This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

Digital Object Identifier

OPCNN-FAKE: Optimized Convolutional Neural Network for Fake News Detection

HAGER SALEH[1], ABDULLAH ALHARBI[2], and SAEED HAMOOD ALSAMHI[3]
[1]Faculty of Computers and Artificial Intelligence,South Valley University, Hurghada,Egypt
[2]Department of Information Technology, College of Computers and Information Technology, Taif University, P.O.Box 11099, Taif 21944, Saudi Arabia
[3]Athlone Institute of Technology, Athlone, Ireland,& IBB University, Ibb, Yemen

Corresponding author: First A. Author (e-mail: hager.saleh.fci@gmail.com

ABSTRACT Recently, there is a rapid and wide increase in fake news, defined as provably incorrect information spread with the goal of fraud. The spread of this type of misinformation is a severe danger to social cohesiveness and well-being since it increases political polarisation and people's distrust of their leaders. Thus, fake news is a phenomenon that is having a significant impact on our social lives, particularly in politics. This paper proposes novel approaches based on Machine Learning (ML) and Deep Learning (DL) for the fake news detection system to address this phenomenon. The main aim of this paper is to find the optimal model that obtains high performance. Therefore, we propose an optimized Convolutional Neural Network model to detect fake news (OPCNN-FAKE). We compare the performance of the OPCNN-FAKE with Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and The six regular ML techniques: Decision Tree (DT), logistic Regression (LR), K Nearest Neighbor (KNN), Random Forest (RF), Support Vector Machine (SVM), and Naive Bayes (NB) using four fake news benchmark datasets. Grid search and hyperopt optimization techniques have been used to optimize the parameters of ML and DL, respectively. In addition, N-gram and Term Frequency—Inverse Document Frequency (TF-IDF) have been used to extract features from the benchmark datasets for regular ML, while Glove word embedding has been used to represent features as a feature matrix for DL models. To evaluate the performance of the OPCNN-FAKE, accuracy, precision, recall, F1-measure were applied to validate the results. The results show that OPCNN-FAKE model has achieved the best performance for each dataset compared with other models. Furthermore, the OPCNN-FAKE has a higher performance of cross-validation results and testing results than the other models, which indicates that the OPCNN-FAKE for fake news detection is significantly better than other models.

INDEX TERMS Fake news, Machine learning, Deep learning, Neural network, convolutional neural network, detection, OPCNN-Fake

I. INTRODUCTION

In recent years, the ability of a user to write anything on online news platforms such as social media and news websites newspapers leads to the propagation of misleading information [1]. Online social media platforms (Twitter, Facebook, Instagram, YouTube, etc.) have become the primary source of news for people around the world, particularly in developing nations. Therefore, anyone from anywhere in the world can use popular social media and social networking as platforms to publish any statement and spread fake news through various networking sites to achieve various goals, which may be illegitimate. There are currently experiencing significant ramifications for society, business, and culture as a result of the increasing use of social media, which have the potential to be both detrimental and beneficial [2].

Fake news is widely regarded as one of the most severe dangers to global commerce, journalism, and democracy, with significant collateral harm. The stock market suffered a $130 billion loss as a result of a false news story that US President Barack Obama had been injured in an explosion [3]. According to statistics published by Stanford University academics, 72.3 percent of fake news originates from official news outlets and online social media platforms [4]. Because of the negative impact of fake news on society, it is critical to building effective fake news detection systems. As a result, fake news is widely regarded as one of the most serious challenges to global commerce, media, and democracy,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

posing significant societal harm.

With Artificial Intelligence (AI) rapid progress, a significant number of experiments are being undertaken to tackle issues that were never addressed in the framework of computer science, such as fake news detection [5]–[8]. Automatic detection approaches based on Machine Learning (ML) have been studied to combat the emergence and dissemination of false news. The majority of fake news detection systems utilize ML approaches to help consumers in filtering the content they are seeing and determining if a given news piece is misleading or not [5], [9]. Deep Learning (DL) techniques recent accomplishments in difficult natural language processing tasks make them viable for detecting fake news effectively and efficiently. Creating automatic, trustworthy, and accurate systems for identifying fake news on social media is a hot topic of research. The process of determining if a certain news item on any field, from any social media domain, is purposefully or inadvertently misleading might be characterized as fake news detection [6]–[8]. Convolutional Neural Network (CNN) has been prominent in many fields with the best performance, including computer vision [10], smart building structures [11], and natural language processing [12]. CNN uses convolution layers, pooling layers, and fully connected layers to extract more features wih high-level and low-level features. Therefore, we proposed an Optimal CNN model for Fake news detection (OPCNN-Fake) that can extract high-level and low-level features from the dataset to detect fake news, and it has registered the best performance compared with others models.

A. MOTIVATION AND CONTRIBUTION

Fake news is an effect in journalism, global commerce, and democracy, with significant collateral harm. Fake news detection is an area of artificial intelligence that has attracted the curiosity of researchers from all over the world. Unfortunately, regular ML techniques have not provided significant performance for the detection of fake news. However, DL is more efficient for extracting features of fake news detection than Regular ML due to its capability to deep extraction of high and low levels. In this paper, we propose an efficient OPCNN-FAKE model based on optimized CNN for detecting fake es. Furthermore, CNN can extract more features by using different layers. The contributions of this study are as follows:

We propose the OPCNN-Fake model for detecting fake news, which uses various layers to extract high-level and low-level features. Also, we optimize OPCNN-FAKE by selecting the best values of OPCNN-FAKE's parameters in each layer using the hyperopt optimization technique. In addition, we utilize four benchmark datasets, divide each one into a 20% testing dataset and an 80% training datset.We evaluate the performance of the OPCNN-FAKE model based on accuracy, precision, recall, F1-measure. Furthermore, we compare the performance of OPCNN-FAKE with different models, DT, RF, SVM, NB, LR, KNN, RNN, and LSTM. Then, registering results for cross-validation (training set) and testing (unseen data). The experimental results demonstrate the effectiveness of the OPCNN-FAKE significant performance compared to other models. We show that our OPCNN-FAKE can efficiently effectively and efficiently detect fake news with high level of accuracy.

B. PAPER ORGANIZATION

The remainder of the paper is organized as follows: Section II reviews related works on fake news detection. The proposed OPCNN-FAKE model in this paper is presented in Section III. The experimental results, as well as a comparison to the baseline categorization and discussion, are presented in Section IV. Finally, Section V provides a summary of the paper.

II. RELATED WORK

This section covers a variety of machine learning algorithms for detecting fake news. Jing [13] have proposed model-based to build hidden representations that capture changes in contextual information in relevant posts over time. They applied experimental using 5 million postings that were collected from Twitter and Sina Weibo microblogs. They made a comparison between DT, RF, SVM, LSTM and Gated Recurrent Unit (GRU), and RNN. On the same dataset, another study developed a hybrid DL model. Ruchansky et al., [14] proposed model Capture, Score, and Integrate (CSI) includes three modules: Capture, Score, and Integrate. The capture module has used LSTM and RNN to extract from particular article mundane patterns of user activity. The first module uses.

Score module has used a fully connected neural network layer to capture characteristics from users' behavior. Both models have integrated with the third model to classify articles that is fake or not. Shu et al. [15] released the FakeNewsNet dataset and applied different algorithms to a dataset: SVM, LR, NB, and CNN. Salem, et al. [16] used the FA-KES dataset comprises news events around the Syrian war. There are 804 news articles in the collection, 376 of which are fraudulent. Semi-supervised with a fact-checking labeling approach are used to annotations dataset. However, the dataset can be used to train machine learning models for detecting fake news. Popat et al. [17] introduced DeClarE, an end-to-end neural network model for debunking fake news and fraudulent claims. To support or reject a claim, it uses evidence and counter-evidence gathered from the internet. The authors trained a bi-directional LSTM model with at least four different datasets and achieved an overall classification accuracy of 80%. Ksieniewicz et al. [18] proposed decision tree ensembles diversified using the Random Subspace method to detect fake news.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

Singh, et al. [19] proposed an Attention-based LSTM network that uses tweet text with thirteen different linguistic and user features to distinguish rumor and non-rumor tweets. They compared the Attention-based LSTM network with various conventional machines and DL models. The results show that Attention-based LSTM network has achieved the best performance. Ahmed et al. [20] proposed the ISOT dataset, made compassion between six machine learning models using n-gram with two feature extraction techniques: Term Frequency (TF) and Term Frequency — Inverse Document Frequency (TF-IDF) to the ISOT dataset. Pérez-Rosas, et al. [21] also developed classification models using linguistic features such as lexical, syntactic, and semantic level features and a linear SVM to detect fake and real news. CNN have been utilised in a variety of computer vision in recent years, and they have improved the state-of-the-art performance of a variety of visual classification tasks, such as image processing [22], face verification [23], object recognition [24], and natural language processing tasks [25].

Yang, Yang, et al. [26] proposed a model using Text and Image information based CNN (TI-CNN). They compared their model with several models such as LSTM, CNN, and GRU using two datasets. Abdullah, A., et al. [27] used CNN and LSTM to classify the fake news articles that achieved significant performance. They made experimental using one Fake news dataset from Kaggle.

To detect fake news, the authors of [28] proposed a Deep Convolutional Neural Network (FNDNet) to learn the discriminatory features for fake news detection. Furthermore, the authors of [29] introduced a hybrid deep learning model that blends CNN and RNN. To detect fake news articles, the authors of [27] presented CNN and LSTM to categorize fake news to produce significant results. Also, the authors of [30] developed multi-level CNN, which incorporated local and global convolutional features to collect semantic information from article texts efficiently. However, the content of the news story and the presence of echo chambers in the social network are considered [31]. While the authors of [32] focused on the substance news piece and the presence of echo chambers in the social network. Table .1 summaries the comparison of existing works and our proposed work.

III. FAKE NEWS DETECTION SYSTEM

Figure 1 presents the main steps of the proposed system. It consists many steps: fake news data collection, text pre-processing, dataset splitting, features extraction methods, training/optimization models, and evaluating models. There are two approaches in the proposed system: the regular ML approach and the DL approach. In the ML approach, six ML models: DT, LR, KNN, RF, SVM, and NB are used to train and evaluate. Different sizes of n-gram, including uni-gram, bi-gram, tri-gram, and four-gram with TF-ID feature extraction method, extract features and build matrix features. Grid search with cross-validation is used to optimize the ML models. In the DL approach, the OPCNN-FAKE model is proposed and it, LSTM, RNN are used to train and evaluate. The hyperopt optimization method are used to optimized OPCNN-FAKE, RNN and LSTM. Word embedding are used for feature extraction. Also, We compared the OPCNN-FAKE model with RNN and LSTM. Word embedding is used to build a feature matrix. Each step is describing the details as following

A. FAKE NEWS DATASET

We trained, optimized, and evaluated models using four datasets. Each dataset was split into 80% training dataset and 20% testing dataset (unseen data). In this section, these datasets are introduced as following.

1) Dataset1

Fake News detection was collected from Kaggle [34]. There are 3988 news articles in this dataset. In addition to the body of the text, each article includes a headline and a list of URLs. There is also a class label with the values "0" for fake news and "1" for real news. Only the article body and headline can be used in models. The 1868 articles are real news, while the remaining 2120 are fake news. The statistics of the training set and testing set for dataset1 are shown in Table 2.

2) FakeNewsNet (dataset2)

FakeNewsNet [33] dataset includes data about two topics: gossipco and politifact. Each topic includes two files.

There are two files in the politifact dataset: politifact_real.csv, which includes 432 tweets and contains samples relevant to real news. politifact_fake.csv contains 618 tweets and samples related to fake news.

There are two files in the gossipco dataset: gossipcop_real.csv, which includes 5328 tweets and contains samples relevant to real news. gossipco_fake.csv contains 5322 tweets and samples related to fake news.

Each file includes id, URL, title, and tweet. We create a new dataset merged between four files and add a new column; the label column consists of two values 0 belongs to fake news and 1 belongs to real news. The total number of tweets is 44280 tweets. The FakeNewsNet has split into 80% training set and 20% testing set. The statistics of the training set and testing set for dataset3 are shown in Table 3.

3) FA-KES5 (dataset3)

The FAKES5 [35] dataset includes 804 article news about Syrian war. Also, it includes a set of articles labeled by 0 (fake) or 1 (real). Each article has the headline, date, location, and full body of text. The 426 articles are true, and the 376 are fake. The statistics of the training set and testing set for dataset3 are shown in Table 4.

1) **TABLE 1.** Summary of the comparison of existing work

Ref.	Highlighted	Techniques	Datasets Name	Optimization methods
[33] (2020)	FakeNewsNet description and fakeNewsNet benefits	SVM LR, CNN, NB	FakeNewsNet	No
[27] (2020)	Hybrid CNN-LSTM to classify the fake news.	hybrid CNN-LSTM model	Fake news dataset from Kaggle	No
[29] (2021)	Hybrid CNN-RNN for fake news classification	LR, SVM, NB, Stochastic Gradient Decent (SDG), Ada Boost, RNN, CNN hybrid CNN-RNN	ISOT dataset FA-KES dataset	No
[28] (2020)	FNDNet model for fake news detection which can learn the features for fake news automatically	RNN, CNN, FNDNet (deep CNN)	Dataset from Kaggle	yes
[32] (2021)	The substance of the news piece and the prevalence of echo chambers in the social network.	Deep neural network	BuzzFeed PolitiFact	No
[31] (2021)	Proposing DeepFakE model for fake news article and also echo chambers existence.	XGBoost and DeepFakE: a multi-layer deep neural network	BuzzFeed PolitiFact	No
[30] (2019)	MCNN-TFW, a multiple-level CNN-based fake news detection system in cultural communication	MCNN	Weibo NewsFN	No
Our work	OPCNN-FAKE for detecting fake news. Applying optimization method for OPCNN-FAKE, LSTM, RNN and ML modes for enhance performance. Comparing OPCNN-FAKE with ML models, RNN, and LSTM using four benchmark datasets. The performance of models was registered for cross-validation result and testing result.	OPCNN-FAKE, SVM, NB, LR, DT, RF, RNN, LSTM,	Four datasets: Dataset1 FakeNewsNet FA-KES5 ISOT	Yes

TABLE 2. The statistics of dataset1

Dataset	News type	Total size
Training dataset	Real news	1494
	Fake news	1696
Testing dataset	Real news	374
	Fake news	424

TABLE 3. The statistics of FakeNewsNet (dataset2)

Dataset	News type	Total size
Training dataset	Real news	26907
	Fake news	8517
Testing dataset	Real news	6727
	Fake news	2129

TABLE 4. The statistics of the dataset3

Dataset	News type	Total size
Training dataset	Real news	341
	Fake news	302
Testing dataset	Real news	85
	Fake news	76

4) The ISOT (dataet4)

The ISOT dataset [20] consists of 44202 news articles, the 21416 news are true and 22756 of news are fake categories. Real news were collected from the Reuters website, and fake news were collected from Wikipedia7 and from Politifact website. Each news consists of title, text, date and subject. The dataset includes two files: fake file and real file. We create a new dataset merged between two files and added a new column, the label column that consists of two values 0 belongs to fake news and 1 belongs to real news. The statistics of the training set and testing set for dataset4 are shown in Table 5 .

TABLE 5. The statistics of The ISOT (dataset4)

Dataset	News type	Total size
Training dataset	Real news	17133
	Fake news	18204
Testing dataset	Real news	4283
	Fake news	4552

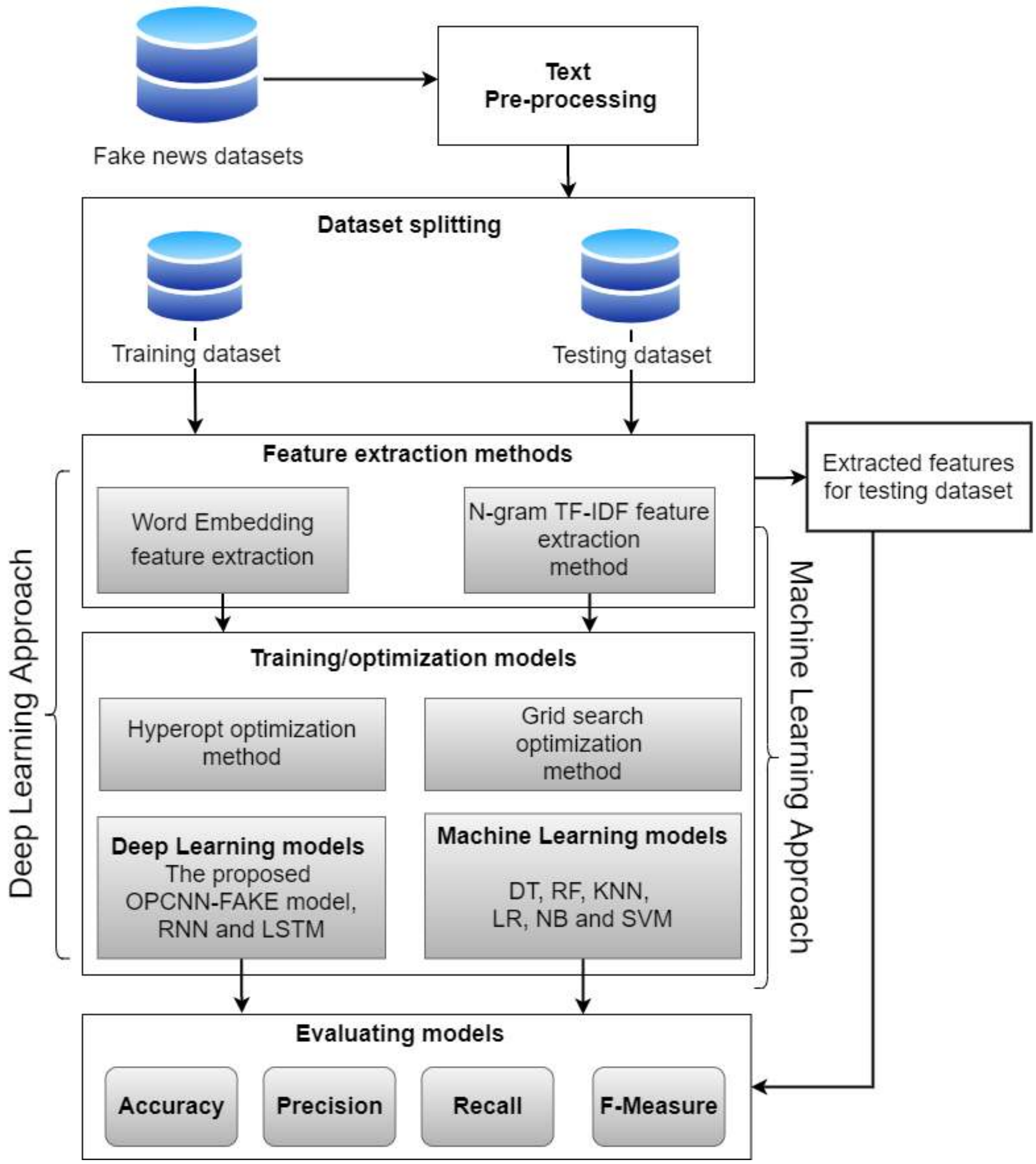
B. DATA PREPROCESSING

Data preprocessing is a critical step of natural language processing, such as fake news detection, as it directly impacts the model's effectiveness to the complexity of the data. Fake news datasets consist of many links, hashtags, special symbols, .etc. Therefore, we applied many steps of preprocessing to each dataset. These steps are described as follows.

- **Lower casing**: The most effective kind of text pre-processing is lowercasing, which ensures correlation within the feature set and solves the sparsity problem.
- **Removal of URL's**: Irrelevant links embedded into news have been removed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

FIGURE 1. The architecture of the proposed system



- **Removal of special symbols** such as punctuations, emojis, , ,, ', ;, #, $, %, & etc.
- **Removal of Stop Word**: Stop words are small words in a language that are useless in text mining and are utilised to structure language grammar. These stop words have been filtered away, including articles, conjunctions, prepositions, some pronouns, and common terms like the, a, an, about, by, from, to, and so on.
- **Tokenization** Tokenization in preprocessing is the process of dividing lengthy text sequences into tokens (i.e., smaller pieces).For example, consider this sentence before tokenization: "Fake news dataset", after tokenization it comes 'Fake', 'news',' dataset'.
- **Stemming** The stemming step is the process of changing the words into their original form. For example, the words " Walking", " Walked" and "Walker" will be reduced to the word "walk".

C. DATA SPLITTING

Using a stratified technique, each dataset is divided into 80% training dataset and 20% testing dataset (unseen dataset). The training dataset is used to optimize and train the machine learning models and deep learning models, while the unseen dataset is used to evaluate the machine learning models and deep learning models.

D. FEATURE EXTRACTION METHODS

N-gram with TF-IDF are used to extract features for the ML models and build feature matrix. To describe the context of the text, we employed several sizes of N-gram approach, ranging from $n=1$ to $n=4$ (i.e., uni-gram, bi-gram, tri-gram, and four-gram). TF-ID assigns a weight to each word representing the importance of the word in the document and corpus.

Word embedding is a technique for converting text data (words) into vectors. Every word is represented as an n-dimensional dense vector, with vectors that are comparable for similar words. We used the Golve [36] for word embedding to build embedding matrix. GloVe is an unsupervised learning technique that generates word vector representations. The resulting representations highlight intriguing linear substructures of the word vector space, which are trained using aggregated global word-word co-occurrence statistics from a corpus. We utilised glove.6B.zip, which contains vectors in four different dimensions: 25d, 50d, 100d, and 200d. The embedding matrix was constructed using 200d vectors.

E. THE PROPOSED MODEL (OPCNN-FAKE)

In this section, we describe the architecture of the proposed OPCNN-FAK model as shown in Figure 2 that is used to detect fake news detection. And, we describe optimization methods to select the best values for OPCNN-FAKE's parameters. OPCNN-FAK consists of six layers: an embedding layer, dropout layer, a convolutional layer, a pooling layer, flatten layer, and an output layer.

- **In the embedding layer**, each news is embedded at the word level and is represented as a matrix with each row corresponding to a word. It is implemented in the Keras library [37]. It has three arguments: the input-dim parameter represents the vocabulary size in the dataset, the output-dim parameter describes the vector space in which words will be embedded, and the input-length parameter describes the length of input sequences. We configured the output-dim as 200 because the length of Golve is 200d vectors and input-dim as 20000, and the input-length as 32.
- **The dropout layer** is an efficient regularization technique that prevents overfitting and reduces the complexity of model [38]. It receives the output of the embedding layer. We adopted the value of dropout using optimization methods range from 0.1 to 0.9.
- **Convolutional layer** receives the output of the dropout layer to reduce the complexity of the model. It includes a convolution filter and feature map (kernel). The convolution filter is applied to the input word matrix to produce a feature map indicating valuable input data patterns. Additionally, each filter employs the Rectified Linear Unit (ReLU) activation function [39] to identify multiple features in news. We used ReLU as the activation function in our DL. It is able to remove negative values from an activation map in a given network by setting them to zero. The most significant benefit of ReLu is the non-saturation of gradient, which considerably accelerates stochastic gradient descent convergence when compared to other activation functions [39]. Furthermore, it addresses the vanishing gradient problem and is more computationally efficient than sigmoid or tanh activation functions.
- **The pooling layer** uses the max operation to reduce the features in the feature map. Choosing the highest value is to capture the most significant features while reducing the amount of computation required in the next layer.
- **The flatten layer** has converted the text into a 1-dimensional array for inputting it to the next layer.
- **The output layer** gets the flatten layer's output to produce the model's final output, in which the neural network model identifies the news as real or fake. It has one neuron that determines if the news is fake or not. The ADAM optimizer [40] was used in this layer, and the activation function is sigmoid [41].

TABLE 6. The values of parameters have adapted for OPCNN-FAKE

Parameter	Values
Dropout rate	between the range of 0.1 to 0.9 rate
Filter sizes	32,64,128
Kernel size	2,3,4
Pool size	3,6
Batch size	73, 146, 219, 500, 1000, 100
Epochs.	within the range of 1 epoch to 200 epoch

For optimization method, A crucial aspect of DL solutions is the selection of hyperparameters. Distributed asynchronous hyper-parameter optimization (hyperopt) [42] technique has been used to optimize RNN, LSTM, and the OPCNN-FAKE. Hyperopt has been designed to accommodate Bayesian optimization algorithms based on Gaussian processes and regression trees. For OPCNN-FAKE, we adapted sets of values for different parameters in OPCNN-FAKE: filter sizes, kernel size, pool size, dropout, batch size, and epochs. Table 6 presents the values of parameters that have been adapted for OPCNN-FAKE.

F. RNN AND LSTM MODELS

We used RNN [43], LSTM [44]. Figure 3 shows architecture of RNN and LSTM models. It consists of five layers: an embedding layer, hidden layers, dropout layer, flatten layer, and an output layer.

The embedding layer is the first layer and it is a similar layer in OPCNN-FAKE. In hidden layers, RNN [13] and LSTM [45] have been used used. For each model one layer and two layers hidden layers have been used. For each hidden layer, L2 weight regularization technique [46] has been used by adopting reg_rate value for l2. Dropout layer has been used for each hidden layer. The next layer is flatten layer that converts the text into the single long feature vector. The output layer gets the flatten layer's output to produce the model's final output, in which the neural network model identifies the news as real or fake. It has one neuron that determines if the news is fake or not. The ADAM optimizer

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

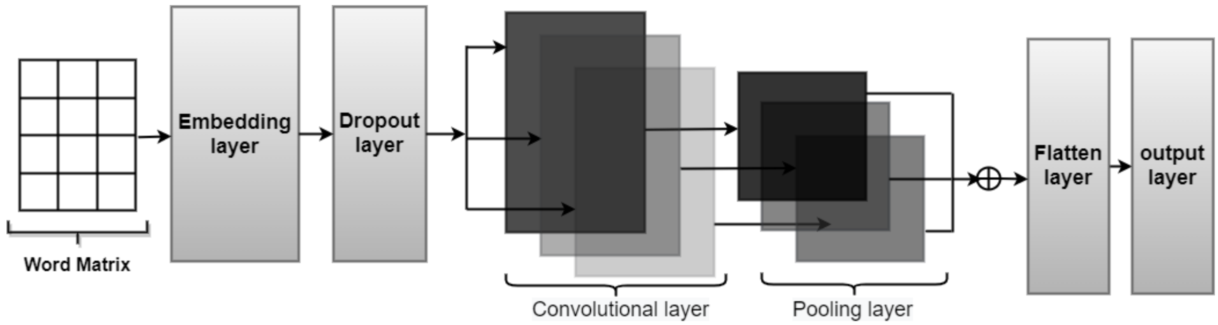


FIGURE 2. The architecture of the proposed OPCNN-FAKE model

TABLE 7. The values of parameters have been adapted for RNN and LSTM.

Parameter	Values
Dropout rate	between the range of 0.1 to 0.9 rate
number of neurons	10 to 200 neurons
batch_size	73, 146, 219, 500, 1000, 100
epochs.	within the range of 1 epoch to 200 epoch
reg_rate	0.01, 0.05, 0.1, .2,.3,.4,.5
Epochs.	within the range of 1 epoch to 200 epoch

was used in this layer, and the activation function is sigmoid.

For optimization RNN and LSTM models, the hyperopt optimization technique is used. we adapted sets of values for different parameters in RNN and LSTM: number of neurons, dropout, reg_rate, batch size, and epochs. Table 7 presents the values of parameters that have been adapted for RNN and LSTM.

Regular ML models

Six Regular ML models: DT [47], LR [48], KNN [49], RF, SVM [50], and NB [51]) were used to compare with OPCNN-Fake.

For optimization ML models, There are many ways to optimize hyperparameters, including grid search, random search, Bayesian optimization, hyperband optimization, gradient-based optimization, and metaheuristics optimization. Each method has its advantages and disadvantages. For example, Hyperparameter optimization search space is not convex and not differentiable, where it is impossible to reach the global optimum. On the other hand, grid search does an exhaustive search in the hyperparameter's search space. This allows the grid search to reach the best results compared to other techniques, especially when hyperparameters are not significant. As a result, we expect that this technique achieved the best results. Grid search with stratified 10-fold cross-validation was used to select the best value for each parameter of regular ML models. Grid search is used to find the optimal hyperparameters of a model that achieves the best performance of ML models. We define the set of values for each parameter of models. Then, the model tests all values for each parameter using stratified 10-fold cross-validation and selects the best values that achieve the best performance. In fold cross-validation, the dataset is split into k equal divisions, with k-1 groups utilised for training and one fold reserved for testing.

G. EVALUATING THE MODELS

The accuracy, precision, recall, and F1-score of the models were used to evaluate models. TP stands for true positive, TN stands for true negative, FP stands for false positive, and FN stands for false negative. Equations 1-4 can be found here.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}. \quad (1)$$

$$Precision = \frac{\text{TP}}{\text{TP + FP}} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (4)$$

IV. EXPERIMENTS RESULTS

A. EXPERIMENT SETUP

The experiments of this paper were conducted on a Google Colab RAM 25 GB, Python 3, and GPU. The Keras library implemented the OPCNN-FAKE, RNN, and LSTM. The sci-kit-learn package implemented the ML models. The hyperopt library, grid search have optimized DL models and ML models, respectively. To initialize the embedding layer, we use the 200-dimensional word vectors pre-trained in the Glove set. Four benchmark fake news datasets were split into 80% training datasets used to optimize models and register cross-validation results, 20% of testing datasets (unseen data) to evaluate models and register testing results. All the experiments were run 10 times separately.

B. RESULT OF DATASET1

The performance of cross-validation and the testing results for ML models and DL models will be discussed in the two sections.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

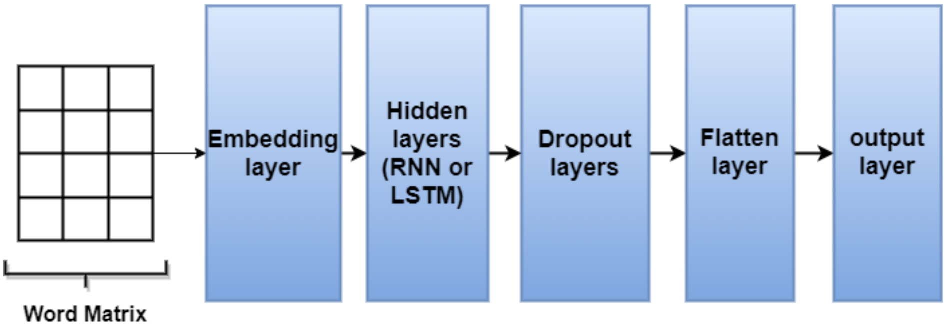


FIGURE 3. The architecture of RNN and LSTM.

1) Regular ML models results

Table 8 shows the performance of cross-validation and testing validation of applying regular ML to dataset1. The performance of cross-validation and the testing results will be discussed in the two sections.

1) Cross-validation results
For DT, unigram has obtained the highest performance (Accuracy = 95.63%, Precision = 95.77%, Recall = 95.67% and F1-score = 95.65%). Four-gram has obtained the lowest performance (Accuracy = 94.98%, Precision = 95.05%, Recall = 95.08% and F1-score = 95.05%). For KNN, four-gram has obtained the highest performance (Accuracy = 95.1%, Precision = 95.13%, Recall = 95.1% and F1-score 95.1%). While, unigram has obtained the lowest performance (Accuracy = 92.68%, , Precision = 95.77%, Recall = 95.67% and F1-score = 95.65%). As the same, for LR, four-gram has obtained the highest performance (Accuracy = 98.5%, Precision = 98.5%, Recall = 98.5% and F1-score 98.5%). While, unigram has obtained the lowest performance (Accuracy = 98.3%, Precision = 98.31%, Recall = 98.3% and F1-score = 98.3%). For RF, Tri-gram has obtained the highest performance (Accuracy = 98.75%, Precision = 98.82%, Recall = 98.78% and F1-score 98.8%). While, unigram has obtained the lowest performance (Accuracy = 96.07%, Precision = 97.08%, Recall = 98.02% and F1-score = 96.07%). For SVM, tri-gram has obtained the highest performance (Accuracy = 98.32%, Precision = 98.34%, Recall = 32% and F1-score 98.3%). While Unigram has obtained the lowest performance (Accuracy = 96.07%, Precision = 97.08%, Recall = 98.02% and F1-score = 96.07%). For NB, Unigram has obtained the highest performance (Accuracy = 94.49%, Precision = 94.49%, Recall = 94.49% and F1-score 94.4%). While four-gram has obtained the lowest performance (Accuracy = 92.92%, , Precision = 93.6%, Recall = 92.92% and F1-score = 92.9%).
2) Testing results

For DT, unigram has obtained the highest performance (Accuracy = 90.0%, Precision = 90.03%, Recall = 9.0% and F1-score = 89.99%). Four-gram has obtained the lowest performance (Accuracy = 89.28%, Precision = 89.28%, Recall = 89.28% and F1-score = 89.27%). For KNN, four-gram has obtained the highest performance (Accuracy = 90.05%, Precision = 90.69%, Recall = 90.05% and F1-score 90.05%) , Unigram has obtained the lowest performance (Accuracy = 87.39%, , Precision = 89.28%, Recall = 89.28% and F1-score = 89.27%). As the same, for LR, four-gram has obtained the highest performance (Accuracy = 96.9%, Precision = 96.92%, Recall = 96.9% and F1-score 96.9%) , Unigram has obtained the lowest performance (Accuracy = 96.24%, Precision = 96.26%, Recall = 96.24% and F1-score = 96.24%). For RF, tri-gram has obtained the highest performance (Accuracy = 96.97%, , Precision = 97.04%, Recall = 96.97% and F1-score = 96.97%). While, unigram has obtained the lowest performance (Accuracy = 94.2%, , Precision = 94.23%, Recall = 94.2% and F1-score = 94.2%). For SVM, tri-gram has obtained the highest performance (Accuracy = 96.29%, , Precision = 96.31%, Recall = 96.29% and F1-score = 96.29%). While, unigram has obtained the lowest performance (Accuracy = 95.3%, , Precision = 95.3%, Recall = 95.3% and F1-score = 95.3%). For NB, Unigram has obtained the highest performance (Accuracy = 91.57%, Precision = 92.27%, Recall = 91.57% and F1-score 91.57%). While four-gram has obtained the lowest performance (Accuracy = 90.44%, Precision = 91.74%, Recall = 90.44% and F1-score = 90.42%).

Overall, RF with Tri-gram has ranked the highest performance for cross-validation results and the testing results compared with other regular ML models.

C. DL MODELS RESULTS

Table 9 shows the cross-validation and test results of OPCNN-FAKE, LSTM, RNN and for dataset1.

1) Cross-validation results

We can see that OPCNN-FAKE has ranked the highest performance (Accuracy = 99.9%, Precision = 100%,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

TABLE 8. The performance of ML for dataset1

Models	Matrix size	Cross-validation performance				Test performance			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
DT	**Unigram**	**95.63±1.29**	**95.77±1.12**	**95.67±1.37**	**95.65±1.24**	**90.0**	**90.03**	**90.0**	**89.99**
	Bi-gram	95.02±1.34	95.11±1.33	95.01±1.18	95.01±1.26	89.95	89.99	89.95	89.93
	Tri-gram	95.07±1.4	95.02±1.35	95.09±1.26	95.04±1.21	89.96	89.99	89.96	89.95
	Four-gram	**94.98±1.38**	**95.05±1.34**	**95.08±1.28**	**95.05±1.2**	**89.28**	**89.28**	**89.28**	**89.27**
KNN	**Unigram**	**92.68±1.38**	**92.73±1.38**	**92.68±1.38**	**92.68±1.38**	**87.39**	**88.26**	**87.39**	**87.39**
	Bi-gram	94.49±1.27	94.52±1.26	94.49±1.27	94.49±1.27	89.37	90.01	89.37	89.38
	Tri-gram	94.94±1.28	94.97±1.27	94.94±1.28	94.93±1.29	89.81	90.45	89.81	89.81
	Four-gram	**95.1±1.22**	**95.13±1.21**	**95.1±1.22**	**95.1±1.22**	**90.05**	**90.69**	**90.05**	**90.05**
LR	**Unigram**	**98.3±0.72**	**98.31±0.71**	**98.3±0.72**	**98.3±0.72**	**96.24**	**96.26**	**96.24**	**96.24**
	Bi-gram	98.41±0.69	98.42±0.69	98.41±0.69	98.41±0.69	96.76	96.78	96.76	96.76
	Tri-gram	98.43±0.64	98.44±0.63	98.43±0.64	98.43±0.64	96.82	96.84	96.82	96.82
	Four-gram	**98.5±0.67**	**98.5±0.67**	**98.5±0.67**	**98.5±0.67**	**96.9**	**96.92**	**96.9**	**96.9**
RF	**Unigram**	**98.07±0.72**	**98.05±0.76**	**98.02±0.72**	**98.02±0.76**	**94.2**	**94.23**	**94.2**	**94.2**
	Bi-gram	98.71±0.61	98.7±0.56	98.68±0.63	98.62±0.61	95.79	95.81	95.79	95.79
	Tri-gram	**98.75±0.67**	**98.82±0.6**	**98.78±0.62**	**98.82±0.6**	**96.97**	**97.04**	**96.97**	**96.97**
	Four-gram	98.4±0.67	98.42±0.6	98.48±0.62	98.42±0.6	96.46	96.49	96.46	96.46
SVM	**Unigram**	**98.32±0.71**	**98.34±0.7**	**98.32±0.71**	**98.32±0.71**	**95.3**	**95.4**	**95.3**	**95.3**
	Bi-gram	98.34±0.7	98.35±0.7	98.34±0.7	98.34±0.7	95.38	95.46	95.38	95.38
	Tri-gram	**98.42±0.7**	**98.43±0.69**	**98.42±0.7**	**98.42±0.7**	**96.29**	**96.31**	**96.29**	**96.29**
	Four-gram	98.37±0.69	98.39±0.69	98.37±0.69	98.37±0.69	95.52	95.52	95.52	95.53
NB	**Unigram**	**94.49±1.14**	**94.79±1.05**	**94.49±1.14**	**94.49±1.14**	**91.57**	**92.27**	**91.57**	**91.57**
	Bi-gram	94.17±1.35	94.3±1.31	94.17±1.35	94.18±1.35	91.2	92.14	91.2	91.2
	Tri-gram	93.41±1.23	93.94±1.07	93.41±1.23	93.41±1.23	90.95	92.07	90.95	90.94
	Four-gram	**92.92±1.17**	**93.6±0.97**	**92.92±1.17**	**92.92±1.18**	**90.44**	**91.74**	**90.44**	**90.42**

Recall = 99.97% and F1-score 95.9%). While, RNN two layers has ranked the lowest performance (Accuracy = 85.94%, , Precision = 87.18%, Recall = 82.92% and F1-score = 95.38%). The second highest performance has registered by LSTM with two layers (Accuracy = 96.13%, Precision = 96.0%, Recall = 95.93% and F1-score 95.86%).

2) Testing result
We can see that OPCNN-FAKE has obtained the highest performance (Accuracy = 97.84%, Precision = 97.86%, Recall = 97.84% and F1-score 97.84%). While, RNN two layers has obtained the lowest performance (Accuracy = 86.76%, Precision = 86.97%, Recall = 86.76% and F1-score = 86.72%).

Overall, **the OPCNN-FAKE** has ranked the highest performance for cross-validation results and the testing results compared with other regular ML models, and DL models (RNN and LSTM).

1) The best values of parameters for dataset1

Table 10 shows the best values of parameters for the OPCNN-FAKE model. Table 11 presents the best values parameters for RNN and LSTM.

D. RESULT OF DATASET2 (FAKENEWSNET)

The performance of cross-validation and the testing results for ML models and DL models will be discussed in the two sections.

1) Regular ML models results

Table 12 shows the performance of cross-validation and testing validation of applying regular ML to detest2. The performance of cross-validation and the testing results will be discussed in the two sections.

1) Cross-validation result
In DT and KNN, we can see that the value of Accuracy, Precision, Recall, and F1-score for unigram, bi-gram, tri-gram and four-gram are similar. For DT, four-gram has obtained the highest performance (Accuracy = 92.49%, Precision = 92.36%, Recall = 92.48% and F1-score = 92.37%). While, unigram has obtained the lowest performance (Accuracy = 92.41%, Precision = 92.3%, Recall =92.42% and F1-score = 92.29%). For KNN, Unigram has obtained the highest performance (Accuracy = 93.7%, Precision = 93.8%, Recall = 93.7% and F1-score = 93.47%). While, unigram has obtained the lowest performance (Accuracy = 93.45%, Precision = 93.64%, Recall =93.45% and F1-score = 93.17%). For LR, Unigram has obtained the highest performance (Accuracy = 93.7%, Precision = 93.8%, Recall = 93.7% and F1-score = 93.47%). While, Four-gram has obtained the lowest performance (Accuracy = 87.28%, Precision = 86.82%, Recall =87.28% and F1-score = 86.84%). For RF, Unigram has obtained the highest performance (Accuracy = 92.36%, Precision = 92.42%, Recall = 92.07% and F1-score = 83.68%). While, Four-gram has obtained the lowest performance (Accuracy = 91.28%, Precision = 91.38%, Recall =91.31% and F1-score = 90.83%). For SVM, Unigram has obtained the highest performance (Accuracy = 93.85%, Pre-

TABLE 9. The performance of ML for dataset1

Models	Cross-validation performance				Testing performance			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
OPCNN-FAKE	**99.99±0.0**	**100.0±0.0**	**99.97±0.01**	**99.98±0.0**	**97.84**	**97.86**	**97.84**	**97.84**
LSTM one layer	95.98±1.52	95.57±2.11	95.61±2.52	95.5±1.74	92.56	92.58	92.56	92.55
LSTM two layers	96.13±1.25	96.0±2.19	95.93±2.6	95.86±1.56	90.9	91.07	90.9	90.89
RNN one layer	95.85±1.52	96.05±2.08	94.95±2.78	95.38±1.71	90.64	90.72	90.64	90.63
RNN two layers	**85.94±1.24**	**87.18±3.7**	**82.92±5.59**	**84.62±1.83**	**86.76**	**86.97**	**86.76**	**86.72**

TABLE 10. The best values of OPCNN-FAKE's parametersdataset1

Models	Filter size	Kernel size	Max pooling	Dropout	Batch size	epochs
OPCNN-FAKE	128	2	3	0.2	219	28

TABLE 11. The best values parameters for LSTM and RNN for dataset1

Models	Neurons	reg_rate	Dropout	Batch size	epochs
LSTM one layer	48	0.05	0.5	146	32
LSTM two layers	[40,41]	[0.1,0.4]	[0.4,0.4]	146	12
RNN one layer	34	0.05	0.4	73	11
RNN two layers	[5,42]	[0.3,0.1]	[0.6,0.6]	219	11

cision = 93.89%, Recall = 93.85% and F1-score = 93.64%). While, Four-gram has obtained the lowest performance (Accuracy = 93.44%, Precision = 93.41%, Recall =93.44% and F1-score = 93.26%). For NB, Unigram has obtained the highest performance (Accuracy = 86.03%, Precision = 85.47%, Recall = 85.3% and F1-score = 81.26%). While, Four-gram and Tri-gram have the same performance.

2) Testing result
For DT, Four-gram has obtained the highest performance (Accuracy = 81.8%, Precision = 81.47%, Recall = 81.8% and F1-score = 81.62%), while Unigram has obtained the lowest performance (Accuracy = 81.27%, Precision = 81.04%, Recall = 81.27% and F1-score = 81.15%). For KNN, Unigram has obtained the highest performance (Accuracy = 81.87%, Precision = 84.15%, Recall = 81.87% and F1-score = 77.69%). While, unigram has obtained the lowest performance (Accuracy = 80.9%, Precision = 83.57%, Recall =80.9% and F1-score = 75.96%). For LR, Unigram has obtained the highest performance (Accuracy = 84.18%, Precision = 83.51%, Recall = 84.18% and F1-score = 83.68%). While, Four-gram has obtained the lowest performance (Accuracy = 82.28%, Precision = 82.45%, Recall =82.28% and F1-score = 82.28%). For RF, Unigram has obtained the highest performance (Accuracy = 85.34%, Precision = 84.8%, Recall = 85.34% and F1-score = 84.2%). While, Four-gram has obtained the lowest performance (Accuracy = 84.63%, Precision = 84.04%, Recall =84.63% and F1-score = 83.26%). For SVM, Unigram has obtained the highest performance (Accuracy = 83.94%, Precision = 84.24%, Recall = 83.94% and F1-score = 81.26%). While, Four-gram has obtained the lowest performance (Accuracy = 82.72%, Precision = 82.89%, Recall =82.72% and F1-score = 80.36%). Unigram has obtained the highest performance (Accuracy = 84.53%, Precision = 83.73%, Recall = 84.53% and F1-score = 83.56%). While, Four-gram and Tri-gram have the same performance.

Overall, SVM with Unigram has ranked the highest performance for cross-validation results than other regular ML models. And NB with Unigram has rated the highest performance for the testing results than other regular ML models.

2) DL models results

Table 13 shows the cross-validation and test results of LSTM, RNN and OPCNN-FAKE for dataset2.

1) Cross-validation result
We can see that OPCNN-FAKE has obtained the highest performance (Accuracy = 98.65%, Precision = 99.34%, Recall = 98.87% and F1-score 99.1%). While, RNN two layers has obtained the lowest performance (Accuracy = 79.49%, Precision = 79.51%, Recall = 98.32% and F1-score = 87.8%). The second highest performance has registered by LSTM with two layers (Accuracy = 85.96%, Precision = 87.55%, Recall = 95.15% and F1-score 91.16%).
2) Testing result
We can see that OPCNN-FAKE has obtained the highest performance (Accuracy = 95.26%, Precision = 95.28%, Recall = 95.26% and F1-score 97.895.274%). While, RNN two layers has obtained the lowest performance (Accuracy = 79.84%, Precision = 80.23%, Recall = 79.84% and F1-score = 74.91%).

Overall, **the proposed OPCNN-FAKE** has ranked the highest performance for cross-validation results and the testing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

TABLE 12. The performance of ML for dataset2 (FakeNewsNet)

Models	Matrix size	Cross-validation performance				Test performance			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
DT	**Unigram**	**92.41±0.42**	**92.3±0.44**	**92.42±0.41**	**92.29±0.42**	**81.27**	**81.04**	**81.27**	**81.15**
	Bi-gram	92.46±0.38	92.36±0.36	92.47±0.4	92.35±0.4	81.54	81.28	81.54	81.4
	Tri-gram	92.47±0.4	92.33±0.42	92.46±0.43	92.35±0.42	81.38	81.11	81.38	81.23
	Four-gram	**92.49±0.43**	**92.36±0.44**	**92.48±0.39**	**92.37±0.43**	**81.8**	**81.47**	**81.8**	**81.62**
KNN	**Unigram**	**93.7±0.34**	**93.8±0.34**	**93.7±0.34**	**93.47±0.36**	**81.87**	**84.15**	**81.87**	**77.69**
	Bi-gram	93.63±0.35	93.73±0.34	93.63±0.35	93.39±0.38	81.13	83.65	81.13	76.39
	Tri-gram	93.62±0.35	93.72±0.34	93.62±0.35	93.38±0.37	80.97	83.59	80.97	76.1
	Four-gram	**93.45±0.34**	**93.64±0.33**	**93.45±0.34**	**93.17±0.37**	**80.9**	**83.57**	**80.9**	**75.96**
LR	**Unigram**	**87.86±0.47**	**87.5±0.5**	**87.86±0.47**	**87.57±0.48**	**84.18**	**83.51**	**84.18**	**83.68**
	Bi-gram	87.39±0.5	87.08±0.53	87.39±0.5	87.17±0.52	83.42	83.02	83.42	83.18
	Tri-gram	87.42±0.46	87.05±0.49	87.42±0.46	87.14±0.48	83.93	83.27	83.93	83.46
	Four-gram	**87.28±0.48**	**86.82±0.52**	**87.28±0.48**	**86.84±0.5**	**82.28**	**82.45**	**82.28**	**82.28**
RF	**Unigram**	**92.36±0.2**	**92.42±0.3**	**92.3±0.37**	**92.07±0.31**	**85.34**	**84.8**	**85.34**	**84.2**
	Bi-gram	92.17±0.15	92.36±0.22	92.26±0.18	91.96±0.29	85.28	84.63	85.28	84.1
	Tri-gram	92.3±0.21	92.34±0.27	92.24±0.3	91.91±0.28	85.02	84.33	85.02	83.97
	Four-gram	**91.28±0.39**	**91.38±0.44**	**91.31±0.41**	**90.83±0.46**	**85.00**	**85.04**	**84.63**	**83.26**
SVM	**Unigram**	**93.85±0.27**	**93.89±0.28**	**93.85±0.27**	**93.64±0.29**	**83.94**	**84.24**	**83.94**	**81.62**
	Bi-gram	93.59±0.23	93.58±0.24	93.59±0.23	93.4±0.24	83.29	84.16	83.29	80.39
	Tri-gram	93.48±0.22	93.45±0.24	93.48±0.22	93.29±0.22	83.83	84.06	83.83	81.49
	Four-gram	**93.44±0.22**	**93.41±0.24**	**±0.22**	**93.26±0.22**	**82.72**	**82.89**	**82.72**	**80.36**
NB	**Unigram**	**86.03±0.49**	**85.47±0.56**	**86.03±0.49**	**85.3±0.53**	**84.53**	**83.73**	**84.53**	**83.56**
	Bi-gram	85.77±0.55	85.16±0.6	85.77±0.55	85.2±0.57	83.77	82.91	83.77	82.98
	Tri-gram	85.66±0.55	85.05±0.6	85.66±0.55	85.08±0.56	83.55	82.63	83.55	82.68
	Four-gram	85.66±0.54	85.04±0.6	85.66±0.54	85.07±0.56	83.52	82.59	83.52	82.63

results compared with other regular ML models and DL models (RNN and LSTM).

3) The best values of parameters's for dataset2 (FakeNewsNet)

Table 14 shows the best values of parameters for the OPCNN-FAKE model. Table 15 presents the best values parameters for RNN and LSTM.

E. RESULT OF DATASET3

The performance of cross-validation and the testing results for ML models and DL models will be discussed in the two sections.

Regular ML models results Table 16 shows the performance of cross-validation and testing validation of applying regular ML to detest3. The performance of cross-validation and the testing results will be discussed in the two sections.

1) Cross-validation results
For DT, Unigram has obtained the highest performance (Accuracy = 51.02%, Precision = 51.11%, Recall = 51.01% and F1-score = 51.39%). While, Bi-gram has obtained the lowest performance (Accuracy = 49.46%, Precision = 50.21%, Recall =49.56% and F1-score = 49.91%). For KNN, Four-gram has obtained the highest performance (Accuracy = 53.17%, Precision = 51.9%, Recall = 53.17% and F1-score = 48.72%). While, Unigram has obtained the lowest performance (Accuracy = 48.76%, Precision = 51.64%, Recall =51.64% and F1-score = 50.67%). For LR, Tri-gram has obtained the highest performance (Accuracy = 51.66%, Precision = 51.64%, Recall = 51.66% and F1-score = 51.44%). For RF, Four-gram has obtained the highest performance (Accuracy = 52.90%, Precision = 52.80%, Recall = 52.64% and F1-score = 52.36%). While, Unigram has obtained the lowest performance (Accuracy = 50.6%, Precision = 51.52%, Recall =52.17% and F1-score = 51.69%). For SVM, Unigram has obtained the highest performance (Accuracy = 53.05%, Precision = 53.34%, Recall = 53.34% and F1-score = 52.58%). While, Four-gram has obtained the lowest performance (Accuracy = 46.96%, Precision = 50.05%, Recall =51.57% and F1-score = 46.96%). For NB, Unigram has obtained the highest performance (Accuracy = 53.37%, Precision = 53.93%, Recall = 53.93% and F1-score = 53.7%). While, Four-gram has obtained the lowest performance (Accuracy = 51.22%, Precision = 51.22%, Recall =51.47% and F1-score = 51.0%).
2) Testing results
For DT, Unigram has obtained the highest performance (Accuracy = 48.76%, Precision = 48.84%, Recall = 48.76% and F1-score = 48.72%). While, Bi-gram has obtained the lowest performance (Accuracy = 46.58%, Precision = 46.58%, Recall =46.58% and F1-score = 46.48%). For KNN, Four-gram has obtained the highest performance (Accuracy = 51.43%, Precision = 49.9%, Recall = 51.43% and F1-score = 48.74%). While, Unigram has obtained the lowest performance (Accuracy = 49.63%, Precision = 48.97%, Recall =49.63% and F1-score = 46.98%). For LR, Tri-gram has obtained the highest performance (Accuracy = 51.92%, Precision = 51.68%, Recall = 51.92% and F1-score = 51.66%). For RF, Four-gram has obtained the highest performance (Accuracy =

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

TABLE 13. The performance of deep neural networks for dataset2 (FakeNewsNet)

Models	Cross-validation performance					Testing performance			
	Accuracy	Precision	Recall	F1-score		Accuracy	Precision	Recall	F1-score
OPCNN-FAKE	**98.65±0.25**	**99.34±0.17**	**98.87±0.24**	**99.1±0.18**		**95.26**	**95.28**	**95.26**	**95.27**
LSTM one layer	83.23±0.42	85.07±1.18	94.56±1.56	89.52±0.31		83.27	82.49	83.27	81.89
LSTM two layers	85.96±0.52	87.55±1.01	95.15±1.2	91.16±0.31		86.43	85.9	86.43	85.95
RNN one layer	82.26±0.36	84.0±1.04	94.65±1.57	88.96±0.26		82.02	81.63	82.02	81.8
RNN two layers	**79.49±0.57**	**79.51±0.8**	**98.32±0.63**	**87.89±0.27**		**79.84**	**80.23**	**79.84**	**74.91**

TABLE 14. The best values of parameters's OPCNN-FAKE for dataset2 (FakeNewsNet)

Model	Filter size	Kernel size	Max pooling	Dropout	Batch size	epochs
OPCNN-FAKE	128	4	3	0.6	219	40

TABLE 15. The best values parameters for LSTM and RNN for dataset2 (FakeNewsNet)

Models	Neurons	reg_{rate}	Dropout	Batch size	epochs
LSTM one layer	37	0.1	0.8	500	33
LSTM two layers	[48,21]	[0.1,0.5]	[0.7,0.8]	100	58
RNN one layer	14	0.1	0.1	219	81
RNN two layers	[24,30]	[0.5,0.5]	[0.2,0.7]	500	52

51.3%, Precision = 50.08%, Recall = 51.3% and F1-score = 52.36%). While, Unigram has obtained the lowest performance (Accuracy = 48.63%, Precision = 47.8%, Recall =48.63% and F1-score = 47.48%). For SVM, Unigram has obtained the highest performance (Accuracy = 52.36%, Precision = 50.7%, Recall = 52.36% and F1-score = 45.84%). While, Four-gram has obtained the lowest performance (Accuracy = 49.32%, Precision = 44.09%, Recall =49.32% and F1-score = 40.94%). For NB, Unigram has obtained the highest performance (Accuracy = 51.68%, Precision = 51.52%, Recall = 51.68% and F1-score = 51.54%). While, Four-gram has obtained the lowest performance (Accuracy = 48.88%, Precision = 48.682%, Recall =48.88% and F1-score = 48.67%).

DL models results Table 17 shows the cross-validation and test results of LSTM, RNN and OPCNN-FAKE for dataset2.

1) Cross-validation results
We can see that OPCNN-FAKE has obtained the highest performance (Accuracy = 97.23%, Precision = 97.37%, Recall = 97.58% and F1-score =97.26%). While, LSTM with one layer has obtained the lowest performance (Accuracy = 71.97%, Precision = 73.32%, Recall = 73.86% and F1-score = 73.85%). The second highest performance has registered by LSTM with two layers (Accuracy = 90.67%, Precision = 91.55%, Recall = 91.55% and F1-score 92.53%).
2) Testing results
We can see that OPCNN-FAKE has obtained the highest performance (Accuracy = 53.99%, Precision = 53.86%, Recall = 53.91% and F1-score =53.99%). While, LSTM with two layers obtained the lowest performance (Accuracy = 47.26%, Precision = 47.17%, Recall = 47.32% and F1-score = 47.26%).

Overall, **OPCNN-FAKE** has ranked the highest performance for cross-validation results and the testing results compared with other regular ML models and DL models (RNN and LSTM).

F. THE BEST VALUES OF PARAMETERS FOR DATASET3

Table 18 shows the best values of parameters for the OPCNN-FAKE model. Table 19 presents the best values parameters for RNN and LSTM.

G. RESULT OF DATASET4

The performance of cross-validation and the testing results for ML models and DL models will be discussed in the two sections.

1) Regular ML models results

Table 20 shows the performance of cross-validation and testing validation of applying regular ML to detest2. The performance of cross-validation and the testing results will be discussed in the two sections.

1) Cross-validation results
For DT, Bi-gram has obtained the highest performance (Accuracy = 99.58%, Precision = 99.59%, Recall = 99.58% and F1-score = 99.58%). While, Four-gram and Tri-gram have the same performance. For KNN, Four-gram has obtained the highest performance (Accuracy = 91.54%, Precision = 91.52%, Recall = 91.63% and F1-score = 91.54%). While, Unigram has obtained the lowest performance (Accuracy = 90.75%, Precision = 90.85%, Recall =90.75% and F1-score = 90.74%). For LR, Tri-gram has obtained the highest performance (Accuracy = 99.44%, Precision = 99.44%, Recall = 99.44% and F1-score = 99.44%). For RF, Bi-gram has obtained the highest performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

TABLE 16. The performance of regular ML for dataset3

Models	Matrix size	Cross-validation performance				Test performance			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
DT	**Unigram**	**51.02±6.05**	**51.11±5.36**	**51.01±4.93**	**51.39±5.94**	**48.76**	**48.84**	**48.76**	**48.72**
	Bi-gram	**49.46±5.49**	**50.21±5.93**	**49.56±5.73**	**49.91±5.78**	**46.58**	**46.58**	**46.58**	**46.48**
	Tri-gram	50.96±6.41	50.85±5.23	49.94±6.55	49.58±5.91	47.82	47.97	47.82	47.83
	Four-gram	50.49±5.64	50.58±5.38	50.14±5.21	50.0±5.31	48.07	48.04	48.07	47.96
KNN	**Unigram**	**48.76±5.35**	**51.64±5.06**	**51.64±5.06**	**50.67±6.7**	**49.63**	**48.97**	**49.63**	**46.98**
	Bi-gram	52.64±4.78	51.77±6.14	52.64±4.78	49.83±5.23	51.24	49.4	51.24	46.01
	Tri-gram	53.34±4.98	52.7±6.31	53.34±4.98	50.72±5.34	49.94	48.01	49.94	45.94
	Four-gram	**53.17±4.74**	**52.65±5.42**	**53.17±4.74**	**51.9±4.98**	**51.43**	**49.9**	**51.43**	**48.74**
LR	Unigram	49.46±5.58	50.05±5.56	50.05±5.56	49.69±5.86	47.08	46.2	47.08	46.03
	Bi-gram	51.18±5.31	51.02±5.44	51.18±5.31	50.83±5.31	50.43	50.26	50.43	50.27
	Tri-gram	**51.66±5.18**	**51.64±5.22**	**51.66±5.18**	**51.44±5.2**	**51.92**	**51.68**	**51.92**	**51.66**
	Four-gram	50.51±5.7	49.83±6.21	50.51±5.7	49.13±5.66	50.75	49.73	50.75	48.75
RF	**Unigram**	**50.6±5.33**	**51.52±4.31**	**52.17±4.41**	**51.69±5.75**	**48.63**	**47.8**	**48.63**	**47.48**
	Bi-gram	52.21±4.89	49.94±6.13	51.62±4.99	49.69±5.83	49.38	48.64	49.38	48.37
	Tri-gram	**52.90±4.78**	**52.80±5.49**	**52.64±5.42**	**52.36±5.15**	**51.3**	**50.08**	**51.3**	**48.66**
	Four-gram	52.72±5.5	52.18±5.7	51.69±5.09	51.97±5.08	50.0	49.3	50.0	48.50
SVM	**Unigram**	**53.05±4.17**	**53.34±4.94**	**53.34±4.94**	**52.58±7.92**	**52.36**	**50.7**	**52.36**	**45.84**
	Bi-gram	48.54±5.56	52.23±6.35	53.05±4.17	48.38±4.54	51.74	49.43	51.74	44.25
	Tri-gram	52.18±2.9	49.24±10.86	52.18±2.9	41.57±3.92	52.05	47.04	52.05	38.48
	Four-gram	**46.96±4.94**	**50.05±6.77**	**51.57±4.59**	**46.96±4.94**	**49.32**	**44.09**	**49.32**	**40.94**
NB	**Unigram**	**53.37±6.11**	**53.93±6.08**	**53.93±6.08**	**53.7±6.32**	**51.68**	**51.52**	**51.68**	**51.54**
	Bi-gram	53.28±5.84	53.22±6.0	53.28±5.84	52.96±5.85	51.49	51.31	51.49	51.32
	Tri-gram	52.34±5.66	52.11±5.93	52.34±5.66	51.87±5.7	49.19	48.8	49.19	48.76
	Four-gram	**51.22±5.59**	**51.22±5.59**	**51.47±5.34**	**51.0±5.44**	**48.88**	**48.68**	**48.88**	**48.67**

TABLE 17. The performance of deep neural networks for dataset3

Models	Cross-validation performance				Testing performance			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
OPCNN-FAKE	**97.23±0.26**	**97.37±0.24**	**97.58±0.36**	**97.26±0.21**	**53.99**	**53.86**	**53.91**	**53.99**
LSTM one layer	**71.97±6.26**	**73.32±6.62**	**73.86±7.21**	**73.85±9.13**	47.64	47.44	47.71	47.64
LSTM two layers	90.67±2.35	91.55±2.46	91.42±2.88	92.53±4.04	**47.26**	**47.17**	**47.32**	**47.26**
RNN one layer	74.23±2.04	79.02±1.43	78.16±1.43	91.08±1.92	48.57	44.24	45.93	48.57
RNN two layers	78.66±2.8	80.6±2.57	78.57±2.87	83.23±3.23	47.58	47.60	47.60	47.58

TABLE 18. The best values parameters' of OPCNN-FAKE for dataset3

Model	Filter size	Kernel size	Max pooling	Dropout	Batch size	epochs
OPCNN-FAKE	64	4	3	0.7	219	25

TABLE 19. The best values parameters for LSTM and RNN for dataset3

Models	Neurons	$reg_{r}ate$	Dropout	Batch size	epochs
LSTM one layer	37	0.4	0.8	146	40
LSTM two layers	[20,39]	[0.01,0.1]	[0.7,0.5]	219	14
RNN one layer	36	0.01	0.9	219	14
RNN two layers	[3,23]	[0.2,0.05]	[0.7,0.3]	73	14

(Accuracy = 99.75%, Precision = 99.74%, Recall = 99.73% and F1-score = 99.75%). For SVM, Bi-gram has obtained the highest performance (Accuracy = 99.65%, Precision = 99.64%, Recall = 99.63% and F1-score = 99.65%). While, Four-gram has obtained the lowest performance (Accuracy = 99.23%, Precision = 99.23%, Recall =99.23% and F1-score = 99.23%). For NB, Four-gram has obtained the highest performance (Accuracy = 95.11%, Precision = 95.11%, Recall = 95.11% and F1-score = 95.11%). While, Unigram has obtained the lowest performance (Accuracy = 92.89%, Precision = 92.9%, Recall =92.89% and F1-score = 92.89%).

2) Testing results
For DT, Bi-gram has obtained the highest performance (Accuracy = 99.40%, Precision = 99.40%, Recall = 99.40% and F1-score = 99.40%). While, Unigram, Four-gram and Tri-gram have the same performance. For KNN, Four-gram has obtained the highest performance (Accuracy = 90.72%, Precision = 90.72%, Recall = 90.93% and F1-score = 90.72%). While, Unigram has obtained the lowest performance (Accuracy = 90.75%, Precision = 90.85%, Recall =90.75% and F1-score = 90.74%). For LR, Tri-gram has obtained

the highest performance (Accuracy = 98.8%, Precision = 98.8%, Recall = 98.8% and F1-score = 98.8%). For RF, Bi-gram has obtained the highest performance (Accuracy = 99.90%, Precision = 99.90%, Recall = 99.90% and F1-score = 99.90%). For SVM, Bi-gram has obtained the highest performance (Accuracy = 99.40%, Precision = 99.40%, Recall = 99.40% and F1-score = 99.40%). While, Four-gram has obtained the lowest performance (Accuracy = 98.90%, Precision = 98.90%, Recall =98.90% and F1-score = 98.90%). For NB, Four-gram has obtained the highest performance (Accuracy = 94.94%, Precision = 94.94%, Recall = 94.94% and F1-score = 94.94%). While, Unigram has obtained the lowest performance (Accuracy = 92.9%, Precision = 92.9%, Recall =92.9% and F1-score = 92.9%).

Overall, RF with Tri-gram has ranked the highest performance for cross-validation results than other regular ML models. And NB with Four-gram has rated the highest performance for the testing results than other regular ML models.

H. DL MODELS RESULTS

Table 21 shows the cross-validation and test results of LSTM, RNN and OPCNN-FAKE for dataset4.

1) Cross-validation results
   We can see that OPCNN-FAKE has obtained the highest performance (Accuracy = 100%, Precision = 100%, Recall = 100% and F1-score 100%). While, RNN two layers has obtained the lowest performance (Accuracy = 79.49%, Precision = 87.89%, Recall = 79.51% and F1-score = 98.32%). LSTM one layer and LSTM two layers have the same performance.
2) Testing results
   We can see that OPCNN-FAKE has obtained the highest performance (Accuracy = 99.99%, Precision = 99.99%, Recall = 99.99% and F1-score 99.99%). While, RNN two layers has obtained the lowest performance (Accuracy = 79.84%, Precision = 74.91%, Recall = 80.23% and F1-score = 79.84%). LSTM one layer and LSTM two layers have the same performance.

Overall, **OPCNN-FAKE** has ranked the highest performance for cross-validation results and the testing results compared with other regular ML models, and DL models (RNN and LSTM).

1) The best values of parameters for dataset4

Table 22 shows the best values of parameters'the OPCNN-FAKE model. Table 23 presents the best values parameters for RNN and LSTM.

I. DISCUSSION

1) The best models of detect fake news for dataset1

Figure 4 and Figure 5 illustrate the experimental results in the broad picture for the cross-validation performances and the testing performance for best models, respectively, based on the results acquired in our experiments for dataset1. Overall, When compared to other models, the OPCNN-FAKE model provides the largest cross-validation and testing performance. While, NB with Unigram has achieved the worst cross-validation and the testing performance compared to other models. For cross-validation results, the OPCNN-FAKE model has achieved the highest performance (Accuracy = 99.99%, precision = 100%, recall = 99.97%, and F1-score = 99.97%). NB with Unigram has registered the worst performance (Accuracy = 94.49%, precision = 94.79%, recall = 94.49%, and F1-score = 94.49%). RF with Tri-gram has achieved the second best performance (Accuracy = 98.75%, precision = 98.82%, recall = 98.78%, and F1-score = 98.82%). For testing results, the OPCNN-FAKE model has achieved the highest performance (Accuracy = 97.84%, precision = 97.86%, recall = 97.84%, and F1-score = 97.84%). NB with Unigram has registered the worst performance (Accuracy = 91.57%, Precision = 92.27%, Recall = 91.57%, and F1-score = 91.57%). RF with Tri-gram has achieved the second best performance (Accuracy = 96.97%, precision = 97.04%, recall = 96.97%, and F1-score = 96.97%).

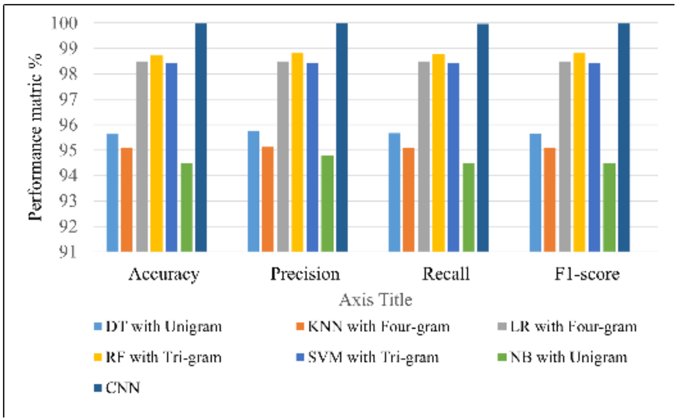


FIGURE 4. The best cross-validation performance models for dataset1.

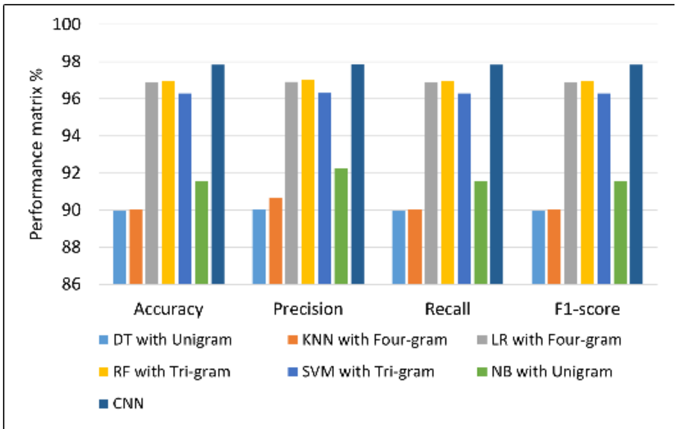


FIGURE 5. The best test performance models for dataset1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

TABLE 20. The performance of regular ML for dataset4

Models	Matrix size	Cross validation performance				Test performance			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
DT	**Unigram**	**99.44±0.11**	**99.43±0.13**	**99.44±0.12**	**99.44±0.12**	**99.22**	**99.22**	**99.22**	**99.22**
	Bi-gram	99.58±0.12	99.59±0.11	99.58±0.11	99.58±0.11	99.40	99.40	99.40	99.40
	Tri-gram	99.48±0.12	99.47±0.12	99.48±0.12	99.48±0.12	99.2	99.2	99.2	99.2
	Four-gram	99.47±0.12	99.47±0.12	99.48±0.12	99.47±0.12	99.25	99.25	99.25	99.25
KNN	**Unigram**	**90.75±0.41**	**90.85±0.4**	**90.75±0.41**	**90.74±0.41**	**89.54**	**89.83**	**89.54**	**89.54**
	Bi-gram	91.31±0.41	91.43±0.41	91.31±0.41	91.3±0.41	90.44	90.65	90.44	90.44
	Tri-gram	91.5±0.4	91.48±0.4	91.61±0.39	91.5±0.4	90.71	90.71	90.91	90.71
	Four-gram	**91.54±0.48**	**91.52±0.48**	**91.63±0.47**	**91.54±0.48**	**90.72**	**90.72**	**90.93**	**90.72**
LR	Unigram	99.3±0.13	99.3±0.13	99.3±0.13	99.3±0.13	98.56	98.56	98.56	98.56
	Bi-gram	99.43±0.13	99.43±0.13	99.43±0.13	99.43±0.13	98.79	98.79	98.79	98.79
	Tri-gram	**99.44±0.15**	**99.44±0.15**	**99.44±0.15**	**99.44±0.15**	**98.8**	**98.8**	**98.8**	**98.8**
	Four-gram	99.43±0.16	99.43±0.16	99.43±0.16	99.43±0.16	98.79	98.79	98.79	98.79
RF	Unigram	99.74±0.08	99.71±0.07	99.7±0.09	99.75±0.05	99.4	99.4	99.4	99.4
	Bi-gram	**99.75±0.04**	**99.77±0.06**	**99.73±0.07**	**99.75±0.07**	**99.90**	**99.90**	**99.90**	**99.90**
	Tri-gram	99.72±0.06	99.75±0.07	99.72±0.07	99.73±0.07	99.48	99.48	99.48	99.48
	Four-gram	99.73±0.07	99.75±0.07	99.77±0.06	99.74±0.05	99.48	99.48	99.48	99.48
SVM	Unigram	99.64±0.08	99.61±0.07	99.6±0.09	99.65±0.05	99.2	99.2	99.2	99.2
	Bi-gram	**99.65±0.04**	**99.64±0.06**	**99.63±0.07**	**99.65±0.07**	**99.40**	**99.40**	**99.40**	**99.40**
	Tri-gram	99.52±0.05	99.55±0.06	99.52±0.06	99.53±0.06	99.28	99.28	99.28	99.28
	Four-gram	**99.23±0.15**	**99.23±0.15**	**99.23±0.15**	**99.23±0.15**	**98.90**	**98.90**	**98.90**	**98.90**
NB	**Unigram**	**92.89±0.36**	**92.9±0.36**	**92.89±0.36**	**92.89±0.36**	**92.9**	**92.9**	**92.9**	**92.9**
	Bi-gram	94.72±0.42	94.72±0.42	94.72±0.42	94.72±0.42	94.67	94.67	94.67	94.67
	Tri-gram	95.05±0.41	95.05±0.41	95.05±0.41	95.05±0.41	94.86	94.86	94.86	94.86
	Four-gram	**95.11±0.4**	**95.11±0.4**	**95.11±0.4**	**95.11±0.4**	**94.94**	**94.94**	**94.94**	**94.94**

TABLE 21. The performance of deep neural networks for dataset3

Models	Cross validation performance				Testing performance			
	Accuracy	F1-score	Precision	Recall	Accuracy	F1-score	Precision	Recall
OPCNN-FAKE	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**99.99**	**99.99**	**99.99**	**99.99**
LSTM one layer	98.87±0.97	98.7±1.16	99.51±0.5	98.09±1.76	99.78	99.78	99.78	99.78
LSTM two layers	99.69±0.86	99.61±1.48	99.89±0.34	99.44±1.75	99.78	99.78	99.79	99.78
RNN one layer	99.46±0.16	99.44±0.17	99.51±0.4	99.37±0.28	99.6	99.6	99.6	99.6
RNN two layers	**79.49±0.57**	**87.89±0.27**	**79.51±0.8**	**98.32±0.63**	**79.84**	**74.91**	**80.23**	**79.84**

TABLE 22. The best values parameters' OPCNN-FAKE for dataset4

Model	Filter size	Kernel size	Max pooling	Dropout	Batch size	epochs
OPCNN-FAKE	22	4	3	0.4	500	66

TABLE 23. The best values parameters for LSTM and RNN for dataset4

Models	Neurons	reg_{rate}	Dropout	Batch size	epochs
LSTM one layer	254	0.1	0.7	400	59
LSTM two layers	[39,41]	[0.4,0.5]	[0.1,0.6]	400	85
RNN one layer	44	0.5	0.2	400	44
RNN two layers	[24,30]	[0.05,0.05]	[0.2,0.7]	500	52

2) The best models of detect fake news for dataset2 (FakeNewsNet)

Figure 6 and Figure 7 illustrate the experimental results in the broad picture for the cross-validation performances and the testing performance for best models, respectively, based on the results acquired in our experiments for FakeNewsNet. Overall,When compared to other models, the OPCNN-FAKE model provides the largest cross-validation and testing performance. For cross-validation results, **OPCNN-FAKE model** has achieved the highest performance (Accuracy = 98.65%, precision = 99.34%, recall = 98.87%, and F1-score = 99.1%). NB with Unigram has registered the worst performance (Accuracy = 86.03%, precision = 85.47%, recall = 86.03%, and F1-score = 85.3%). SVM with Unigram has achieved the second best performance (Accuracy = 93.85%, precision = 93.89%, recall = 93.85%, and F1-score = 93.64%). For testing results, **OPCNN-FAKE model** has achieved the highest performance (Accuracy = 95.26%, precision = 95.28%, recall = 95.26%, and F1-score = 95.27%). DT with Four-gram has registered the worst performance (Accuracy = 81.8%, precision = 81.47%, recall = 81.8%, and F1-score = 81.62%). RF with Unigram has achieved the second best performance (Accuracy = 85.34%, precision = 84.8%, recall = 85.34%, and F1-score = 84.2%).

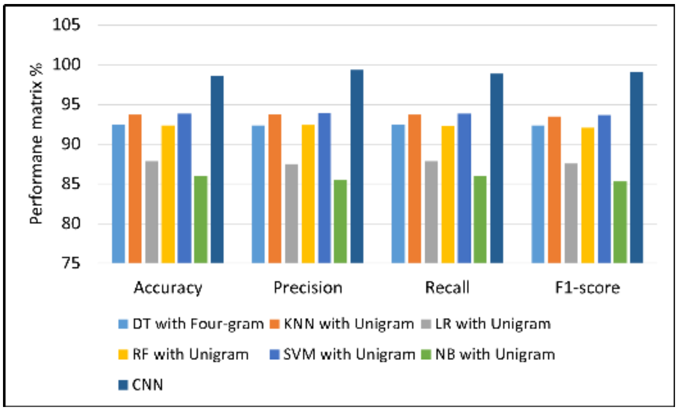


FIGURE 6. The best cross-validation performance models for dataset2 (FakeNewsNet).

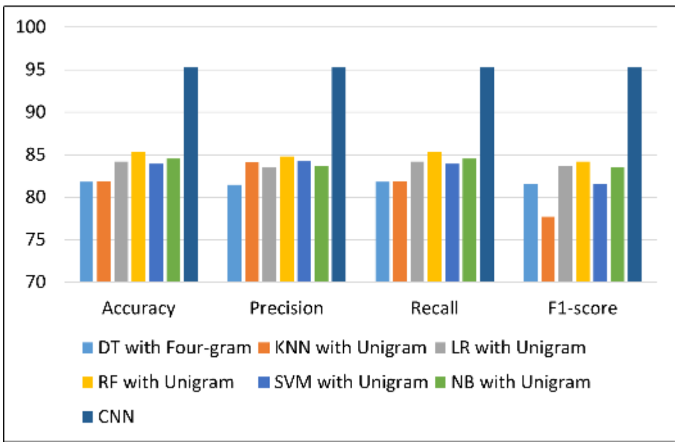


FIGURE 7. The best test performance models for dataset2.

3) The best models of detect fake news for dataset3

Figure 8 and Figure 9 illustrate the experimental results in the broad picture for the cross-validation performances and the testing performance for best models, respectively, based on the results acquired in our experiments for datase3. Overall, When compared to other models, the OPCNN-FAKE model provides the largest cross-validation and testing performance. For cross-validation results, OPCNN-FAKE model has achieved the highest performance (Accuracy = 97.23%, precision = 97.37%, recall = 97.58%, and F1-score = 97.26%). DT with Unigram has registered the worst performance (Accuracy = 51.02%, precision = 51.11%, recall = 51.01%, and F1-score = 51.39%). NB with Unigram has achieved the second best performance (Accuracy = 53.37%, precision = 53.93%, recall = 53.93%, and F1-score = 53.7%). For testing results, the OPCNN-FAKE model has achieved the highest performance (Accuracy = 53.99%, precision = 53.86%, recall = 53.91%, and F1-score = 53.99%). DT with Four-gram has registered the worst performance (Accuracy = 48.76%, precision = 48.84%, recall = 48.76%, and F1-score = 48.72%).

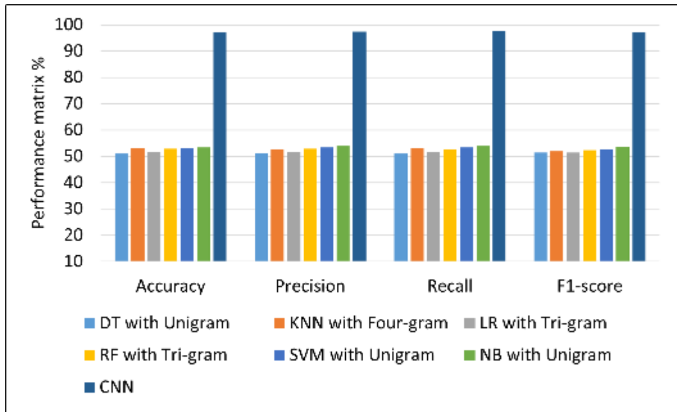


FIGURE 8. The best cross-validation performance models for dataset3.

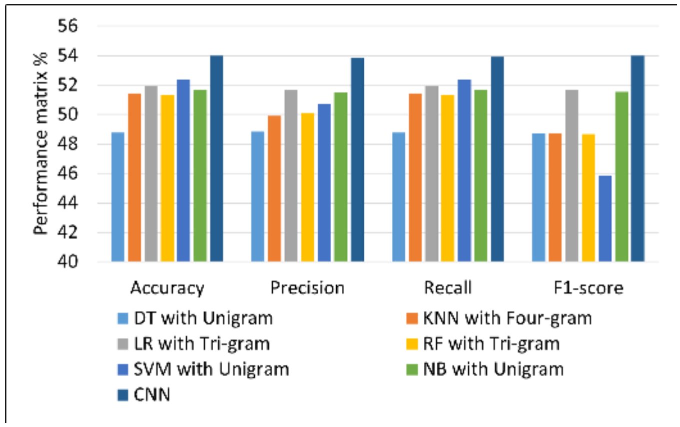


FIGURE 9. The best test performance models for dataset3.

4) The best models of detect fake news for dataset4

Figure 10 and Figure 11 illustrate the experimental results in the broad picture for the cross-validation performances and the testing performance for best models, respectively, based on the results acquired in our experiments for datase4. Overall, When compared to other models, the OPCNN-FAKE model provides the largest cross-validation and testing performance. For cross-validation results, OPCNN-FAKE model has achieved the highest performance (Accuracy = 100%, precision = 100%, recall = 100%, and F1-score = 100%). KNN with Four-gram has registered the worst performance (Accuracy = 91.54%, precision = 91.52%, recall = 91.63%, and F1-score = 91.54%). RF with Tri-gram has achieved the second best performance (Accuracy = 99.75%, precision = 99.74%, recall = 99.73%, and F1-score = 99.75%). For testing results, the OPCNN-FAKE model has achieved the highest performance (Accuracy = 99.99%, precision = 99.99%, recall = 99.99%, and F1-score = 99.99%). DT with Four-gram has registered the worst performance (Accuracy = 90.72%, precision = 90.72%, recall = 90.93%, and F1-score = 90.72%). RF with Tri-gram has achieved the second best performance (Accuracy = 99.9%, precision = 99.9%, recall = 99.9%, and F1-score = 99.9%).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

Briefly, the proposed OPCNN-FAKE model has a higher performance than the other models based on Accuracy, Precision, Recall, f1-score. Furthermore, it indicates that the OPCNN-FAKE model for fake news detection performance is significantly better than other existing works that were used methods based on CNN. For instance, the authors of [27] used dataset from Kaggle and accuracy was 97.5% and in [28] performance for CNN only was (Accuracy of 91.50%, Precision = 90.74%, Recall = 92.07, F1-Score = 91.40)and performnce for the FNDNet was (Accuracy of 98.36%, Precision =99.40, Recall =96.88, F1-Score =98.12), while OPCNN-Fake achieved the highest performance for cross-validation result (Accuracy = 99.99%, Precision = 100%, Recall = 99.97%, and F1-score = 99.97%), and for the testing result (Accuracy = 97.84%, Precision = 97.86%, Recall = 97.84%, and F1-score = 97.84%).

In case of FakeNewsNet, the preference in [15] was Accuracy= 62.9% and F1-score =58.3%, while, OPCNN-Fake performance for cross-validation result was (Accuracy = 98.65%, and F1-score = 99.1%), and performance for tesing result was (Accuracy = 95.26%, and F1-score = 95.27%). Furthermore, FA-KES5 used by [29] and the performance was hybrid CNN-RNN Accuracy = 60% of training set,while OPCNN-Fake performence for cross-validation result (Accuracy = 97.23). In the case of ISOT, the performance in [29] was Accuracy =100% of the training set, but OPCNN-Fake performance for cross-validation result was Accuracy = 100%. Briefly, Table. 25 illustrates the difference between the OPCNN-FAKE and existing work based on dataset and performance.

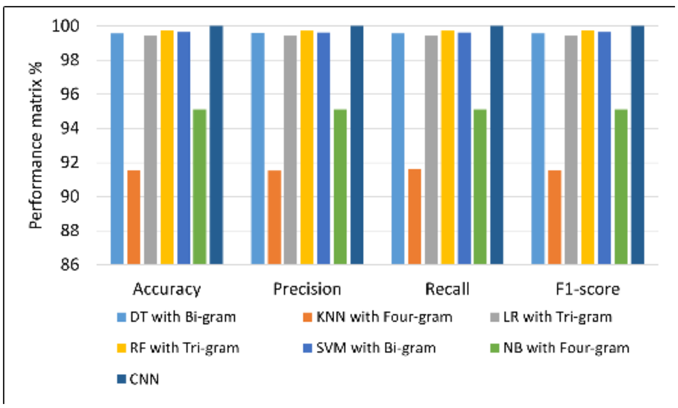


FIGURE 10. The best cross-validation performance models for dataset4.

V. CONCLUSION

This paper has introduced a fake news detection system using two approaches, regular ML and DL. In DL, we proposed the OPCNN-FAKE model that has achieved the best performance. The proposed OPCNN-FAKE model consists of six layers: an embedding layer, dropout layer, a convolutional layer, a pooling layer, flatten layer, and an output layer. Also, it has optimized using hyperopt optimization technique, the different values of parameters for each layer have been adapted, and the best values that achieved the best performance have been selected. Also, n-gram with TF-ID and word embedding feature extraction methods have been used for ML and DL, respectively. We compared the OPCNN-FAKE with RNN, LSTM, and the six regular ML techniques: DT, LR, KNN, RF, SVM, NB using four fake news benchmark datasets. Each dataset has split into 80% training dataset and 20% testing dataset. Training datasets have been used to optimize, train models, and test datasets to evaluate models. Also, cross-validation and testing results have been registered that show the OPCNN-FAKE model has achieved the best performance for each dataset compared with other models. For dataset1, the OPCNN-FAKE model achieved the best performance for testing result (Accuracy = 97.84%, precision = 97.86%, recall = 97.84%, and F1-score = 97.84%). For dataset2 (FakeNewsNet), For testing results, OPCNN-FAKE model has achieved the highest performance for testing results (Accuracy = 95.26%, precision = 95.28%, recall = 95.26%, and F1-score = 95.27%). For dataset3, OPCNN-FAKE model has achieved the highest performance for testing results (Accuracy = 53.99%, precision = 53.86%, recall = 53.91%, and F1-score = 53.99%). For dataset4, the OPCNN-FAKE model has achieved the highest performance For testing results (Accuracy = 99.99%, precision = 99.99%, recall = 99.99%, and F1-score = 99.99%). In future, we will use our proposed model to detect COVID-19 fake news. Also, we plan to apply multimodel-based methods with recently pre-trained word embeddings (i.e., Elmo, XLNet, etc.) to handle visual information like video and images. In addition, we may use knowledge-based and fact-based approaches to detect fake news. We will also expand our planned dataset to include data from additional languages.

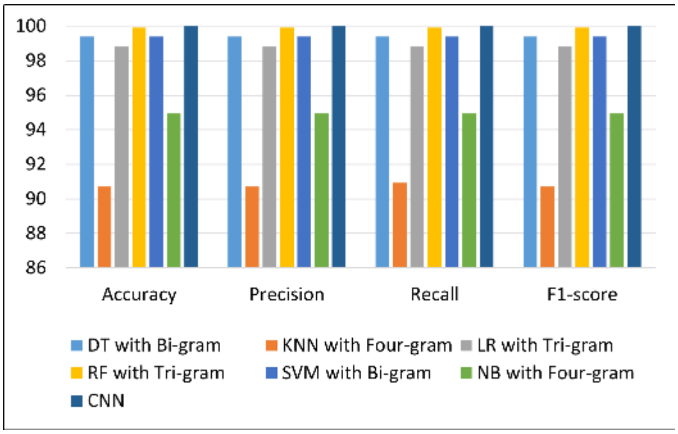


FIGURE 11. The best test performance models for dataset4

ACKNOWLEDGMENT

This research was supported by Taif University Researchers Supporting Project number (TURSP- 2020/231), Taif University, Taif, Saudi Arabia

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

TABLE 24. Comparisons results with existing work

Ref	The number of dataset	Dataset name	Models' performance
[33] (2020)	One dataset	FakeNewsNet (PolitiFact topic)	CNN Accuracy= 62.9% and F1-score =58.3%
		FakeNewsNet (GossipCop topic)	CNN Accuracy= 72.3% and F1-score =72.5%
[27] (2020)	One dataset	The dataset from Kaggle	hybrid CNN-LSTM Accuracy =97.5%
[29] (2021)	Two datasets	ISOT dataset	hybrid CNN-RNN Accuracy =100% of training set
		FA-KES dataset	hybrid CNN-RNN Accuracy = 60% of training set
[28] (2020)	One dataset	The dataset from Kaggle	The CNN only (Accuracy of 91.50%, Precision = 90.74%, Recall = 92.07%, F1-Score = 91.40%) The FNDNet (deeper CNN) model (Accuracy of 98.36%, Precision =99.40%, Recall =96.88%, F1-Score =98.12%)
[30] (2019)	Two dataset	Weibo	Accuracy = 88.82%
		NewsFN	Accuracy=90.10%
Our work	Four datasets	Dataset from Kaggle (dataset1)	The OPCNN-FAKE model Cross-validation result (Accuracy = 99.99%, Precision = 100%, Recall = 99.97%, and F1-score = 99.97%). Testing set (Accuracy = 97.84%, Precision = 97.86%, Recall = 97.84%, and F1-score = 97.84%).
		FakeNewsNet	The OPCNN-FAKE model Cross-validation result (Accuracy = 98.65%, Precision = 99.34%, Recall = 98.87%, and F1-score = 99.1%). Testing result (Accuracy = 95.26%, Precision = 95.28%, Recall = 95.26%, and F1-score = 95.27%).
		FA-KES5	The OPCNN-FAKE model Cross-validation result (Accuracy = 97.23%, Precision = 97.37%, Recall = 97.58%, and F1-score = 97.26%) Testing result (Accuracy = 53.99%, Precision = 53.86%, Recall = 53.91%, and F1-score = 53.99%).
		ISOT	OPCNN-FAKE Cross-validation result (Accuracy = 100%, Precision = 100%, Recall = 100%, and F1-score = 100%). Testing result (Accuracy = 99.99%, Precision = 99.99%, Recall = 99.99%, and F1-score = 99.99%).

REFERENCES

[1] M. H. Goldani, R. Safabakhsh, and S. Momtazi, "Convolutional neural network with margin loss for fake news detection," Information Processing & Management, vol. 58, no. 1, p. 102418, 2021.

[2] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," Information Sciences, vol. 497, pp. 38–55, 2019.

[3] "Can 'fake news' impact the stock market?"

[4] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," Journal of economic perspectives, vol. 31, no. 2, pp. 211–36, 2017.

[5] L. Bozarth and C. Budak, "Toward a better performance evaluation framework for fake news classification," in Proceedings of the international AAAI conference on web and social media, vol. 14, 2020, pp. 60–71.

[6] Y. Liu and Y.-F. B. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in Thirty-second AAAI conference on artificial intelligence, 2018.

[7] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," arXiv preprint arXiv:1902.06673, 2019.

[8] S. Girgis, E. Amer, and M. Gadallah, "Deep learning algorithms for detecting fake news in online text," in 2018 13th International Conference on Computer Engineering and Systems (ICCES). IEEE, 2018, pp. 93–97.

[9] S. Ghosh and C. Shah, "Towards automatic fake news classification," Proceedings of the Association for Information Science and Technology, vol. 55, no. 1, pp. 805–807, 2018.

[10] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Insights into imaging, vol. 9, no. 4, pp. 611–629, 2018.

[11] Y. Yu, C. Wang, X. Gu, and J. Li, "A novel deep learning-based method for damage identification of smart building structures," Structural Health Monitoring, vol. 18, no. 1, pp. 143–163, 2019.

[12] H. Li, "Deep learning for natural language processing: advantages and challenges," National Science Review, 2017.

[13] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," 2016.

[14] N. Ruchansky, S. Seo, and Y. Liu, "Csi: A hybrid deep model for fake news detection," in Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 797–806.

[15] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media," arXiv preprint arXiv:1809.01286, 2018.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3112806, IEEE Access

[16] F. K. A. Salem, R. Al Feel, S. Elbassuoni, M. Jaber, and M. Farah, "Fakes: A fake news dataset around the syrian war," in Proceedings of the International AAAI Conference on Web and Social Media, vol. 13, 2019, pp. 573–582.
[17] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, "Declare: Debunking fake news and false claims using evidence-aware deep learning," arXiv preprint arXiv:1809.06416, 2018.
[18] P. Ksieniewicz, M. Choraś, R. Kozik, and M. Woźniak, "Machine learning methods for fake news classification," in International Conference on Intelligent Data Engineering and Automated Learning. Springer, 2019, pp. 332–339.
[19] J. P. Singh, A. Kumar, N. P. Rana, and Y. K. Dwivedi, "Attention-based lstm network for rumor veracity estimation of tweets," Information Systems Frontiers, pp. 1–16, 2020.
[20] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using n-gram analysis and machine learning techniques," in International conference on intelligent, secure, and dependable systems in distributed and cloud environments. Springer, 2017, pp. 127–138.
[21] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," arXiv preprint arXiv:1708.07104, 2017.
[22] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in 2014 13th international conference on control automation robotics & vision (ICARCV). IEEE, 2014, pp. 844–848.
[23] M. Khalil-Hani and L. S. Sung, "A convolutional neural network approach for face verification," in 2014 International Conference on High Performance Computing & Simulation (HPCS). IEEE, 2014, pp. 707–714.
[24] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3367–3375.
[25] Y. Goldberg, "Neural network methods for natural language processing," Synthesis lectures on human language technologies, vol. 10, no. 1, pp. 1–309, 2017.
[26] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu, "Ti-cnn: Convolutional neural networks for fake news detection," arXiv preprint arXiv:1806.00749, 2018.
[27] A. Abdullah, M. Awan, M. Shehzad, and M. Ashraf, "Fake news classification bimodal using convolutional neural network and long short-term memory," Int. J. Emerg. Technol. Learn, vol. 11, pp. 209–212, 2020.
[28] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, "Fndnet–a deep convolutional neural network for fake news detection," Cognitive Systems Research, vol. 61, pp. 32–44, 2020.
[29] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid cnn-rnn based deep learning approach," International Journal of Information Management Data Insights, vol. 1, no. 1, p. 100007, 2021.
[30] Q. Li, Q. Hu, Y. Lu, Y. Yang, and J. Cheng, "Multi-level word features based on cnn for fake news detection in cultural communication," Personal and Ubiquitous Computing, pp. 1–14, 2019.
[31] R. K. Kaliyar, A. Goswami, and P. Narang, "Deepfake: improving fake news detection using tensor decomposition-based deep neural network," The Journal of Supercomputing, vol. 77, no. 2, pp. 1015–1037, 2021.
[32] ——, "Echofaked: improving fake news detection in social media with an efficient deep neural network," Neural computing and applications, pp. 1–17, 2021.
[33] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," Big data, vol. 8, no. 3, pp. 171–188, 2020.
[34] "Fake news detection."
[35] F. K. A. Salem, R. Al Feel, S. Elbassuoni, M. Jaber, and M. Farah, "Fakes: A fake news dataset around the syrian war," in Proceedings of the International AAAI Conference on Web and Social Media, vol. 13, 2019, pp. 573–582.
[36] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
[37] F. Chollet et al., "Keras: The python deep learning library," Astrophysics Source Code Library, pp. ascl–1806, 2018.
[38] B. Zhong, X. Xing, P. Love, X. Wang, and H. Luo, "Convolutional neural network: Deep learning-based classification of building quality problems," Advanced Engineering Informatics, vol. 40, pp. 46–57, 2019.
[39] A. F. Agarap, "Deep learning using rectified linear units (relu)," arXiv preprint arXiv:1803.08375, 2018.
[40] Z. Zhang, "Improved adam optimizer for deep neural networks," in 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). IEEE, 2018, pp. 1–2.
[41] A. Wanto, A. P. Windarto, D. Hartama, and I. Parlina, "Use of binary sigmoid function and linear identity in artificial neural networks for forecasting population density," IJISTECH (International Journal of Information System & Technology), vol. 1, no. 1, pp. 43–54, 2017.
[42] J. Bergstra, D. Yamins, and D. Cox, "Hyperopt: Distributed asynchronous hyper-parameter optimization," Retrieved May, vol. 21, p. 2020, 2012.
[43] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," Physica D: Nonlinear Phenomena, vol. 404, p. 132306, 2020.
[44] A. Graves, "Long short-term memory," in Supervised sequence labelling with recurrent neural networks. Springer, 2012, pp. 37–45.
[45] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in Thirteenth annual conference of the international speech communication association, 2012.
[46] T. Van Laarhoven, "L2 regularization versus batch and weight normalization," arXiv preprint arXiv:1706.05350, 2017.
[47] A. Sikandar, W. Anwar, U. I. Bajwa, X. Wang, M. Sikandar, L. Yao, Z. L. Jiang, and Z. Chunkai, "Decision tree based approaches for detecting protein complex in protein protein interaction network (ppi) via link and sequence analysis," IEEE Access, vol. 6, pp. 22 108–22 120, 2018.
[48] J. H. Cheon, D. Kim, Y. Kim, and Y. Song, "Ensemble method for privacy-preserving logistic regression based on homomorphic encryption," IEEE Access, vol. 6, pp. 46 938–46 948, 2018.
[49] Y. Guo, H. Cao, S. Han, Y. Sun, and Y. Bai, "Spectral–spatial hyperspectralimage classification with k-nearest neighbor and guided filter," IEEE Access, vol. 6, pp. 18 582–18 591, 2018.
[50] A. Chakraborty, S. Mitra, D. De, A. J. Pal, F. Ghaemi, A. Ahmadian, and M. Ferrara, "Determining protein-protein interaction using support vector machine: A review," IEEE Access, 2021.
[51] L. Li, Y. Zhang, W. Chen, S. K. Bose, M. Zukerman, and G. Shen, "Naïve bayes classifier-assisted least loaded routing for circuit-switched networks," IEEE Access, vol. 7, pp. 11 854–11 867, 2019.