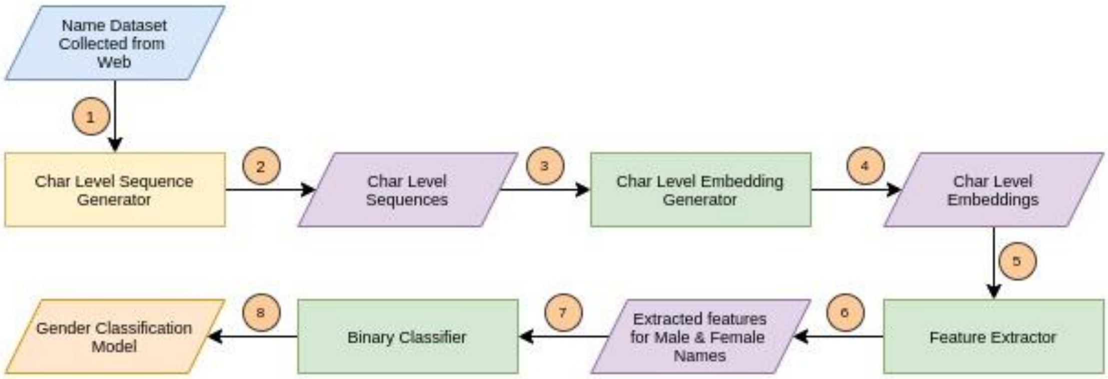

Name Dataset Collected from Web
1
Char Level Sequence Generator
2
Char Level Sequences
3
Char Level Embedding Generator
4
Char Level Embeddings
5
Feature Extractor
6
Extracted features for Male & Female Names
7
Binary Classifier
8
Gender Classification Model

**Figure 2.** Flow diagram of gender classification from Bengali names.

### 4.1. Char-Level Sequence Generator

We converted all the names in our dataset to char-level sequences. For generating sequences, we choose the maximum length of the sequences to be 29. If any sequence is less than the maximum length, we padded zero as the default value.

### 4.2. Char-Level Embedding

Embedding is one of the most important parts of working with text data. We used character-level embedding for the following two reasons: Firstly, the majority of the names in our dataset are short; that is, most of them contain only one string, which may be the person’s first name or last name. Secondly, occurrences of characters in a person’s name largely depend on the person’s gender. Normally, the suffix (‘আ’ = ‘a’), (‘ই’ = ‘i’), (‘উ’ = ‘u’) prevails at the end of a female name.

### 4.3. Feature Extractor and Binary Classifier

To learn the features from the embedding, we performed several experiments using a convolution neural network (CNN) as well as with long short-term memory (LSTM) and their variations. Table 1 shows the detailed configurations of those feature extractors. We used a binary classification layer with a sigmoid activation for our classification purpose.

## 5. DNN-Based Models

A brief introduction of CNN- and LSTM-based DNNs used in this research work is provided in this section.

### *5.1. Convolutional Neural Network*

Convolutional neural network (CNN/ConvNet) is a multiple-layer deep neural network that is commonly used in the design and calculation of natural language processing (NLP)-related tasks, including word segmentation (WS), information extraction (IE), relation extraction (RE), named entity recognition (NER), parts of speech tagging (POS), coreference resolution, parsing, word sense disambiguation (WSD), speech recognition, text to speech (TTS), machine translation (MT), etc. [27]. The idea behind using CNNs in NLP is to make use of their ability to extract features that capture salient information about an input. CNNs are applied to embedding vectors of a given sentence with the hopes that they will manage to extract useful features, such as phrases and relationships between words that are closer together in the sentence, which can be used for text classification. The NLP CNN is usually made up of 3 or more 1D convolutional and pooling layers, unlike traditional CNNs. In this work, we used one-dimensional CNN (Conv1D) and a combination of Conv1D and stacked LSTM-based models.

### *5.2. Long Short-Term Memory*

Long short-term memory (LSTM) networks are a kind of RNN that can learn long-term dependencies. It is essentially their default behavior to remember information for longer durations. LSTM holds promise for any sequential processing tasks, including language modeling, machine translation, handwriting recognition and generation, speech recognition, speech synthesis, etc. [28,29]. Several models based on LSTM, Bi-LSTM, and stacked LSTM are used in this paper.

### *5.3. Loss Function*

The purpose of loss functions is to compute the quantity that a model should seek to minimize during training. Since there are only two label classes (assumed to be 0 and 1) in this work, we used the binary cross-entropy loss function, which is a kind of probabilistic loss function that computes the cross-entropy loss between true labels and predicted labels. For each example, there should be a single floating-point value per prediction. The binary cross-entropy loss function calculates the loss (L) of an example by computing the following average:

$$\mathrm{Loss} = -\frac{1}{\mathrm{output\ size}} \sum_{\mathrm{i}=1}^{\mathrm{output\ size}} \mathrm{y_i} \cdot \log \hat{\mathrm{y}}_\mathrm{i} + (1 - \mathrm{y_i}) \cdot \log(1 - \hat{\mathrm{y}}_\mathrm{i}) \tag{1}$$

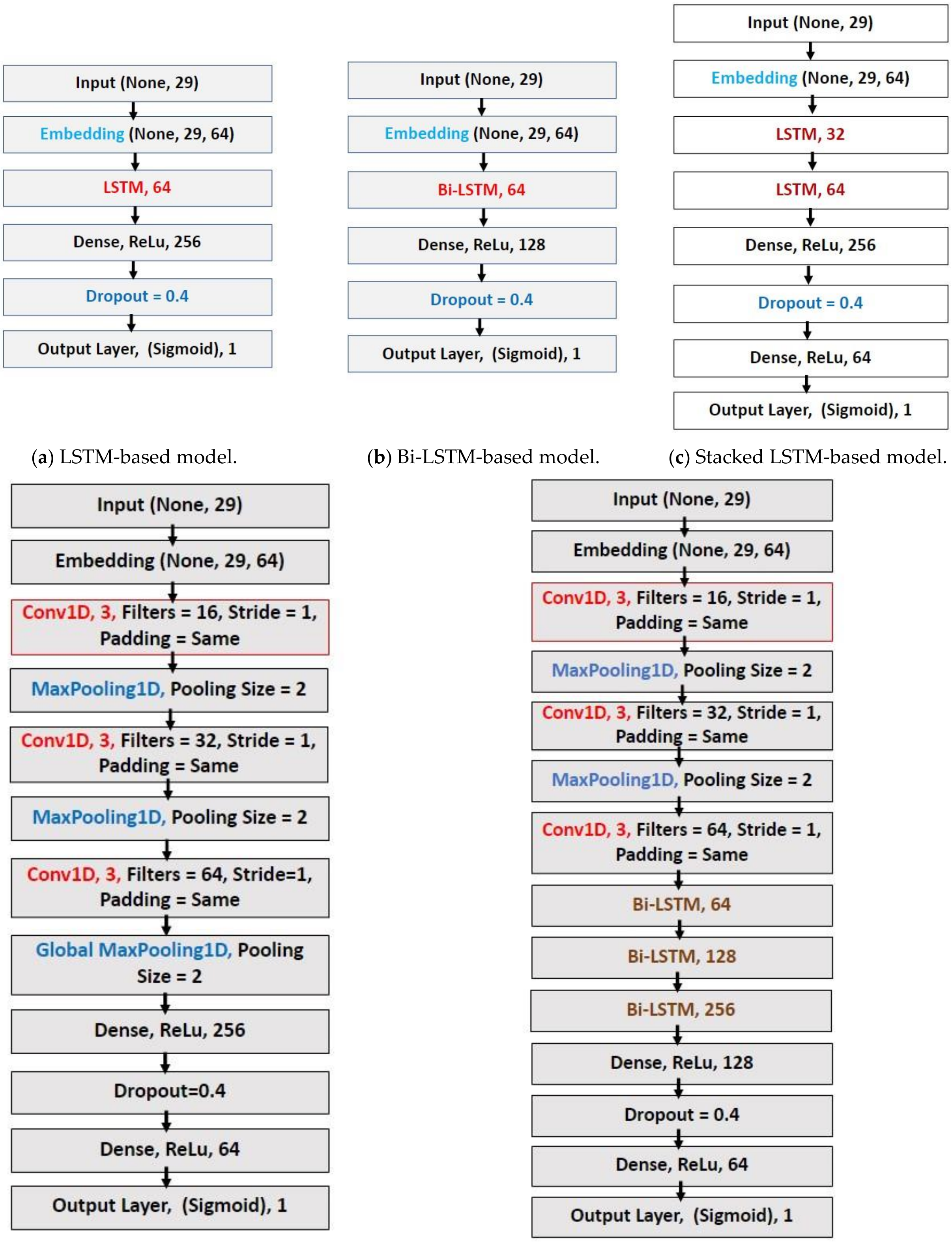
where $\hat{y}_i$ is the $i^{th}$ scalar value in the model output, $y_i$ is the corresponding target value, and outputsize is the number of scalar values in the model output [30].

### *5.4. Optimizer*

The Adam optimizer, which implements the Adam algorithm, was utilized in this research. Adam optimization is a stochastic gradient descent approach based on adaptive first- and second-order moment estimation. This technique is computationally efficient, requires minimal memory, is insensitive to diagonal rescaling of gradients, and is well suited for large data/parameter issues [31].

### *5.5. Model Configurations*

Five distinct deep neural networks based on LSTM, Bi-LSTM, and Conv1D were employed in total. Table 2 lists the technical specifications for each model. In addition, from Figure 3a through Figure 3e, each model's layer visualization with data dimension is depicted.


Input (None, 29)
Embedding (None, 29, 64)
LSTM, 64
Dense, ReLu, 256
Dropout = 0.4
Output Layer, (Sigmoid), 1
Input (None, 29)
Embedding (None, 29, 64)
Bi-LSTM, 64
Dense, ReLu, 128
Dropout = 0.4
Output Layer, (Sigmoid), 1
Input (None, 29)
Embedding (None, 29, 64)
LSTM, 32
LSTM, 64
Dense, ReLu, 256
Dropout = 0.4
Dense, ReLu, 64
Output Layer, (Sigmoid), 1
Input (None, 29)
Embedding (None, 29, 64)
Conv1D, 3, Filters = 16, Stride = 1, Padding = Same
MaxPooling1D, Pooling Size = 2
Conv1D, 3, Filters = 32, Stride = 1, Padding = Same
MaxPooling1D, Pooling Size = 2
Conv1D, 3, Filters = 64, Stride=1, Padding = Same
Global MaxPooling1D, Pooling Size = 2
Dense, ReLu, 256
Dropout=0.4
Dense, ReLu, 64
Output Layer, (Sigmoid), 1
Input (None, 29)
Embedding (None, 29, 64)
Conv1D, 3, Filters = 16, Stride = 1, Padding = Same
MaxPooling1D, Pooling Size = 2
Conv1D, 3, Filters = 32, Stride = 1, Padding = Same
MaxPooling1D, Pooling Size = 2
Conv1D, 3, Filters = 64, Stride = 1, Padding = Same
Bi-LSTM, 64
Bi-LSTM, 128
Bi-LSTM, 256
Dense, ReLu, 128
Dropout = 0.4
Dense, ReLu, 64
Output Layer, (Sigmoid), 1

(**a**) LSTM-based model.

(**b**) Bi-LSTM-based model.

(**c**) Stacked LSTM-based model.

(**d**) Conv1D-based model.

(**e**) Combined Conv1D and stacked Bi-LSTM-based model.

**Figure 3.** (**a**–**e**) Layer visualization of five different DNN models.

**Table 2.** Experimental setting of different DNN models.

| Model No. | Model Name | Embedding Layer | Conv1D Layer | MaxPooling1D Layer | LSTM Layer | Bi-LSTM Layer | Fully Connected Layer | Dropout Layer | Classification Layer |
|---|---|---|---|---|---|---|---|---|---|
| Model 1 | LSTM Based Model | Dimension: 64 | N/A | N/A | Layer: 1<br>Unit: 64 | N/A | Layer: 1<br>Unit: 256 | Layer: 1<br>(40%) | Unit: 1<br>(Sigmoid) |
| Model 2 | Bi-LSTM Based Model | Dimension: 64 | N/A | N/A | N/A | Layer: 1<br>Unit: 64 | Layer: 1<br>Unit: 128 | Layer: 1<br>(40%) | Unit: 1<br>(Sigmoid) |
| Model 3 | Stacked LSTM Based Model | Dimension: 64 | N/A | N/A | Layer: 2<br>Unit: 32 (L1), 64 (L2) | N/A | Layer: 1<br>Unit: 256 | Layer: 1<br>(40%) | Unit: 1<br>(Sigmoid) |
| Model 4 | Conv1D Based Model | Dimension: 64 | Layer: 3<br>Filter: 16 (L1), 32 (L2), 64 (L3)<br>Filter Size: 3 | Layer: 2<br>Pooling Size: 2 | N/A | N/A | Layer: 1<br>Unit: 256 | Layer: 1<br>(40%) | Unit: 1<br>(Sigmoid) |
| Model 5 | Conv1D & Stacked Bi-LSTM Based Model | Dimension: 64 | Layer: 3<br>Filter: 16 (L1), 32 (L2), 64 (L3)<br>Filter Size: 3 | Layer: 2<br>Pooling Size: 2 | N/A | Layer: 3<br>Unit: 64 (L1), 128 (L2), 256 (L3) | Layer: 1<br>Unit: 128 | Layer: 1<br>(40%) | Unit: 1<br>(Sigmoid) |

## 6. Experimental Setup

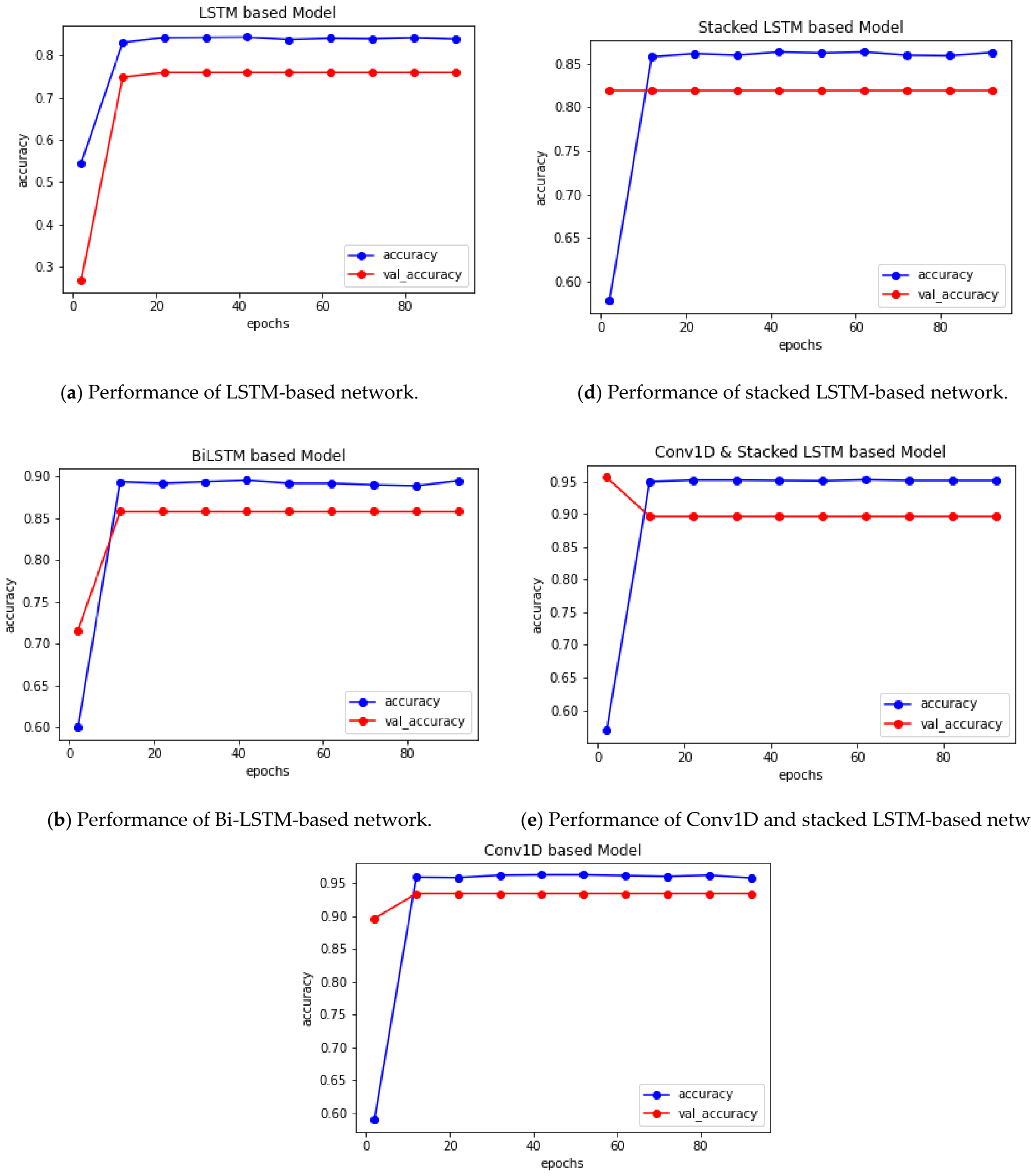
We have tried different deep learning-based algorithms to investigate the adaptability and improvements for various approaches. Five different classifiers have been trained and tested on the test set for comparing and selecting the suitable classifier as the best one for classifying gender form name data. We have executed the whole experiment in python 3.8. For training the deep learning models/classifiers, we have used keras with tensorflow (version 2.4.1) backend. We have used it to train, validate, and infer the predictions for test data to calculate the performance metrics. We have also used pandas (version 1.3.5) for handling, and preprocessing the dataset, matplotlib for visualizing the dataset, scikitlearn has been used for calculating the performance metrics. In order to evaluate the performance of our proposed approach, we conducted five different experiments using the dataset with total data point 2030, shown in Table 3. In every algorithm, our data train test split was 80/20; that is, 80% of the total data was used for training, and 20% was for testing. We have followed the 10-fold cross-validation method to evaluate how well our model can work. There is no overlap between the train and test datasets.

**Table 3.** Splitting of dataset for training, validation, and testing.

| Data Type | Data Point |
|---|---|
| Training | 1643 |
| Validation | 183 |
| Test | 204 |

## 7. Prediction and Performance Analysis

Using the training dataset, we built the experimental models. We tuned the essential parameters while constructing the models to make them more accurate and reliable. Once the models were completed, we ran our test dataset to assess the inferring methodology for gender prediction from Bangla names. We have achieved a satisfactory performance in each experiment in terms of accuracy, precision, recall, F1-score, ROC AUC score, and loss performance metrics, which is clearly demonstrated in Table 4. Table 5 depicts the percentage accuracy values for training, validation, and test dataset. The percentage accuracy of trained networks is also graphically illustrated in Figure 4a–e, and the comparison of accuracy values for training, validation, and test dataset are clearly depicted in Figure 5. From both tabular and pictorial representation, it is definitely depicted that Conv1D-based DNN provides maximum accuracy of 95.65%, 93.44%, and 92.16% for training, validation, and test dataset, respectively. It is clearly shown from Table 3 that Conv1D and the combination of Conv1D and stacked Bi-LSTM-based DNNs are more effective, whereas simple LSTM-based DNNs had the worst performance among the five variations of CNN and LSTM-based models.



(**a**) Performance of LSTM-based network.

(**d**) Performance of stacked LSTM-based network.

(**b**) Performance of Bi-LSTM-based network.

(**e**) Performance of Conv1D and stacked LSTM-based network.

(**c**) Performance of Conv1D-based network.

**Figure 4.** (**a**–**e**) Graphical illustration of epochs versus accuracy of each network for training and validation dataset.

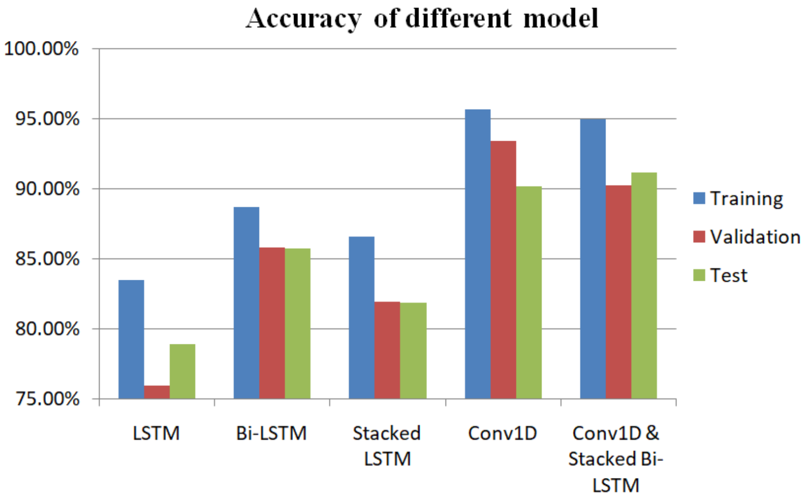


**Figure 5.** Performance comparison among the different models in terms of accuracy for training, validation, and test dataset.

**Table 4.** The performance of gender detection from Bangla name data.

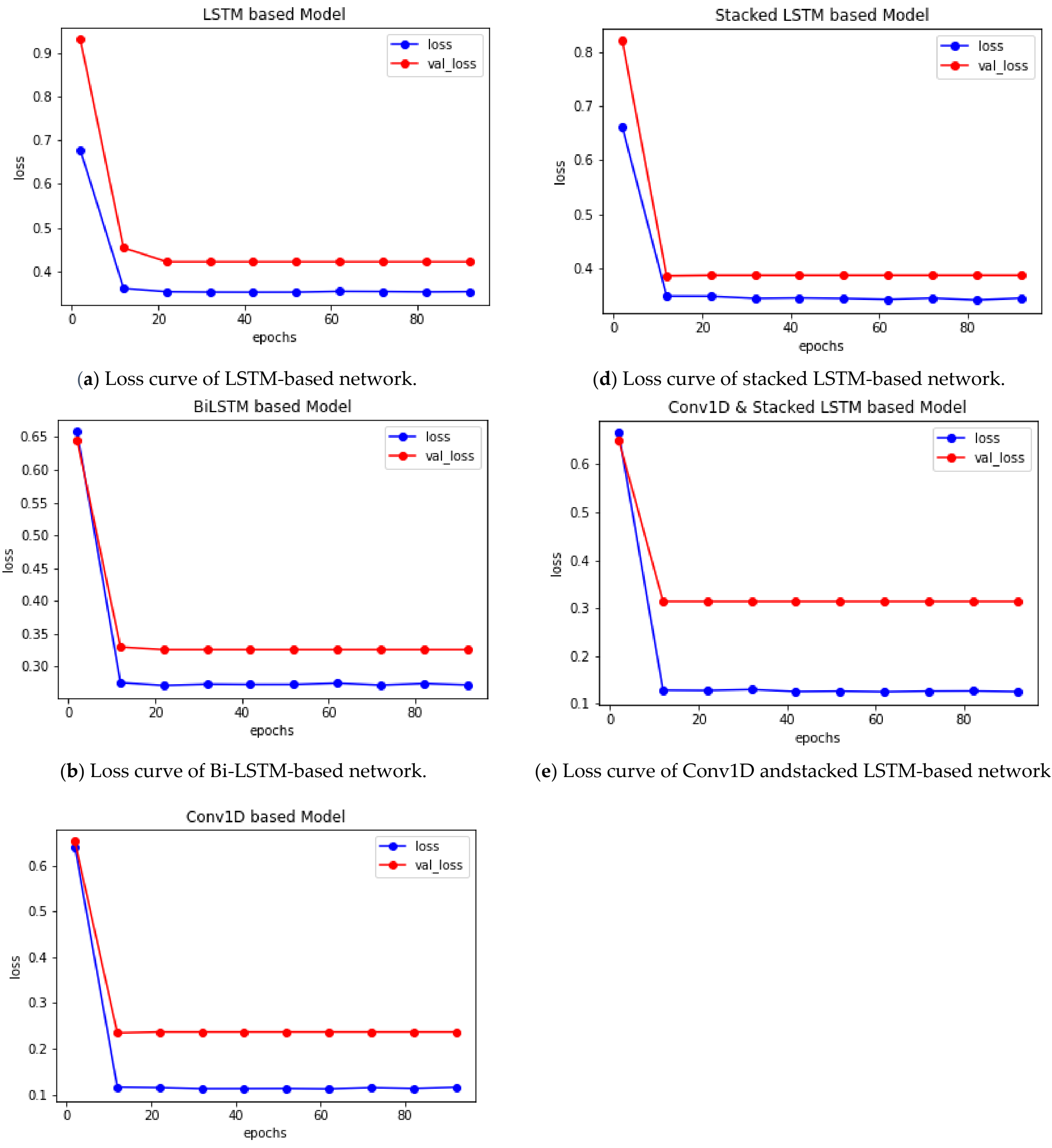
| Model Name | Accuracy | Precision | Recall | F1-Score | ROC AUC Score |
|---|---|---|---|---|---|
| LSTM-Based Model | 0.715686 | 0.638712 | 0.980198 | 0.773438 | 0.718254 |
| Bi-LSTM-Based Model | 0.872549 | 0.878788 | 0.861386 | 0.870012 | 0.872441 |
| Stacked LSTM-Based Model | 0.813725 | 0.811881 | 0.811881 | 0.811881 | 0.813708 |
| Conv1D-Based Model | 0.921569 | 0.929293 | 0.910891 | 0.920021 | 0.921465 |
| Conv1D and Stacked Bi-LSTM-Based Model | 0.921569 | 0.938144 | 0.900991 | 0.919192 | 0.921369 |

**Table 5.** Accuracy of five different classifiers for training, validation, and test dataset.

| Gender Classification Model Name | Prediction Accuracy | | |
|---|---|---|---|
| | Training | Validation | Test |
| LSTM-Based Model | 83.51% | 75.96% | 71.57% |
| Bi-LSTM-Based Model | 88.68% | 85.79% | 87.25% |
| Stacked LSTM-Based Model | 86.62% | 81.97% | 81.37% |
| Conv1D-Based Model | 95.65% | 93.44% | 92.16% |
| Conv1D and Stacked Bi-LSTM-Based Model | 94.96% | 90.26% | 92.16% |

## 8. Experimental Results and Discussion

It is a very difficult and critical issue to extract information from Bangla names. In this work, five deep learning classifiers, such as LSTM, Bi-LSTM, stacked LSTM, and Conv1D and stacked Bi-LSTM, have been applied using our own dataset. To evaluate the effectiveness of our developed classifier models perfectly, we examined several performance metrics: accuracy, precision, recall, F1-score, and ROC AUC score. The tabular presentation of the performance metrics' values of the five proposed systems, along with the comparison, are neatly shown in Tables 4 and 5. Table 6 represents the summary of the experimental results. It is clearly depicted from Table 4 that the ConvNet-based model outperforms all in terms of accuracy, precision, recall, F1-score, and ROC AUC score performance metrics. From Table 5, it is also shown that the prediction accuracy of the Conv1D-based model is high for training, validation, and test dataset. The CNN-based system achieved the highest accuracy, 92.16%, and foremost, an F1 score of 0.920, which is clearly presented in Table 6. On the contrary, the LSTM-based system provides minimal performance with an accuracy of 71.57% and an F1 score value of 0.773 only. In addition, the graphical representation of the accuracy and loss values of the five different models are evidently displayed in Figures 4–6. From the experimental results, we can conclude that the Conv1D-based classifier is very much robust and gives the best output among all of these. The implementation and trained models of this research are publicly available at the GitHub (https://github.com/MdHumaunKabir/Gender-Prediction-from-Bangla-Names, accessed on 26 December 2022).



(**a**) Loss curve of LSTM-based network.

(**d**) Loss curve of stacked LSTM-based network.

(**b**) Loss curve of Bi-LSTM-based network.

(**e**) Loss curve of Conv1D andstacked LSTM-based network.

(**c**) Loss curve of Conv1D-based network.

**Figure 6.** (**a**–**e**) Pictorial representation of epochs versus the loss of different networks for training and validation dataset.

**Table 6.** Comparison among the experimental results.

| Best Achieved Accuracy | Best Performing Classifier | Second Best Performing Classifier | Least Accuracy | Least Performing Classifier | Best F1 Score |
|---|---|---|---|---|---|
| 92.16% | Conv1D-Based Model | Conv1D and Stacked Bi-LSTM-Based Model | 71.57% | LSTM-Based Model | 0.920 |

In Section 2, we presented some related works. From this section, it is shown that there is no study available in the literature that classifies gender from Bangladeshi names in the Bengali language. This is the first research to infer gender from Bangladeshi names in the Bengali language using our own dataset. For this reason, it is difficult to compare our achieved results with previous works. We looked into the challenge of determining gender from a user's first name in this study. We suggested character-based machine learning models and proved that our models are able to infer the gender of users with much higher accuracy. In different cultures, only initial names may have diverse gender connotations. In this case, we illustrated that using both first and last names may be effective. One drawback of our system is that if the first name is unisex, determining gender might be difficult. It is beneficial to utilize both first and last names, or the first name and content information, in this circumstance. In the future, we intend to investigate whether integrating these entire factors (first and last names, as well as content information) can provide us even better outcomes.

## 9. Conclusions and Future Work

Automatic gender detection can be beneficial for gender analysis in any sector/ service to discover gender bias or inequality, specifically to understand women's position in tech, media, professional domain, academic domain, etc. Additionally, it will be a smart way to predict the gender from the name, which will be helpful for gender-based data analysis using text data. If the dataset is increased, we hope to get better performance by using Conv1D and a combination of Conv1D and stacked Bi-LSTM. Other methods can also be tried for better results. In the near future, we will work to develop an online-based name-to-gender inference service API that will be available on the Internet for Bengali and English names. Other applications and users can utilize the API. We will also aim to collect additional data and employ other innovative algorithms and technologies to strengthen the models.

**Author Contributions:** Conceptualization, M.H.K., F.A. and J.S.; methodology, M.H.K., F.A. and J.S.; investigation, M.H.K., F.A. and J.S.; data curation, M.H.K. and F.A.; writing—original draft preparation, M.H.K.; writing—review and editing, M.H.K., J.S. and M.A.M.H.; visualization, M.H.K., F.A. and M.A.M.H.; supervision, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Competitive Research Fund of The University of Aizu, Japan.

**Institutional Review Board Statement:** Not Applicable.

**Informed Consent Statement:** Not Applicable.

**Data Availability Statement:** Our prepared own dataset with full documentation, called the Bengali female and male names dataset, is publicly available at the Hugging Face (https://huggingface.co/datasets/faruk/bengali-names-vs-gender?fbclid=IwAR1lT_GGiWRquGAJ4tefNSUqQj8OuCeTCiaOY1t_79SbLdq8Pv94W4Fto7Y, accessed on 26 December 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Iman Ghosh. Ranked: The 100 Most Spoken Languages around the World. *Last Modified 15 February 2020.* Available online: https://www.visualcapitalist.com/100-most-spoken-languages/ (accessed on 8 January 2021).
2. Sen, O.; Fuad, M.; Islam, M.N.; Rabbi, J.; Masud, M.; Hasan, M.K.; Awal, M.A.; Fime, A.A.; Fuad, M.T.H.; Sikder, D.; et al. Bangla Natural Language Processing: A Comprehensive Analysis of Classical, Machine Learning, and Deep Learning-Based Methods. *IEEE Access* **2022**, *10*, 38999–39044. [CrossRef]
3. Patel, S.; Shah, B.; Kaur, P. Leveraging User Comments in Tweets for Rumor Detection. In *Advances in Intelligent Systems and Computing*; Springer: Singapore, 2022; Volume 1388. [CrossRef]
4. Bhowmik, N.; Arifuzzaman, M.; Mondal, M. *Sentiment Analysis on Bangla Text Using Extended Lexicon Dictionary and Deep Learning Algorithms*; Elsevier Inc.: Amsterdam, The Netherlands, 2022; Volume 13, pp. 1–14.
5. Ani, J.F.; Islam, M.; Ria, N.J.; Akter, S.; Masum AK, M. Estimating Gender Based On Bengali Conventional Full Name With Various Machine Learning Techniques. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021; pp. 1–6. [CrossRef]
6. Abdallah, E.E.; Alzghoul, J.R.; Alzghool, M. Age and Gender prediction in Open Domain Text. *Procedia Comput. Sci.* **2020**, *170*, 563–570. [CrossRef]
7. Karako, C.; Manggala, P. Using image fairness representations in diversity-based re-ranking for recommendations. In Proceedings of the Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, Singapore, 8–11 July 2018; pp. 23–28.
8. Yao, S.; Huang, B. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems*; MIT press: Cambridge, MA, USA, 2017; pp. 2921–2930.
9. Gattal, A.; Djeddi, C.; Bensefia, A.; Ennaji, A. Handwriting Based Gender Classification Using COLD and Hinge Features. In *Lecture Notes in Computer Science Image and Signal Processing*; Springer: Cham, Switzerland, 2020; pp. 233–242.
10. Roy, P.; Bhagath, P.; Das, P. Gender Detection from Human Voice Using Tensor Analysis. In Proceedings of the 1st Joint SLTU and CCURL Conference on Language Resources and Evaluation (LREC), Marseille, France, 16 May 2020; pp. 211–217.
11. Bérubé, N.; Ghiasi, G.; Sainte-Marie, M.; Larivière, V. Wiki-Gendersort: Automatic gender detection using first names in Wikipedia. *arXiv* **2020**. [CrossRef]
12. To, H.Q.; Nguyen, K.V.; Nguyen, N.L.; Nguyen, A.G. Gender Prediction Based on Vietnamese Names with Machine Learning Techniques. In Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval, Seoul, Republic of Korea, 18–20 December 2020.
13. Sotelo, A.F.; Gómez-Adorno, H.; Esquivel-Flores, O.; Bel-Enguix, G. Gender Identification in Social Media Using Transfer Learning. In *Lecture Notes in Computer Science Pattern Recognition*; Springer: Cham, Switzerland, 2020; pp. 293–303.
14. Kowsher, M.; Sanjid, M.Z.I.; Das, A.; Ahmed, M.; Sarker, M.M.H. Machine Learning and Deep Learning based Information Extraction from Bangla Names. *Procedia Comput. Sci.* **2020**, *178*, 224–233. [CrossRef]
15. Karim, R.; Islam, M.A.; Simanto, S.R.; Chowdhury, S.A.; Roy, K.; Al Neon, A.; Hasan, M.S.; Firoze, A.; Rahman, R.M. A Step Towards Information Extraction: Named Entity Recognition in Bangla Using Deep Learning. *J. Intell. Fuzzy Syst.* **2019**, *37*, 1–13. [CrossRef]
16. Chollet, F. *Deep Learning with Python*; Manning Publications: New York, NY, USA, 2017.
17. Shuai, Q.; Wang, R.; Jin, L.; Pang, L. Research on Gender Recognition of Names Based on Machine Learning Algorithm. In Proceedings of the 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 25–26 August 2018; Volume 2, pp. 335–338.
18. Zhang, S.; Zhang, S.; Wang, X. Automatic Recognition of Chinese Organization Name Based on Conditional Random Fields. In Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering, Beijing, China, 30 August–1 September 2007; pp. 229–233.
19. Hu, Y.; Hu, C.; Tran, T.; Kasturi, T.; Joseph, E.; Gillingham, M. What's in a name?—Gender classification of names with character based machine learning models. *Data Min. Knowl. Discov.* **2021**, *35*, 1537–1563. [CrossRef]
20. Rego, R.C.; Silva, V.M. Predicting gender of Brazilian names using deep learning. *arXiv* **2021**, arXiv:2106.10156.
21. Panchenko, A.; Teterin, A. Detecting gender by full name: Experiments with the russian language. In *International Conference on Analysis of Images, Social Networks and Texts*; Springer: Cham, Switzerland, 2014; pp. 169–182.
22. Tripathi, A.; Faruqui, M. Gender prediction of Indian names. In Proceedings of the IEEE Technology Students' Symposium, Kharagpur, India, 14–16 January 2011; pp. 137–141. [CrossRef]
23. Tang, Q.; Lin, H. Research on Gender Recognition for Character in Text. *J. Chin. Inf. Process.* **2010**. Available online: https://en.cnki.com.cn/Article_en/CJFDTotal-MESS201002005.htm (accessed on 18 January 2020).
24. Paiva, E.; Paim, A.; Ebecken, N. Convolutional Neural Networks and Long Short-Term Memory Networks for Textual Classification of Information Access Requests. *IEEE Lat. Am. Trans.* **2021**, *19*, 826–833. [CrossRef]
25. Han, X.; Li, B.; Wang, Z. An attention-based neural framework for uncertainty identification on social media texts. *Tsinghua Sci. Technol.* **2020**, *25*, 117–126. [CrossRef]
26. Arkhipenko, K.; Kozlov, I.; Trofimovich, J.; Skorniakov, K.; Gomzin, A.; Turdakov, D. Comparison of neural network architectures for sentiment analysis of Russian tweets. *Proc. Dialogue* **2016**. Available online: http://www.dialog-21.ru/media/3380/arkhipenkoetal.pdf (accessed on 20 January 2020).

27. Wang, W.; Gang, J. Application of Convolutional Neural Network in Natural Language Processing. In Proceedings of the International Conference on Information Systems and Computer Aided Education (ICISCAE), Changchun, China, 6–8 July 2018; pp. 64–70.
28. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *J. Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
29. Yao, L.; Guan, Y. An Improved LSTM Structure for Natural Language Processing. In Proceedings of the IEEE International Conference of Safety Produce Informatization (IICSPI), Chongqing, China, 10–12 December 2018; pp. 565–569. [CrossRef]
30. Keras API Docs. Binary Cross-Entropy Class. Available online: https://keras.io/api/losses/probabilistic_losses/#binarycrossentropy-class (accessed on 2 January 2021).
31. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2015**, arXiv:1412.6980.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.