OptNet-Fake: Fake News Detection in Socio-cyber platforms using Grasshopper Optimization and Deep Neural Network

Sanjay Kumar, Akshi Kumar, Abhishek Mallik, and Rishi Ranjan Singh

***Abstract*—Exposure to half-truths or lies has the potential to undermine democracies, polarise public opinion, and promote violent extremism. Identifying the veracity of fake news is a challenging task in distributed and disparate cyber-socio platforms. To enhance the trustworthiness of news on these platforms, in this paper, we put forward a fake news detection model, *OptNet-Fake*. The proposed model is architecturally a hybrid that uses a meta-heuristic algorithm to select features based on usefulness and trains a deep neural network to detect fake news in social media. d dimensional feature vectors for the textual data are initially extracted using the Term Frequency Inverse Document Frequency (TF-IDF) weighting technique. The extracted features are then directed to a modified grasshopper optimization (MGO) algorithm, which selects the most salient features in the text. The selected features are then fed to various convolutional neural networks (CNNs) with different filter sizes to process them and obtain the n-gram features from the text. These extracted features are finally concatenated for the detection of fake news. The results are evaluated for four real-world fake-news datasets using standard evaluation metrics. A comparison with different meta-heuristic algorithms and recent fake news detection methods is also done. The results distinctly endorse the superior performance of the proposed *OptNet-Fake* model over contemporary models across various datasets.**

***Index Terms*—Convolutional Neural Network, Fake News Detection, TF-IDF, Grasshopper Optimization Algorithm, Feature Selection**

I. INTRODUCTION

Owing to the low-cost internet-enabled devices with easy and anytime access to the Web, the use of social media platforms like Facebook, Twitter, Instagram, WhatsApp, etc. has grown fast and profound. The current global statistics reveal an active social media user pool of 4.55 billion [1]. While the amount of user-generated content is proliferating, the speed of diffusion is unusually striking, thus creating a quintessential 'breeding ground' for posting and disseminating antagonistic content, which includes fake news, rumors, and offensive, hateful, and bullying content. Fake news stories have been afflicting countries globally. Some online information is blatantly fake or misleading, and some stories are subtly wrong. Fake news is not new, but new communication technologies such as social media have led to the propagation of fake news. The recent pandemic is a witness to this 'contamination' of information, where unconfirmed and falsified news on proven prevention, cures, and medication is being circulated, putting lives at risk. The World Health Organisation (WHO) has called the spread of fake news about COVID-19 an "infodemic." Countering falsehood and fact-checking the avalanche of information is a conscientious and time-consuming process, whereas masquerading (purposely creating false accounts) or automated bots can plague the digital media with alarming speed. Moreover, fake news could be in multiple forms, such as rumors, satire, false advertisements, cyber-bullying, hate speech, etc. [2]. Such false or manipulated information not only creates a sense of distrust in online news and communication but also has a significant impact on our opinion and may lead to distrust and unrest in society. However, evidence has shown that debunking can be effective if delivered correctly, and importantly before it accelerates the polarization process, creating a fake news panic, causing harm to health, and inflaming the social conflict.

Usually, fake information can be of two types- misinformation and disinformation. Misinformation is factually incorrect facts. However, disinformation is false information that is circulated to mislead the public leading to economic, political, and social impacts. The information is manipulated so that the reader feels it is correct. Fake news designers utilize social engineering and deception techniques on social media platforms and influence users' behaviors by persuading them, for example, to click on web links. The deception techniques involve a psychological process; in this way, fake news creators intend to achieve financial benefits. According to Kshetri and Voas citekshetri2017economics, someone will engage in the creation and spreading of fake news based on the following mathematical equation.

$$M_b + P_b > I_c + O_{1c} + P_c + (O_{2c} \times \pi_{arr} \times \pi_{con}) \qquad (1)$$

Here, the monetary and psychological gains or benefits reaped by fraudsters are represented by M_b and P_b, respectively. The direct investments, opportunity costs, and psychological costs associated with creating and managing fake news are represented by I_c, O_{1c}, and P_c, respectively. Also, O_{2c}, π_{arr}, and π_{con} represent the monetary costs of conviction, probability of arrest, and probability of conviction, respectively.

Manuscript received-; revised- .

Abhishek Mallik and Sanjay Kumar are with the Department of Computer Science and Engineering, Delhi Technological University, New Delhi-110042, India, Email:- sanjay.kumar@dtu.ac.in abhishekmallik265@gmail.com
Akshi Kumar is with the Department of Computing and Mathematics, Faculty of Science and Engineering, Manchester Metropolitan University, Manchester, United Kingdom. Email:- akshi.kumar@mmu.ac.uk
Rishi Ranjan Singh is with the Department of Electrical Engineering and Computer Science, Indian Institute of Technology, Bhilai, India, Email:- rishi@iitbhilai.ac.in
Corresponding author: Akshi Kumar (akshi.kumar@mmu.ac.uk)

Fake news detection is a challenging task as multilingual, multimodal social media acts as an amplifier and rapid distribution channel in this process. The extensive number of fake news spreaders further complicates the process of debunking. Further, the constant adaptation of the design and presentation of content such that it appears more legitimate makes it nearly impossible to halt this widespread phenomenon in the socio-cyber world [3]. At the same time, not all users can discern fake from actual news. Moreover, anyone with internet access can easily create malicious accounts, such as cyborg users, social bots, etc., leading to increased fake news. This makes it impossible to assess the never-ending online data manually. To minimize the circulation of fake news, several fact-checking projects such as the Google News Initiative, Verizon, etc. have been introduced.

In general, fake news detection tasks can be accomplished with the help of feature extraction followed by model building. Few of the commonly used methods to detect fake news include content-based identification, feedback-based identification, and intervention-based solutions [4]. Most of the existing fake news detection methods use textual content, user responses, and computational methods [3]. However, these methods have several limitations, such as reduced feature relevance, redundant features, and increased feature correlation that lead to a low detection rate and accuracy of the model. Moreover, as the number of features increases, the problem of variable selection becomes harder, thereby leading to the problems of the 'curse of dimensionality and model overfitting. Further, the deep learning models achieve state-of-the-art results in various natural language processing (NLP) tasks, including fake news detection, but often suffer from the vanishing gradient problem and take longer training times to achieve a considerable quality of results. All these limitations motivated us to put forward a novel model, *FakeScan* for fake news detection using text-based features.

The proposed *FakeScan* model is built by pairing the modified grasshopper optimization (MGO) meta-heuristic algorithm with a convolutional neural network (CNN) architecture. We start by preprocessing the textual data (news articles and social media posts) to achieve uniformity across the dataset. To create the initial feature matrix, we extract features using the TF-IDF technique. This gives a d dimensional feature set for every news article represented in a d dimensional feature space. The extracted features are then passed to the MGO algorithm, which selects the most relevant features from the generated feature space. As the classical grasshopper optimization is suitable for continuous tasks, hence we modify it to work on discrete tasks like feature selection [5]. The selected features are then fed to a deep convolutional neural network to process them and obtain the n-gram features from the text, which are further used to perform the final fake news detection. We examine the performance of the proposed *FakeScan* model on four benchmark real-world datasets, namely, Kaggle Fake News Dataset [6], ISOT Fake News Dataset [7], [8], COVID-19 Fake News Dataset [9], and WELFake Dataset [10] and evaluate several performance metrics for the task of fake news detection. The results of the MGO-CNN based *FakeScan* model are compared with several contemporary methods of fake news detection to obtain a comparative performance and utility of the introduced model. The main contribution of our work can be summarized as follows:

(i) We propose a novel fake news detection model, *FakeScan* using a Modified Grasshopper Optimization (MGO) and convolutional neural network (CNN).
(ii) We adopt a feature generation technique using TF-IDF to exploit the importance of occurrences of words in a text segment and across a text corpora.
(iii) We modify the classical Grasshopper Optimization algorithm for feature selection to capture the most relevant and the least correlated features for fake news detection.
(iv) The proposed work utilizes a deep CNN architecture to extract the n-gram features to better characterize the news articles.
(v) A comparative study involving various metaheuristics and contemporary fake news detection methods has been performed using four benchmarked real-world datasets.

The rest of the paper is organised as follows. Section II presents a discussion on the already existing work in the field of fake news detection. The details about the preliminary concepts required for better understanding this paper is described in Section III. Various datasets and evaluation metrics used by us for our experimental study is described in Section IV. We illustrate our proposed work in detail in Section V. The experimental results and analysis is presented in Section VI. Finally, the concluding remarks are mentioned in Section VII.

II. Related Work

The proliferation of fake news on online social networks has gained much attention in the literature, and an increasing number of researchers have been exploring this field. We can broadly divide various fake news detection methods into three groups, namely, Knowledge-graph based, Linguistic based, and Machine-learning and deep-learning based techniques [11]. The knowledge graph-based approach analyses network behavior and structure to bring out false news. This can be carried out in multiple ways, including knowledge graph analysis, which has an accuracy of 61% to 95%, as claimed by Ciampaglia et al. [12]. They also studied the relationship between entities and put forward the theory of 'network effect' variables. Another graph-based approach was used by Gangireddy et al. [13] in their study, where they identified misinformation with the help of label spreading, biclique identification, and feature vector learning. Their approach comprises three phases and achieves an accuracy near 80% for unsupervised detection of fake news. Notable work was also presented by Shu et al. [14], who used network influence minimization methods along with network estimation to assess as well as mitigate the effect of fake news.

Linguistic-based techniques analyze the news content and detect fake news with the help of differences in language, writing style, and sentiment of the text. Yang et al. [15] explored the source user's characteristics on Sina Weibo, China's popular social media platform. They examined a set of features and put forward a classifier to bring out false information. In genereal, the linguistic analysis methods are based on the n-gram Approach, Part-of-Speech Tags, and Probabilistic Context-Free Grammar [4]. The n-gram approach uses patterns of n continuous words within a text, consisting of words and phrases. Syntactic features like Part-of-Speech tags are acquired by tagging every word according to a syntactic feature such as adjectives and nouns. The Probabilistic Context-Free Grammar (PCFG) uses a CFG to denote a sentence's grammatical structure. The intermediate and terminal nodes represent syntactic constituents and words, respectively. Except for these three categories, some researchers determined a hierarchical structure among documents and used its syntax to bring out fake news [16]. While others, such as Karimi et al. [17], used a discourse-level structure to do the same. However, only linguistic features may not be enough to discern fake from actual news, and hence, these techniques are often merged with machine learning techniques.

Liu et al. [18] performed a study distinguishing fake from real stories based on machine learning techniques. They did this by classifying news propagation paths and then applying both convolutional and recurrent networks to obtain the differences in user characteristics along these paths. Reis et al. tested the effectiveness of multiple supervised learning classifiers when detecting fake news from recent datasets [19]. Zhang et al. [20] used supervised machine learning for effective and efficient spammer detection. They did this by collecting a dataset, classifying the users as spammers and non-spammers, and then using an SVM-based spammer detection algorithm.

Karimi et al. [21] used a set of Long Short-Term Memory's (LSTM) for multi-class and multi-source fake news detection to discover the numerous degrees of fake news. Wu et al. [22] presumed that intentionally spread false information is often manipulated to seem authentic. To bring out this falsified news, they used embeddings along with a combination of Long Short-Term Memory(LSTM) and Recurrent Neural Networks (RNN) to build a classifier based on propagation pathways in social media. A Convolution Neural Network (CNN) based study was given by Yang et al. [23]. They presented a model called TI-CNN, using latent and explicit features to analyze texts and images for incorrect information. Paka et al. [24] proposed a novel framework named Cross-SEAN for fake tweets detection related to Covid-19. They introduced CTF, a large labeled Fake Tweets dataset. As part of Cross-SEAN, they proposed a cross-stitch-based semi- supervised end-to-end neural attention model. It is a semi-supervised approach and exploits a large amount of unlabelled data. They used a combination of word embeddings, BiLSTM, Attention neural networks, and the BERT framework. They also presented a Google Chrome extension named Chrome-SEAN. Sahoo et al. [25] proposed a Google Chrome-centric automated fake news detection based approach. They devised their approach to work on Facebook. They leveraged various features associated with a user's Facebook account and a deep learning-based analyzer to analyze the news content features. Trueman et al. [26] proposed an approach for fake news detection and its classification into six sub-categories. They presented an attention based Convolutional Bi-directional LSTM framework.

III. Preliminaries

This section describes the text vectorizer technique, namely TF-IDF and evolutionary computing-based technique, Grasshopper Optimization Algorithm used in our paper.

A. TF-IDF

TF-IDF stands for *Term Frequency - Inverse Document Frequency*. It is a methodology to quantify the occurence of words in a text document. Term Frequency is used to measure that how many times a term is present in a document. It is often the case that the frequency of a particular word in a large text is more than a smaller text. To rectify this issue, the occurrence of any term in a document is divided by the total number of terms present in that document. Hence the Term Frequency is given by Eq. 2 as follows:

$$TF = \frac{Number\ of\ repetitions\ of\ word\ in\ a\ document}{Number\ of\ words\ in\ a\ document} \tag{2}$$

When the term frequency of a document is calculated, it can be observed that the algorithm treats all keywords equally, it doesn't matter if it is a stop word like "of," which is incorrect. All keywords have different importance. Let's say the stop word "of" is present in a document 2000 times, but it is of no use or has very little significance. That is exactly what IDF is for. The inverse document frequency assigns lower weight to frequent words and assigns greater weight for the infrequent words. For example, we have ten documents and the term "technology" is present in 5 of those documents so that the inverse document frequency can be calculated as Eq. 3

$$IDF = \log \frac{Number\ of\ documents}{Number\ of\ documents\ containing\ the\ word} \tag{3}$$

So finally, TF-IDF is calculated as follows using Eq. 4:

$$TF - IDF = TF * IDF \tag{4}$$

B. Grasshopper Optimization Algorithm

Grasshopper Optimization Algorithm (GOA) is a recent swarm intelligence algorithm proposed by Saremi et al. [27] which mimics the grasshoppers' foraging and swarming behavior. The life cycle of grasshoppers usually has two phases: the nymph phase and the adult phase. The nymph phase includes small steps and slow movements, while the adult phase includes long-range and abrupt movements. The nymph and adult phases of life of a grasshopper create the intensification and diversification of the Grasshopper Optimization Algorithm. The social interaction amongst the grasshoppers is defined by attraction and repulsion amongst them. The distance is considered in the range [0,15]. The attraction between the grasshoppers increases from 2.079 to 4, and after that, it decreases gradually. While for a distance less than 2.079, the repulsion occurs. There is no attraction or repulsion between the grasshoppers at exactly 2.079 units of the distance between them. This area is called the comfort zone and is represented by Fig. 1.

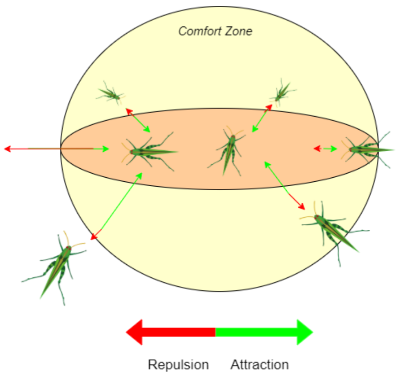

Fig. 1. Interaction between grasshoppers with respect to the comfort area.

The position of every grasshopper is updated based on its current position, best position globally, and the position of other grasshoppers within the swarm. This helps Grasshopper Optimization Algorithm to avoid better being trapped in local optima. Fig. 2 represents the algorithmic flowchart for a typical Grasshopper Optimization Algorithm. It can be seen that firstly we initialize a population of grasshoppers. Then we evaluate the fitness value for every grasshopper using the fitness function. Then the fittest grasshopper is selected from the population and the positions of the grasshoppers are updated as per the fittest grasshopper following the updation criteria. This process is iteratively repeated multiple times until a termination condition is satisfied and the fittest grasshopper forms the required solution to the optimization problem. The source code for the Grasshopper Optimization algorithm can be found at:
http://www.alimirjalili.com/GOA.html.

IV. Dataset and Evaluation Metrics

This section describes about the various datasets used in the study and the evaluation metrics used to gather the results.

A. Dataset

We have used various types of datasets for our study, and these include Kaggle Fake News Dataset [6], ISOT Fake News Dataset [7], [8], COVID-19 Fake News Dataset [9], and WELFake Dataset [10]. Each dataset represents a different context and is suitable for our study. The composition of fake articles and real articles in the used datasets is given in Table I.

TABLE I
Composition of fake and real articles in the used datasets

Dataset	Real	Fake
Kaggle Fake News	10413	10387
ISOT Fake News	21417	23481
COVID-19 Fake News	5600	5100
WELFake	35028	37106

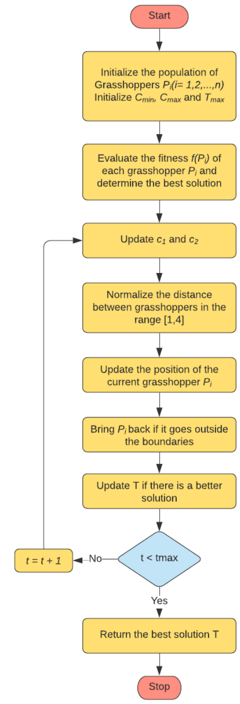

Fig. 2. Algorithmic flowchart of the Grasshopper Optimization algorithm.

1) Kaggle Fake News: The Kaggle dataset contains both reliable and unreliable articles that context the 2016 US presidential elections. It includes 20800 IDs, 20242 titles, 18843 authors, 20671 texts, and 20800 labels. Attributes & Number of Instances in the Kaggle Fake News Dataset is given in Table II.

TABLE II
Attributes & Number of Instances in the Kaggle Fake News Dataset

Attribute	Number of instances in the dataset
ID	20800
Title	20242
Author	18843
text	20761
label	20800

2) ISOT Fake News: The dataset [7], [8] contains fake and real news articles. The truthful articles were obtained from Reuters.com, and the fake articles were collected from various websites indicated by Politifact. The data set contains articles related to political news, world news, government news, and regional news of the United States and the Middle East. The type and size of every article per category for the ISOT dataset are given in Table III.

3) COVID-19 Fake News: The dataset contains 10,700 social media posts based on COVID-19 news [9] collected from various platforms like Facebook, Twitter, Instagram, Politifact, World Health Organisation (WHO), Indian Council of Medical Research (ICMR), Centers for Disease Control and Prevention (CDC), etc. It contains 5600 real and 5100 fake

TABLE III
TYPE AND SIZE OF EVERY ARTICLE PER CATEGORY FOR ISOT DATASET PROVIDED BY AHMED ET AL. [7], [8]

News	Size (Number of articles)	Subjects	
Real-News	21417	**Type**	**Articles Size**
		World-News	10145
		Political-News	11272
Fake-News	23481	**Type**	**Articles Size**
		Government-News	1570
		Middle-East	778
		US News	783
		Left News	4459
		Politics	6841
		News	9050

news articles. Numerical features of the Covid Fake News dataset are given in Table IV.

TABLE IV
NUMERICAL FEATURES OF THE COVID FAKE NEWS DATASET PROVIDED BY PATWA ET AL. [9]

Attribite	Fake	Real	Combined
Unique Words	19728	22916	37503
Avg words per post	21.65	31.97	27.05
Avg chars per post	143.26	218.37	182.57

4) WELFake: Word Embedding over Linguistic Features for Fake News Detection (WELFake) dataset [10] contains 72134 news articles with a distribution of 35,028 real and 37,106 fake articles. The news articles were collected from 4 popular platforms: Kaggle, McIntire, Reuters, and BuzzFeed Political. The dataset contains four columns, namely Serial Number, Title, Text, and Label, whether the article is fake, where 0 represents fake and 1 represents real.

B. Evaluation metrics

In this section, we mention and formulate the evaluation metrics used by us to measure the fake news classification prowess. To assess the performance of our proposed model, we have used benchmarked and standard evaluation metrics, namely precision, recall, F1-Score, and accuracy. Since all these metrics use information represented in the confusion matrix, we start by illustrating the confusion matrix.

1) Confusion Matrix: The information about actual and predicted classifications performed by a classifier is represented by a confusion matrix. Performance evaluation of a classifier is commonly done using the data in the confusion matrix. A confusion matrix for the two-class problem is given in tab. V.

TABLE V
REPRESENTATION OF CONFUSION MATRIX

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

2) Accuracy: Accuracy is a measure of total correctly identified samples out of all the samples. Through accuracy, the quality of the produced solution is evaluated based on the percentage of correct predictions over total instances. It helps us identify the quality of our framework to classify the fake news articles as fake and the real news articles as real. We define accuracy in Eq. 5 as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

3) Precision & Recall: The measure of the ratio between the true positives and all the positives is known as precision. Precision also gives us a measure of the relevant data points. Precision helps us understand the performance of our model to classify actual fake news articles as counterfeit amongst all the news articles classified as fake. We define precision in Eq. 6 as follows.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

Whereas the measure of the ability of the model to accurately identify the occurrence of a positive class instance is determined by recall. Recall helps us in deciding the number of actual fake news articles that were classified as fake. We define recall in Eq. 7 as follows.

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

4) F1 Score: F1 Score, defined as the harmonic mean of precision and recall value, is also used to measure the performance of our method. F1 Score helps ascertain the model's performance in striking a balance between Precision and Recall when there is an imbalanced dataset. We define F1 Score in Eq. 8 as follows.

$$F1\,Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

where *True positive (TP)* represents Correctly identified, *False positive (FP)* represents Incorrectly identified, *True negative (TN)* represents Correctly rejected and *False negative (FN)* represents the Incorrectly rejected data points.

V. THE PROPOSED *FakeScan*MODEL

This section illustrates the details of the proposed *FakeScan* model for fake news detection using a Modified Grasshopper Optimization and a convolutional neural network-based framework (MGO-CNN). The proposed model has four components:

Data Processing, Feature Generation, Feature Selection using Modified Grasshopper Optimization, and Classification using Convolutional Neural Network. In the first component, we start by cleaning and processing the input news articles to obtain uniformity across the news articles. Then in the second component, we represent each news article in a lower-dimensional vector space (d). We perform feature selection using modified grasshopper optimization (MGO) to obtain more relevant features, reducing dimensionality in component three. Finally, in component four, we process the selected features using convolutional neural networks to obtain n-gram features and classify each news article as real or fake based on its characteristics. Fig. 3 shows the overall architectural flow of *FakeScan*.

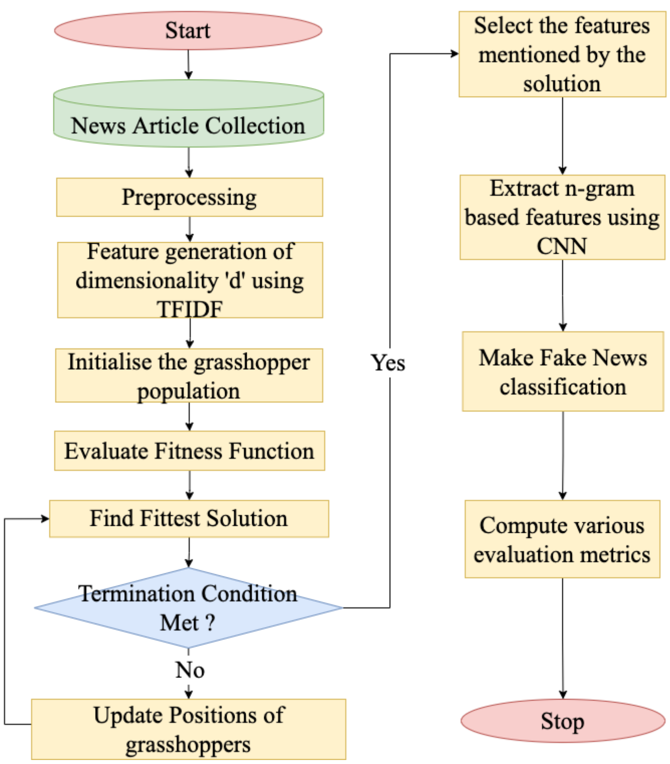

Fig. 3. Flow diagram for our proposed framework for fake-news detection using Modified Grasshopper optimization and Convolutional Neural Network.

A. Data Processing

The raw news articles present in the dataset or available on the internet have too many aberrations and are very noisy. Hence, we preprocess every news article to remove such words and obtain uniformity across the articles. First, we remove all the URLs, HTML tags, parentheses, slashes, dashes, and multiple white spaces from the news articles. Then we convert all the words of type "@Alice" and "#Bob" to "Alice" and "Bob." Then we convert all the news articles into lower case and remove all the stopwords like a, an, the, etc. http://www.ranks.nl/stopwords. If a word has a character repeated more than three times consecutively, we replace that with a single occurrence of that character. For instance, "Wowwwwww" is replaced with "Wow." We also replace acronyms with their full forms www.netlingo.com/acronyms.php. For example, "UN" changes to "United Nations." Finally, we obtain a well-processed dataset with uniformity across the news articles.

B. Feature Generation

In general, for a classification task, a well-defined and uniform feature set for the data points and classes in which they will be classified is required. For our study of fake news detection, we consider the individual news articles as data points for our classification while real and fake act as our classes. We obtain the Term Frequency Inverse Document Frequency (TF-IDF) values for every word in the article and then sort the words in descending order of their TF-IDF values. We then select the top d words based on their TF-IDF values to obtain a feature set in a pre-decided (d) dimensional vector space. The notion behind selecting the top d words and not all words in the corpora is that articles may have a lot of words, and all words are not equally relevant. Moreover, expanding the dimensionality of the vector space (d) to the number of different words in the corpora also tends to increase the computational complexity by a considerable margin. The obtained vector space acts as the feature space for news articles, and the corresponding vectors act as feature set for each news article. We vary the value of d to understand its impact on the performance of our proposed work and choose that value that yields the optimal results.

C. Feature Selection using Modified Grasshopper Optimization

In recent years, metaheuristic algorithms have gained attraction for solving various optimization problems like feature selection because of their ability to avoid local optima and search for a solution close to global optima [5]. Grasshopper Optimization Algorithm (GOA) is a recent swarm intelligence algorithm proposed by Saremi et al. [27] which mimics the grasshoppers' foraging and swarming behavior. The life cycle of grasshoppers has two phases: the nymph phase and the adult phase. The nymph phase includes small steps and slow movements, while the adult phase includes long-range and abrupt movements. We introduce a modified version of the recently proposed Grasshopper Optimization algorithm for feature selection as classical Grasshopper Optimization Algorithm being continuous in nature can't be used for a discrete problem like feature selection. Feature selection refers to selecting a subset of features from a feature space to yield the most optimal results with less computational cost. We start by generating a population of grasshoppers characterized by a d dimensional vector where d is the number of features with their values randomly initialized in the range of 0 to 1. Then based on the fitness function, we estimate the fitness value of all the grasshoppers and update the positions of all

TABLE VI
NOTATION TABLE FOR THE MODIFIED GRASSHOPPER OPTIMIZATION ALGORITHM FOR FEATURE SELECTION.

Notation	Description
N	Size of the population
d	Dimensionality of the feature space
X_i	Position of the i^{th} grasshopper
D_{ij}	Distance between the i^{th} & j^{th} grasshoppers
c	Comfort, attraction & repulsion zone decreasing coefficient
l	Current iteration
L	Maximum number of iterations
ub_k & lb_k	Upper and lower bounds of k^{th} dimension
S()	The social interaction function
T	The fittest grasshopper so far

of them based on the fittest grasshopper. This step is repeated iteratively until the termination condition is met. After the process is terminated, we choose the attributes based on the fittest grasshopper. This is done by selecting the attributes for which the value of the grasshopper's vector space is greater than 0.5. This helps us modify the classical grasshopper optimization algorithm to suit the discrete task of feature selection. To better understand the algorithm, we also present a notation Tab. VI. The various phases of the proposed Modified Grasshopper Optimization algorithm is described below.

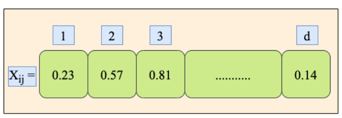

Fig. 4. Representation of the position of a grasshopper.

1) Population Initialization: For any population based algorithm, the first is to initialize a population. Let N be the size of the population and d be the dimensionality of the feature space. So for every grasshopper in the population we generate a d dimensional vector having random values in the interval [0, 1]. Hence the position of every grasshopper can be represented as X_i, (i = 1, 2, 3,..., N). This is shown in Fig. 4. Each dimension of the grasshopper i can be represented as follows:

$$X_{ij} \;=\; random(), j \in [1, d] \; and \; i \in [1, N] \tag{9}$$

Here, random() gives a random number in the range [0, 1]. Fig. 5 shows the representation of the positions of all the grasshoppers.

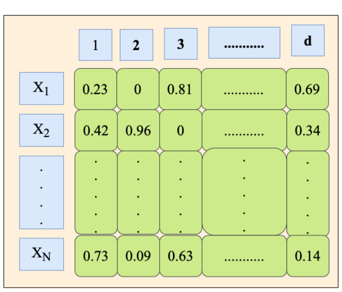

Fig. 5. Representation of the positions of the entire population of the grasshopper.

2) Fitness function calculation: After initializing the population, we need to evaluate the fitness function for every solution, i.e., we need to evaluate the fitness value of every grasshopper. For our work, we use the following fitness function for the i^{th} grasshopper.

$$Fitness(X_i) \;=\; errorRate * \frac{\sum_{j=1}^{d} round(X_{ij})}{d} \tag{10}$$

Here, round(X_{ij}) returns the rounded-off value of X_{ij}, i.e., for values greater than 0.5, it returns 1, while for values less than 0.5, it returns 0. The $errorRate$ is the classification error using the selected features made by an artificial neural network classifier. For the i^{th} possible solution, the j^{th} feature is selected if the value of $X_{ij} > 0.5$. For this study, we try to minimize the fitness function, i.e., Eq. 10. This is done based on the notion that we try to reduce the classification error for any classification task while selecting the minimum number of features.

3) Position Update: After calculating the fitness function for every grasshopper, next we go onto updating their positions by considering the social interaction operator (S_i), the gravity force operator (G_i) and the wind advection operator (A_i) as follows.

$$X_i \;=\; S_i + G_i + A_i \tag{11}$$

To fit the feature selection task in a better way, we modify the above equation and ignore the effect of gravity operator (G_i) and assume that the direction of wind is always towards the target. The target is the fittest grasshopper. Therefore, the position update equation changes as follows.

$$X_{ik}^{t+1} \;=\; c\left(\sum_{j=1 j\neq i}^{N} c\frac{ub_k - lb_k}{2} S(\mid X_{jk}^t - X_{ik}^t \mid)\frac{X_j^t - X_i^t}{D_{ij}}\right) + T_k^t \tag{12}$$

Here, X_{ik}^t represents the value of the k^{th} dimension of the position of the i^{th} grasshopper at time t, c is a decreasing coefficient to shrink the comfort zone, attraction zone and repulsion zone. The value of c is defined below.

$$c \;=\; C_{max} - l\frac{C_{max} - C_{min}}{L} \tag{13}$$

where C_{max} is the maximum value, C_{min} is the minimum value, l indicates the current iteration, and L is the maximum number of iterations. We use C_{max} = 1 and C_{min} = 0.00001. The upper and lower bound of the k^{th} dimension is denoted as ub_k and lb_k, respectively. Also, $S()$ is a function that defines the social forces and is defined as follows.

$$S(r) \;=\; fe^{\frac{-r}{l}} - e^{-r} \tag{14}$$

where f and l are constants representing the intensity of attraction and attractive length scale, respectively, while r is a real value. The distance between the i^{th} and j^{th} grasshopper is denoted as d_{ij} and it is calculated as calculated as $\mid X_j^t - X_i^t \mid$. The $\frac{X_j^t - X_i^t}{D_{ij}}$ is a unit vector from i^{th} to j^{th} grasshopper. The value of the k^{th} dimension of the fittest solution or the fittest grasshopper so far is denoted by T_k^t. Eq. 12 is used repeatedly and iteratively to update the position of the grasshoppers based on the position of other grasshoppers as well as the fittest grasshopper found so far. This process is carried out for a fixed and pre-stipulated number of maximum iterations, L.

4) Termination: Now, we mention the termination conditions for the proposed modified grasshopper optimization algorithm. Our algorithm terminates after running for a fixed number of pre-decided iterations (L). After this, we select the grasshopper with the smallest value of the fitness function, or we choose the fittest solution. Then based on this solution, we choose the features whose corresponding dimension in the fittest solution is greater than 0.5. These features form our final feature set for which we make our final classification.

D. Classification using Convolutional Neural Network

In this phase, we utilize deep convolutional neural networks (CNN) to extract the n-gram based features from the news articles and make the final fake news classification. We alter the feature vector of each news article by using the TF-IDF of the features that are selected in the previous step and zero for the features that are not selected. To comply with the input requirements of the CNN, we convert the features of every news article into a 2D matrix and then pass it to the CNN. The feature vector of each news article is converted into a 2 dimensional matrix of dimension 25×100. This dimensionality is in accordance with the size of the feature vector, which is chosen to be 2500 based on the experimental analysis discussed ahead in Section VI-A. Fig. 6 depicts the pictorial representation of generation of 2D feature matrix using the selected features from the feature vector of every news article.

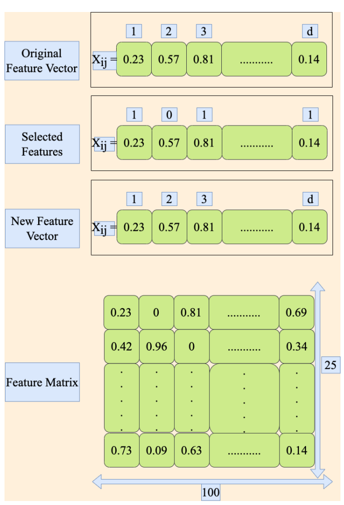

Fig. 6. Generation of 2D feature matrix using the selected features from the feature vector of every news article.

The feature matrix generated above is then passed through three different convolutional layers concurrently. The filter sizes of these convolutional layers are 2, 3, and 5, respectively. Different filter sizes are chosen to capture the details of the news articles based on n-gram models. Therefore, the three convolutional layers select the 2-gram, 3-gram, and 5-gram features of the news article. These extracted n-gram features helps our model to incorporate the impact of a combination of words in signalling whether a news article is real or fake. Then we concatenate the output of these layers using a concatenation layer to generate a combined output containing the features extracted from all the convolutional layers. This output is then passed through a fully-connected dense layer to finally classify the news articles as real or fake. Fig. 7 shows the process of extracting the n-gram features from the feature matrix of the news articles using a convolutional neural network which are then concatenated using the concatenation layer. The output of the concatenation layer is then passed through a dense layer and classified as real or fake depending on their characteristics.

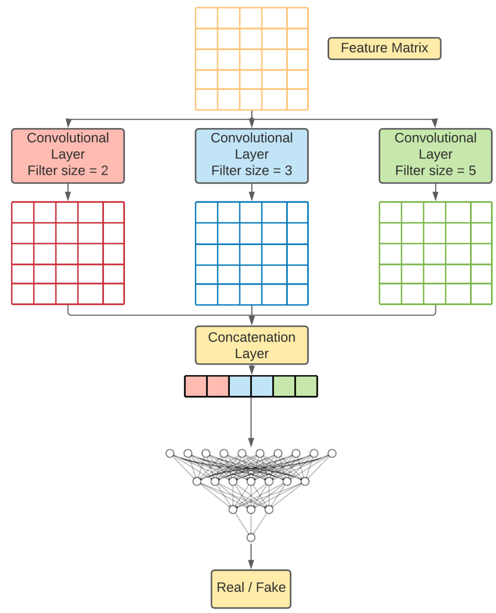

Fig. 7. Process of extraction of n-gram features from the feature matrix of the news articles using the convolutional layers and making classifying them as fake or real.

For training the classifier, we first split the entire dataset into training and testing datasets. The split is done in an 80:20 ratio, which is one of the common practice, with 80% of the dataset kept for training purposes while 20% of the dataset is reserved for testing purposes. The split has been done in such a manner so that we can prevent under-fitting and over-fitting. This also reduces the bias of the classifier towards a single output class. After training the classifier on the input news articles for the selected features, we test the efficiency of the proposed framework on test data.

VI. Experimental Results and Analysis

In this section, we present the performed experimental result and analysis to validate the efficacy of the proposed fake news detection framework. We obtained the results on all the datasets mentioned in Section IV-A based on all the evaluation metrics as presented in Section IV-B. We have split the entire dataset into two parts for our experiments, namely training and testing datasets. The split was done in an 80:20 ratio with 80% of the dataset being reserved for training while 20% of the dataset reserved for testing. The model parameters were fixed and used in similar settings across the datasets. The parameters considered for the proposed MGO-CNN framework are listed in Tab. VII. For our Modified Grasshopper Optimization (MGO) algorithm, we chose the number of grasshoppers to be 200 while the maximum number of iterations is chosen to be 300. For initial feature generation, as mentioned in Section V-B, we picked the value of d to be 2500. The value of C_{max}, C_{min}, l and f were taken to be 1, 0.00001, 1.5 and 0.5 respectively. For every dataset and every evaluation metrics, we run our experiments 100 times and the results were averaged out, to obtain more stable results that are free from statistical aberrations. The size of the filters for CNN are chosen to be 2, 3, and 5. We compare the performance of our algorithm with several contemporary fake news detection algorithms like Cross-SEAN [24], C-BiLSTM [26], BerConvoNet [11], Semantics FND [28], and DeepFakE [29] to present the utility of our approach with other fake news algorithms. We also compare the performance of our proposed framework with some of the classical, and contemporary metaheuristic algorithms like Dragonfly optimization (DGO) [30], Grey Wolf Optimization (GWO) [31], Particle Swarm Optimization (PSO) [32], Firefly optimization (FO) [33], Ant Colony Optimization (ACO) [34] and Whale optimization (WO) [35]. This provides us an understanding of the performance comparison as opposed to other metaheuristic based algorithms. The entire code is developed in the python programming language. To obtain results on existing techniques, we use several python libraries like sklearn, NumPy, pandas, etc., and some publicly available GitHub repositories. We performed simulations on a personal computer with Intel i7 11th generation processor, 16 GB RAM, and RTX 3070 graphics card. The various simulations performed for the hyperparameters and evaluation metrics are discussed below.

TABLE VII
PARAMETER SETTINGS FOR OUR PROPOSED MGO-CNN FRAMEWORK.

Parameter	Value
Dimensionality of feature space (d)	2500
Number of grasshoppers in the population (N)	200
Maximum number of iterations (L)	300
Maximum value of decreasing coefficient (C_{max})	1
Minimum value of decreasing coefficient (C_{min})	0.00001
Attractive length scale (l)	1.5
Intensity of attraction(f)	0.5
Number of simulations	100
CNN Filter Size	2, 3, & 5

A. Dimensionality of feature space (d) vs. Accuracy

Fig. 8 shows the effect of varying the dimensionality of feature space via TF-IDF as mentioned in Section V-B. Here, we vary the dimensionality in steps of 500 starting from 500 and study its impact on accuracy across all the datasets. We observe three maxima at 1500, 2500, and 4000. But there is a global maximum at 2500, and the accuracy on all the datasets is maximum at 2500 only. Also, as the dimensionality of the feature space increases, the time it takes for the algorithm to run also increases. Therefore, we choose dimensionality (d) to be 2500, as discussed in Section V-B, for the feature space of our algorithm.

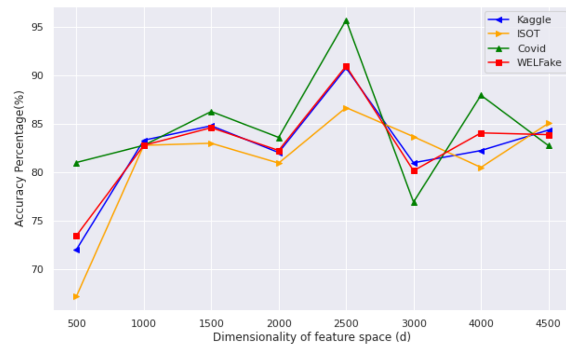

Fig. 8. Dimensionality of feature space (d) vs. Accuracy for various datasets.

B. Parameter Setting

As part of the parametric study of the Modified Grasshopper Optimization (MGO) algorithm, we studied the variations in the classification capability of the algorithm due to the variation in the maximum number of iterations and full population size. We evaluated the performance on all the datasets mentioned in Section IV-A. The results obtained are as follows.

1) Maximum number of iterations (L) vs. Accuracy: Fig. 9 shows the impact on prediction accuracy of our proposed model based on the variation in the maximum number of iterations across all the datasets. From Fig. 9 we can observe that as the maximum number of iterations (L) increases from 50 to 100, there is a steep increase in the accuracy of our proposed Modified Grasshopper Optimization (MGO) algorithm. But for the values of L between 100 to 250, there is a steady and almost horizontal growth. But when the maximum number of iterations is 300, we can see a peak in the performance of the proposed MGO algorithm, post which we see a decline in the performance. Moreover, as the number of iterations increases, the computational time also increases. This suggests we choose the maximum number of iterations L to be 300 as optimal. This gives us the justification for using the maximum number of iterations to be 300. It provides an optimal fake news detection accuracy across all the datasets while maintaining the computational feasibility of the process.

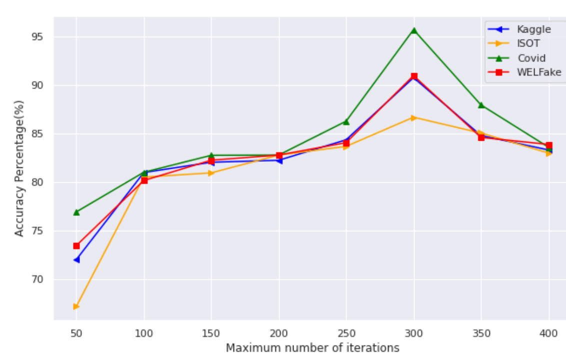

Fig. 9. Maximum number of iterations (L) vs. Accuracy for various datasets.

2) Size of the population (N) vs. Accuracy: Fig. 10 depicts the impact of increasing population size on the accuracy achieved by our proposed framework for fake news detection. It is evident that with an increase in the size of the population, the accuracy increases. But after reaching a threshold size of 200, the accuracy starts to drop for all the datasets. The computational time for the algorithm also increases. Hence, the selected choice of population size offers an optimally efficient and effective balance in the parameters.

Fig. 10. Size of the population (N) vs. Accuracy for various datasets.

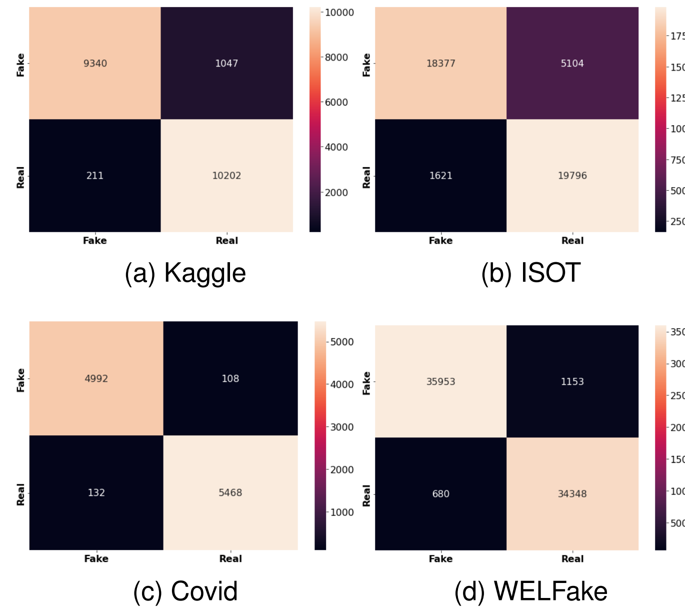

Fig. 11. Confusion matrix comparisons of various algorithms on all the datasets chosen by us.

C. Confusion Matrix

Fig. 11 shows the confusion matrix obtained by our proposed algorithm for various datasets. It clearly shows the exemplary performance of our proposed framework in classifying the fake news articles as fake and real news articles as real. We can see that the number of fake news articles classified as fake is 9340, 18377, 4992, and 35953 for Kaggle, ISOT, Covid, and WELFake datasets, respectively. Also, the number of news articles classified as real which are real by our proposed Modified Grasshopper Optimization algorithm are 10202, 19796, 5468 and 34348 for Kaggle, ISOT, Covid and WELFake datasets respectively. This shows that our model delivers good performance in detecting fake as well as real news articles. Fig. 11 also shows that our Modified Grasshopper Optimization algorithm has very few misclassifications. This is due to the proper feature generation using TF-IDF and then an appropriate feature selection using the Modified Grasshopper Optimization algorithm.

D. Comparison with contemporary fake-news methods

In this section, we compare the performance of our proposed algorithm MGO-CNN with several recent contemporary fake news detection methods like Cross-SEAN [24], C-BiLSTM [26], BerConvoNet [11], Semantics FND [28], and DeepFakE [29]. Tab. VIII shows the accuracy, precision, recall and F1 Score results obtained by various algorithms on all the different datasets used by us. Across all the datasets, we can see that proposed fake news detection framework performs the best in terms of accuracy, precision, recall, and F1 Score and gives a stable performance throughout. Semantics FND performs the worst for the Kaggle and Covid dataset. BerConvoNet and Cross-SEAN perform the worst for the ISOT and WELFake fake news dataset, respectively. It can be seen that for all the datasets, our method outperforms all the contemporary fake news detection methods by a considerable difference. This generates the utility of our proposed MGO-CNN algorithm as a benchmarked algorithm for the detection of fake news. Such excellent values of precision show the capability of our proposed approach to predict very few real news articles as fake. The high recall values also show that our proposed approach classifies a very less number of fake news articles as real. The superior performance obtained by our proposed algorithm can be understood due to the proper feature generation using TF-IDF and better exploration and exploitation capabilities of the Modified Grasshopper Optimization (MGO) algorithm compared to other algorithms. Moreover, the proper use of a deep convolutional neural network (CNN) to extract n-gram features from the text also helps our algorithm to extract the latent features from the news articles and classify the news articles optimally. The above discussion shows the utility of our proposed work in terms of fake news detection compared to other recent methods.

E. Comparison with metaheuristic optimization methods

The proposed fake news detection framework adopts Modified Grasshopper Optimization (MGO) algorithm as the feature selection method. In this section, we present the performance comparison of the adopted Modified Grasshopper Optimization (MGO) algorithm for feature selection against some popular metaheuristic optimization algorithms in our proposed framework. For this analysis, we utilized all the strategies of our proposed framework, namely, data processing, feature generation, and classification using convolutional neural network except the future selection methods. For future selection, we replaced the Modified Grasshopper Optimization (MGO) in

TABLE VIII
PERFORMANCE COMPARISON OF OUR PROPOSED MGO-CNN FRAMEWORK WITH SEVERAL CONTEMPORARY FAKE-NEWS DETECTION METHODS IN TERMS OF ACCURACY (ACC.), PRECISION (PREC.), RECALL (REC.) AND F1 SCORE (F1).

c— Methods	Kaggle				ISOT				Covid				WELFake			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
Cross-SEAN	80.16	80.47	80.47	80.47	88.92	90.15	87.72	88.92	78.91	76.37	82.3	79.22	85.89	84.66	87.5	86.06
C-BiLSTM	74.6	73.98	73.98	73.98	80.73	74.8	89.81	81.62	84.22	84.34	84.34	84.34	90.2	97.91	91.04	94.35
Semantics FND	71.43	73.98	69.47	71.65	90.59	89.41	91.11	90.25	72.84	68.36	85.5	75.97	90.94	98.34	91.47	94.78
DeepFakE	75.79	72.26	86.15	78.6	70.56	65.35	92.11	76.46	89.23	86.96	90.61	88.75	93.55	95.78	97.1	96.44
BerConvoNet	96.85	82.79	82.79	82.79	69.35	61.65	94.92	74.75	89.07	86.99	91.77	89.32	93.43	99.13	93.52	96.24
MGO-CNN	97.86	95.97	97.54	96.75	98.04	99.21	96.92	98.05	98.43	98.42	98.42	98.42	95.55	98.14	96.89	97.51

TABLE IX
PERFORMANCE COMPARISON OF OUR PROPOSED MGO-CNN FRAMEWORK WITH SEVERAL METAHEURITIC ALGORITHMS. HERE, THE MGO IS REPLACED BY OTHER METAHEURISTICS IN OUR PROPOSED FRAMEWORK IN TERMS OF ACCURACY (ACC.), PRECISION (PREC.), RECALL (REC.) AND F1 SCORE (F1).

Methods	Kaggle				ISOT				Covid				WELFake			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
DGO	75.79	72.26	86.15	78.6	70.56	65.35	92.11	76.46	62.32	62.32	82.16	70.88	90.2	97.91	91.04	94.35
GWO	81.35	78.52	85.48	81.85	69.35	61.65	94.92	74.75	72.84	68.36	85.5	75.97	85.33	98.52	84.96	91.24
PSO	80.16	80.47	80.47	80.47	80.73	74.8	89.81	81.62	78.91	76.37	82.3	79.22	93.43	99.13	93.52	96.24
FA	71.43	73.98	69.47	71.65	85.89	84.66	87.5	86.06	84.22	84.34	84.34	84.34	85.89	84.66	87.5	86.06
ACO	74.6	73.98	73.98	73.98	85.49	83.66	82.5	83.06	89.23	86.96	90.61	88.75	90.94	98.34	91.47	94.78
WOA	83.33	82.79	82.79	82.79	90.59	89.41	91.11	90.25	89.07	86.99	91.77	89.32	93.55	95.78	97.1	96.44
MGO-CNN	97.86	95.97	97.54	96.75	98.04	99.21	96.92	98.05	98.43	98.42	98.42	98.42	95.55	98.14	96.89	97.51

our proposed framework by some popular metaheuristic algorithm like Dragonfly optimization (DGO) [30], Grey Wolf Optimization (GWO) [31], Particle Swarm Optimization (PSO) [32], Firefly optimization (FO) [33], Ant Colony Optimization (ACO) [34] and Whale optimization (WO) [35]. Tab. IX presents the results obtained for the various evaluation metrics obtained on all the datasets for all the chosen metaheuristic algorithms augmented in our framework. From the obtained results, we can infer that our proposed fake news detection framework using Modified Grasshopper Optimization (MGO) algorithm performs the best across all the datasets for all the evaluation metrics except for precision in WELFake dataset. In terms of precision, our framework lags by a very slight margin from PSO, GWO, and ACO algorithms for the WELFake dataset. Overall, the performance of our framework is followed by the Whale Optimization algorithm (WOA). All the other metaheuristic algorithms utilized in the proposed framework follow thereby and perform closely to each other for fake news detection across the datasets and evaluation metrics. The high accuracy values obtained by our framework demonstrate its capability to make appropriate classification of the news articles to their corresponding categories. Moreover, the good F1 Score results exhibit the performance of our approach in attaining a proper balance between precision and recall, thus making fewer wrong classifications. The superior results of our model compared to other metaheuristics optimization algorithms justify our choice of using the Modified Grasshopper optimization algorithm as a feature selection method in our framework.

All the above experimental results discussion shows that the proposed fake news detection framework is the best performer across all the evaluation metrics mentioned in Section IV-B for all the datasets presented in Section IV-A. The generated feature vectors for the news articles using the TF-IDF approach, followed by proper hyperparameter tuning, enabled us to choose an optimal dimension (d) of feature vectors to capture all the important features of the news articles. Using the Modified Grasshopper Optimization Algorithm with proper parameter tuning helped us to select the most relevant and the least correlated features from amongst the feature vectors. This can be attributed to the strong search space exploration and exploitation capabilities and robustness of the Modified Grasshopper Optimization algorithm. Also, by using a deep convolutional neural network, we were able to extract the n-gram features of the news articles. The result obtained in Section VI-E shows that the proposed framework utilizing MGO as a feature selection method significantly beats all other metaheuristics algorithms used for feature selection. Overall, we can say that the results obtained demonstrate the utility of the proposed Modified Grasshopper Optimization and Convolutional Neural Network (MGO-CNN) framework for fake news detection.

VII. CONCLUSION

Fake news refers to false or misleading information that often leads to serious harm to individuals, organizations, and societies. Fake news detection is a popular research topic in social network analysis. In this paper, we introduced a novel fake news detection framework using Modified Grasshopper Optimization (MGO) algorithm and Convolutional Neural Network (CNN). We started by cleaning and processing each news article. Then we generated a d dimensional feature vector for every news article. The dimension of the feature vectors are chosen using proper hyperparameter tuning to capture all the essential characteristics from the news articles. The extracted features are then passed through the MGO algorithm to select the most relevant and the least correlated features. We modified the classical Grasshopper Optimization algorithm to fit the problem of feature selection required for fake news detection. The initial population size and the termination condition for the MGO were chosen using proper hyperparameter tuning to obtain the optimal results. The selected features are then passed through deep convolutional neural networks to extract the n-gram based features used to make the final fake news classification. We ran experiments on four benchmarked fake news detection datasets and evaluated several popular performance metrics for the fake news detection task. We compared the performance of our model with several contemporary and metaheuristic algorithms to obtain a sense of comparative study. The obtained experimental results reveal the exemplary performance of our proposed framework. As part of future work, we can explore multimedia datasets like audio video based fake news detection. In addition to that, we can explore our framework for fake news detection on multi-lingual datasets.

REFERENCES

[1] D. Chaffey, "Global social media statistics research summary 2022."

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

[3] P. Meel and D. K. Vishwakarma, "Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities," *Expert Systems with Applications*, vol. 153, p. 112986, 2020.

[4] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu, "Combating fake news: A survey on identification and mitigation techniques," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 3, pp. 1–42, 2019.

[5] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm and Evolutionary Computation*, vol. 54, p. 100663, 2020.

[6] 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.

[7] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Security and Privacy*, vol. 1, no. 1, p. e9, 2018.

[8] ——, "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.

[9] P. Patwa, S. Sharma, S. Pykl, V. Guptha, G. Kumari, M. S. Akhtar, A. Ekbal, A. Das, and T. Chakraborty, "Fighting an infodemic: Covid-19 fake news dataset," *Communications in Computer and Information Science*, p. 21–29, 2021.

[10] P. K. Verma, P. Agrawal, I. Amorim, and R. Prodan, "Welfake: Word embedding over linguistic features for fake news detection," *IEEE Transactions on Computational Social Systems*, 2021.

[11] M. Choudhary, S. S. Chouhan, E. S. Pilli, and S. K. Vipparthi, "Berconvonet: A deep learning framework for fake news classification," *Applied Soft Computing*, vol. 110, p. 107614, 2021.

[12] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational fact checking from knowledge networks," *PloS one*, vol. 10, no. 6, p. e0128193, 2015.

[13] S. C. R. Gangireddy, C. Long, and T. Chakraborty, "Unsupervised fake news detection: A graph-based approach," in *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, 2020, pp. 75–83.

[14] K. Shu, H. R. Bernard, and H. Liu, "Studying fake news via network analysis: detection and mitigation," in *Emerging research challenges and opportunities in computational social network analysis and mining*. Springer, 2019, pp. 43–65.

[15] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *Proceedings of the ACM SIGKDD workshop on mining data semantics*, 2012, pp. 1–7.

[16] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2012, pp. 171–175.

[17] H. Karimi and J. Tang, "Learning hierarchical discourse-level structure for fake news detection," *arXiv preprint arXiv:1903.07389*, 2019.

[18] 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.

[19] J. C. Reis, A. Correia, F. Murai, A. Veloso, and F. Benevenuto, "Supervised learning for fake news detection," *IEEE Intelligent Systems*, vol. 34, no. 2, pp. 76–81, 2019.

[20] X. Zheng, Z. Zeng, Z. Chen, Y. Yu, and C. Rong, "Detecting spammers on social networks," *Neurocomputing*, vol. 159, pp. 27–34, 2015.

[21] H. Karimi, P. Roy, S. Saba-Sadiya, and J. Tang, "Multi-source multi-class fake news detection," in *Proceedings of the 27th international conference on computational linguistics*, 2018, pp. 1546–1557.

[22] L. Wu and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate," in *Proceedings of the eleventh ACM international conference on Web Search and Data Mining*, 2018, pp. 637–645.

[23] 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.

[24] W. S. Paka, R. Bansal, A. Kaushik, S. Sengupta, and T. Chakraborty, "Cross-sean: A cross-stitch semi-supervised neural attention model for covid-19 fake news detection," *Applied Soft Computing*, vol. 107, p. 107393, 2021.

[25] S. R. Sahoo and B. B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Applied Soft Computing*, vol. 100, p. 106983, 2021.

[26] T. E. Trueman, A. Kumar, P. Narayanasamy, and J. Vidya, "Attention-based c-bilstm for fake news detection," *Applied Soft Computing*, p. 107600, 2021.

[27] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.

[28] A. M. Braşoveanu and R. Andonie, "Integrating machine learning techniques in semantic fake news detection," *Neural Processing Letters*, pp. 1–18, 2020.

[29] 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.

[30] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

[31] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[32] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[33] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *Computers & Security*, vol. 81, pp. 148–155, 2019.

[34] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[35] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.