International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Detection of Fake News Stance Employing Swarm Intelligence Based Feature Extraction and Ensemble Deep Learning

Vaideghy A.[1], Thiyagarajan C.[2]

Submitted:23/04/2023 **Revised**:27/06/2023 **Accepted**:06/07/2023

Abstract*:* Recently, the internet being easily accessible has helped the people in finding and consuming news through social platforms owing to its reduced expense, simplicity of access, and rapid transfer of information. It is possible for the users to publish and share various kinds of information in every form just with one click of a button. Due to their detrimental effects on society and the nation, the dissemination of false information through social media and other platforms has given birth to frightening circumstances. Although MLTs (machine learning techniques) detect fake news contents in social platforms, they are complex issues that are challenging due to changing fake news contents that are presented on the internet. In this technical study, a methodology for identifying fake news using intelligent feature extraction and ensemble-based classifiers is suggested to address the aforementioned issue. This recommended approach uses a four-step process to spot fake news on social media. The dataset is initially pre-processed in the approach to turn unorganized data sets into sorted data sets. The second stage, which employs the MBDFO (Modified Binary Dragonfly Optimization) algorithm, is brought on by the varying linkages between news pieces and the unknown features of false news. d on FPSO (fuzzy particle swarm optimization) is presented in the third phase to carry out the feature reduction operation. At last, in this research work, an ELM (Ensemble Learning Model) is built for learning how the news articles are represented and the fake news detection is carried out effectively. The reasoning is developed in this research by getting a dataset from kaggle. The results achieved prove that the proposed system is effective.

Keywords: *Fake news, preprocessing, Modified Binary Dragonfly Optimization algorithm (MBDFO), Fuzzy Particle Swarm Optimization (FPSO), Ensemble Learning Model (ELM), fake news detection*.

1. Introduction

In the recent times, social media has gone much above its primary functionality in the form of a communication tool between humans. Presently, users instantly depend on social networking platforms to know about different subjects, which involves novel social concerns that are evolving everyday [1]. Also, to have information on breaking news articles. With these platforms, anybody who has an internet enabled tool can put forward their views or post a news on a budding incident, which can be seen in real-time. Consequently, social platforms has evolved to be a highly efficient tool for newsmakers. Even though social platforms facilitates remarkable access to information, the deficit of elaborate endeavours on platforms for monitoring the posts leads to the lies and misinformation being spread easily. Hence, few more research is necessary to ensure its root source and accuracy [2]. Breaking news reports are typically updated frequently in chunks, which can cause a significant portion of those changes to remain uncertain at the time of publication and later few may reveal it to be wrong. Since there is no official statement that supports or refutes a spreading rumour, social media users regularly share their own opinions on its accuracy in terms of mutual, inter-subjective insights, which may indicate the rumor's reality.

Additionally, it is extremely difficult to evaluate truth in social media information [3]. As a result, robotic technology is now essential in the struggles against spread of ambiguous social media information. Exponential increases of fake news are not only deceptive but also a serious challenge to democracy [4]. Current advancements in verifications of news can address growing demands for automations of differentiating real and false news in massive amounts of data. In fact, depending on how they were created, the most current methods for identifying fake news may be divided into two groups: linguistic or network methods. NLP (Natural language processing), for instance, focuses on news contents and aim to identify false news patterns by investigating the semantics [5]. Network approaches, on the other hand, rely on fact-checking based on well-known knowledge networks. Many other types of operations research projects are currently utilising the possibilities of these text analytics and modelling technologies. As an example, text analytics methods have been utilised to suggest changes for developing important reality health apps.

[1]*Assistant Professor, Department of Computer Science, PSG College of Arts & Science.* [1]*vaideghy@gmail.com,*
[2]*Associate Professor, Department of Computer Science, PSG College of Arts & Science. pcthiyagu@gmail.com*
** Corresponding Author Email:* [1]*vaideghy@gmail.com,*

The increase in bogus news on social platforms was instrumental in reduced QoS (Quality of Services) that the news platforms must adhere to [6]. In order to weed out the false news from the news material, it is urgently necessary to confirm the veracity of news contents. Techniques based on data mining which comprises of feature extractions and model development are frequently used for classifications of wrong news items i.e. detections of erroneous news [7]. Feature extraction stages aim to formalize news contents based on mathematical evaluations and subsequently based on feature definitions, models of MLTs are constructed to logically differentiate between ambiguous and real news. In spite of all these techniques, there is not fully stopping of fake news. The available solutions are deficit of the useful potential for the accurate classification of news [8]. This degradation has led to the research carried out on detecting fake news with efficiency and accuracy. It is a huge challenge to predict the possibilities of a certain news article to be deliberately fake.

Due to the rate of fake news production and the speed at which it spreads virally in just minutes, conventional techniques that typically rely on the use of lexical features or manual annotations by third parties resulting in limited effectiveness [9]. Hence, automated tools for detecting bogus news are imperative. A range of automated techniques that either highlight News or Social context models have been proposed for recognising false news on social media. Unlike social context models, which emphasize on social behaviours and signals that typically indicate reactions or responses that news readers exhibit on news contents, news content models focus on contents in false news detections [10]. This type of automated model is made up of data mining algorithms and false representations of social and psychological ideas. Even though earlier studies introduced MLTs for identifying social platform's ambiguous news strips, developments of deceptive news detections are incredibly challenging because of variety of news subjects on social platforms in addition to rise in fake news contents on the internet [11]. The aforementioned problems are resolved in this technical endeavour, which provides an intelligent feature extraction and ensemble-based classifier model for spotting bogus news.

The other sections of the research work are organized as given, In section 2, few latest fake news detection approaches are overviewed. In section 3, the process involved in the proposed study is discussed. In section 4, the results and the corresponding discussion are analysed. section 5 provides the conclusion and work intended for the future.

2. Literature Review

In this section, few current techniques for identifying fake news employing MLTs and deep learning approaches are reviewed.

Amer et al [12] used transformers, DLTs (deep learning techniques), and MLTs in three different studies. Word embedding was employed in each experiment to extract contextual characteristics from the articles. The studies' experimental results demonstrated better values for DLTs in terms of accuracies when compared to MLT classifiers and BERT transformers. The findings show that the accuracy levels of the LSTM and GRU models were almost identical. It was discovered that using a machine learning algorithm or DLT in conjunction with an enhanced linguistic feature set can help detect false news more accurately. A novel scheme proposed by Seddari et al. [13] combined linguistics and knowledge for detecting fake news. The scheme used feature sets which included (1) linguistic information including headlines, word counts, ease of reading, lexical diversities, and sentiments (2) new classes of knowledge based features (fact verifications including multiple information types like (i) reputation of news featuring websites (ii) coverage or counts of sources that feature news and (iii) fact checking. The suggested system uses fewer features—just eight—than the benchmark techniques do. Akinyemi et al. [14] established a methodology that accurately classifies and identifies false news pieces that dishonest individuals post on social media. Entropy-based feature selection was used to extract news contents, social context information, and the appropriate categorization of the published news from the PHEME dataset. Approaches to Min-Max Normalisation normalise the selected characteristics. A stacking fusion of three algorithms was used to create a false news prediction model. A comparison with a well-known model was made in order to simulate the model and assess its performance in terms of metrics like detection accuracy, sensitivity, and precision. In comparison to the benchmark systems, the testing results showed enhanced 17.25% detection accuracy, 15.78% sensitivity, and decreased 0.2% precision. Hence, the proposed system is successful in detecting more counts of fake news instances with improved accuracy considering news contents and social content perceptions.

Kaliyar et al [15] created effective DLTs with tensor factorization in the forefront. Using linked matrix-tensor factorizations, tensors and news contents were combined in this approach to produce latent descriptions of social environments, news contents. Social context based information were processed with deep neural networks where for independent and ensemble classifications of news contents hyper-parameters were tuned. Using actual fake news datasets that included BuzzFeed and PolitiFact, efficacy of their proposed strategy was demonstrated and in comparative benchmarks for fake news detections, their classification results for their proposed Echo FakeD was good and validation accuracies of 92.30% were reached. Their findings supported that their suggested model significantly outperformed benchmarked models in fake news detections, and they demonstrated the usefulness of applying the r false news categorization technique. Albahar et al. [16] developed a hybrid model that combined RNN (recurrent neural network) and SVM (support vector machine) for recognizing ambiguous news items. The items in the form of texts were transformed into numerical values by RNN using encoding of feature vectors and subsequently categorized as true and false using SVM with radial basis function kernels. Their experimental results on real datasets with other benchmark models showed wide margins. Bauskar et al. [17] suggested innovative MLTs based on NLP (Natural Language Processing) to handle the issue of ambiguous news of social characteristics of news contents were processed. Their experimental outcomes demonstrated accuracy values of 90.62% with F1 Scores of 90.33% on benchmarked datasets for classifications of bogus news items.

Prediction performances of most current methodologies and characteristics of automated identifications of false news was measured by Reiset al. [18], who proposed a novel bunch of features that were useful in fake news detections. The use of false news detection systems in real life is finally possible, with an emphasis on the advantages and disadvantages. A theory-oriented methodology for identifying bogus news was published by Zhou et al. [19]. The method analyses news content on a variety of levels, including the lexical, syntactic, semantic, and discourse levels. False news trends were investigated for improved comprehensions utilising feature engineering, and linkages between false information, deceit/misinformation, and click baits were researched. This was done using supervised MLTs which identified ambiguous news from two benchmark datasets early, in spite of scarce availability of information. Utilising user-related and content-related variables, Jarrahi et al. [20] suggested an exceptionally accurate multi-modal model named FR-Detect to identify false news. Additionally, to discover the best fusion of publishers' traits and latent textual content features, a convolutional neural network working at the sentence level is used. Their findings demonstrated the value of publisher attributes in improving the accuracy and performance of content-based models by over 16% and 31%, respectively. and with corresponding F1values. Moreover, the publishers' attitude in various news domains has gone through statistical study and analysis.

Sitaula et al.'s [21] presented a detection model for identification of false news through assessments of truthfulness. The examination of publicly available fake news data shows that knowledge of news sources (and publishers) might potentially serve as a standard for validity. From the data, it can be concluded that counts of writers who publish a news piece and the author's history with false news can both be significant factors in the fake news detection process. This approach improved content characteristics for spotting phoney news in conventional fake news detection algorithms. Mobile nodes and static network architecture lack access to a centralised host to assume IDs or IP addresses [33]. The use of DAGA-NN (domain-adversarial and graph-attention neural networks) in text environments with numerous events or domains was investigated by Yuan et al. [22] where it was found that the scheme allowed for accurate detections of ambiguous news when compared to conventional MLTs, even if the samples were very few and limited. After thorough testing on two multimedia datasets from Twitter and Weibo, their proposed technique demonstrated its successful identifications of false news in several areas.

3. Proposed Methodology

The four-step approach used in this suggested model is used to detect bogus news on social media. This study's inspiration came from assessing the headline-focused relative position of a news piece.

- Pre-processing is used on the data set in the first stage of the technique to turn unorganised data sets into structured data sets.
- Feature Extraction using MBDFO is used in the second stage to find the unknown features of fake news and various associations among news articles.
- Feature selections based on FPSO as third stage for reducing counts of features.
- In the end, our research successfully detects bogus news and develops an ELM for learning how to characterise news pieces. Datasets from Fake News Challenges (FNC) website was used in this work and it contained four kinds of markers namely agree, disagree, discuss, and irrelevant.

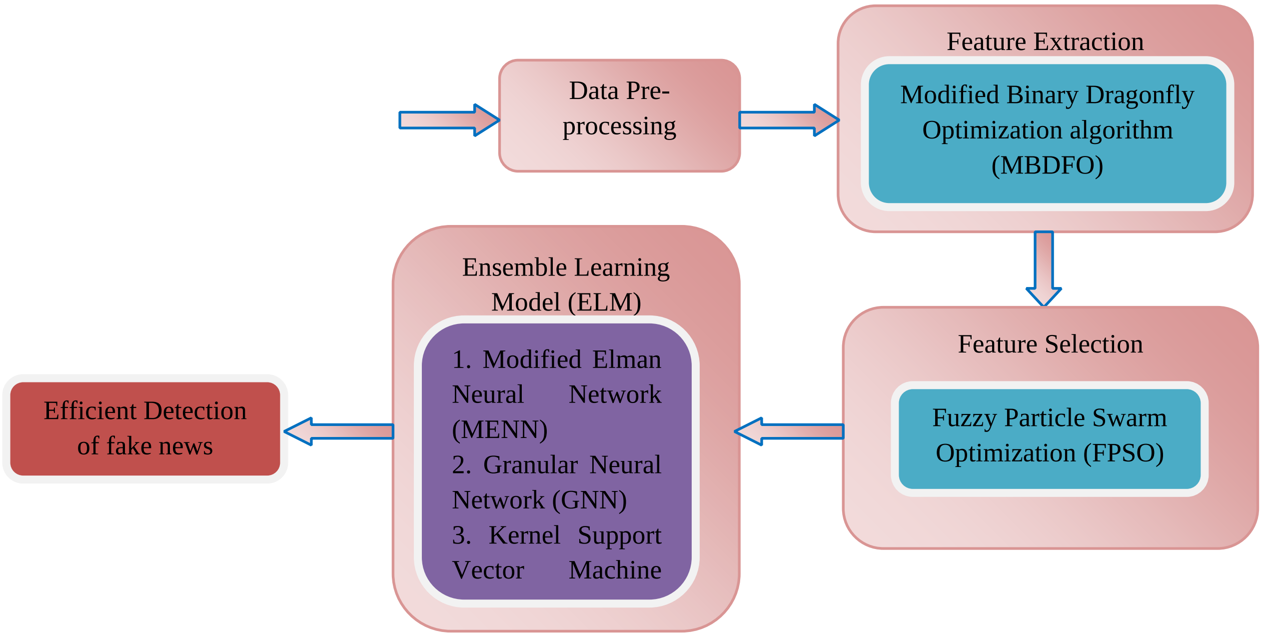

Fig 1. Fake news detection model

3.1. DATASET

The official website's standard dataset for Fake News Challenges was collected [23]. The FNC dataset consists of 75,385 labelled occurrences, 2,587 article bodies, and approximately 300 headlines. There are between 5 and 20 news items for each allegation. According to Table1, of these headlines, 7.4% are accepted, 2.0% are rejected, 17.7% are discussed, and 72.8% are unimportant. Human participation marks the assertions related to the body of the articles. The labels are explained as follows:

Agree: I concur that there is a connection between an article's headline and body.

Disagree: There is no connection between the article's substance and headline.

Discuss: how little similarity, or neutrality, exists between the title and the body.

Unrelated: The headline's subject was completely different from the content.

The dataset was divided based on FNC-1 challenge directions into testing (413 instances) and training (972 instances). The ratio of headlines to article bodies in training data is 1: 648 to 683. Nearly 880 headlines and 904 article bodies are included in the test data.

Table 1. Dataset statistics.

Dataset	Headlines	Tokens	Instances	Agree	Disagree	Discuss	Unrelated
FNC-1	**2,587**	**372**	**75,385**	**7.4%**	**2.0%**	**17.7%**	**72.8%**

3.2. PRE-PROCESSING

Pre-processes are common approaches in data mining that transform irregular and incomplete raw data into comprehensible computer forms. The processes used on the FNC-1 dataset included conversions into lowercases, eliminations of stop words, tokenization and stemming using algorithms from the Keras toolkit. Stopwords are generic words that are included in the text, are only marginally significant in terms of characteristics, and demonstrate irrelevance for this study, such as "of," "the," "and," "an," etc. By removing stopwords, processing time was reduced and space was saved by omitting the aforementioned unneeded words. Games and games are two examples of words with the same meaning that repeatedly appear in the text. In such case, rendering the text in a simple, universal format is quite beneficial. The Porter stemmer method from the NLTK's open-source implementation is used to carry out this stemming operation.

Counts of words in the title is reduced to 372 once the pre-processing processes described above have been completed. The Keras library's tokenizer function came in handy for turning the headline into a words vector. Following pre-processing, words/texts were mapped to collections of vectors using word embedding (word2vec). Finally, a dictionary of 5,000 unigram terms from article headlines and body text is created. All headlines have a set length that is equal to their longest length. The headlines that are less than the longest length are padded with zero. The functionalities are then provided to ELM.

3.3. DIMENSIONALITY REDUCTION METHODS

Counts of dimensions in text categorizations can be reduced using feature extractions or selections. Only the most important and relevant attributes are maintained in feature selection procedures, leaving out the other characteristics. However, feature extraction methods change the actual vector space in order to produce a new vector space with certain characteristics. A new vector space is used for the feature reduction. The benefit with feature reduction is that there is a consequent reduction in the processing speed which can lead to performance improvement. Feature reduction influences the results of text classification immensely. Hence, it is highly critical to select the apt selection algorithm for dimensionality reduction. The scalability of the text classifier can be improved using MBDFO and FPSO, which are two-dimensionality reduction techniques.

3.3.1. DFO (Dragonfly Optimization algorithm)

DFO is a contemporary algorithm that relies on the environment's static and moving swarms of dragonflies. The two crucial steps in the meta-heuristics optimisation area are exploration and exploitation, which are produced in DFO by the social interactions of dragonflies, which are represented for the hunt for food and escape from enemies whether their swarming is either dynamic or static [24]. Nymph and adult are the two key phases in the life cycle of dragonflies, which are fascinating insects. The most crucial portion of their lifetime was staying in the nymph stage before they underwent metamorphosis and became adults. The flight behaviour of sub-swarms across diverse areas in a static swarm is used to mimic the exploration stage. When dragonflies fly in greater swarms, the exploitation stage is successful in same directions. Three important behaviours of swarms are represented mathematically below:

To avoid static collusion, people are separated from one another in the neighbourhood. This separation is calculated as follows:

$$S_i = -\sum_{j=1}^{N} X - X_j \qquad (1)$$

where X and X_j denote, respectively, the current person's position and the position of the jth surrounding individual. N is the total counts of nearby persons. As alignments are processes of matching personal velocities with other nearby residents, and defined as follows.,

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \qquad (2)$$

where V_j is the speed of the j^{th} nearby person. Cohesions imply tendencies of groups to gravitate towards neighbourhood cores.

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \qquad (3)$$

where X and X_j stand for the current person's position and the position of the j^{th} surrounding individual, respectively. Counts of neighbours is denoted by the letter N. Any dragonfly's main goal is to survive; to do this, each individual must be drawn to food sources while avoiding predators. Attractions towards food sources are computed as:

$$F_i = X^{+} - X \qquad (4)$$

Distractions from adversaries are calculated using:

$$E_i = X^{-} + X \qquad (5)$$

where X represents current individual locations, locations of food sources and adversaries. The techniques discussed above can be utilised to imitate dragonfly behaviours in both dynamic and static swarms. In this case, step (X) and position (X) update search space's dragonflies' positions, simulating their migrations. The formula below was used to change step vectors.

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \qquad (6)$$

Where, the letters s, a, c, f and e stand for weight vectors for separations, alignments, cohesions, food, and enemies, respectively. Inertial weights are denoted by w, while iteration counter is denoted by t. Position vector computations are computed using Equation (7):

$$X_{t+1} = X_t + \Delta X_{t+1} \qquad (7)$$

During the optimisation phase, simulations of various types of exploration and exploitation are conducted using the five varied variables. Dynamic flights in search spaces that use random walks may be introduced to increase the randomness and enhance the stochastic and explorative behaviour of the dragonflies when nearby solutions are not discovered.

3.3.2. BDFO (Binary Dragonfly Optimization Algorithm)

Continuous DFOs can be transformed into binary DFOs using transfer functions without changing architectures where transfer functions with S and V shapes are commonly used [25]. V-shaped transfers outperformed s-shaped transfers as they do not need particles in the range (0, 1). On receiving velocities (step) as inputs, transfer functions return values between 0 and 1. The transfer function shown below assesses possibilities of artificial dragonflies movements towards locations.

$$T(\Delta X) = \left|\frac{\Delta X}{\sqrt{1+\Delta X^2}}\right| \qquad (8)$$

Equation (9) is used to update positions of search agents in binary search spaces..

$$X_{t+1} = \begin{cases} \sim X_t rand < T(\Delta X_{t+1}) \\ X_t rand \geq T(\Delta X_{t+1}) \end{cases} \quad (9)$$

BDFO can resolve binary issues by using transfer functions and new positional updates. The goodness/fitness evaluation procedure is the first stage in the BDFO-based feature extraction process. Usually, while designing a fitness function, the two typical measures of classification accuracy and error rate will be useful. The following phrase was used in this investigation to evaluate each dragon fly's fitness.

$$Fitness = \alpha * (Error\ Rate) + (1 - \alpha) * \frac{number\ of\ selected\ features}{Total\ number\ of\ features} \quad (10)$$

where α represents a constant that ranges from 0 to 1, giving the amount of features and classification performance (error) relative relevance. Because classification performances have far more relevance than feature counts, the value of α was fixed at 0.9. The fitness functions (Equation 10) was used to compute qualitative search agents. The two most fit dragonflies and their fitness were stored after all dragonflies had their fitness assessed. Following the completion of the update for each dragon fly's position and velocity (step), the Food supply and enemy are both updated.

3.3.3. MBDFO

MBDFO uses a permutation-based learning approach (PLS), which applies the idea of personal best and personal worst solutions, to update locations (see Fig. 2). The dragonflies employ the world's greatest solution—a plentiful food supply—and the world's worst solution—an adversary—to complete the processes of attraction and distraction in the traditional DFO. However, it has been demonstrated that including the best and worst dragonflies into these tasks enhances the ability to obtain food and hinders enemy escape tactics.

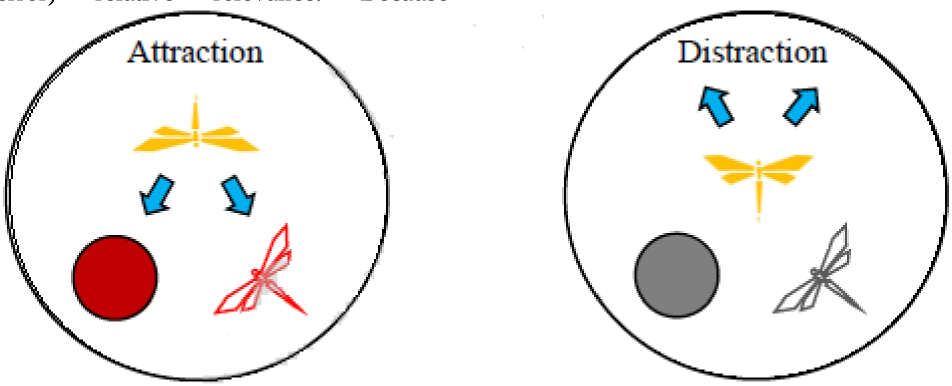

Fig. 2. Attractive and distractive natures of the proposed PLS-DFO

In contrary to DFO, the attraction and distraction of MLS are computed applying the expressions below.

$$F_i = \frac{(Xpb_i - X_i) + (Xf - X_i)}{2} \quad (11)$$

$$E_i = \frac{(Xpw_i + X_i) + (Xe - X_i)}{2} \quad (12)$$

where Xpb_i designates best dragonfly positions, Xpw_i represents worst dragonfly positions, X denotes dragonfly locations, Xf_i show food sources, and Xe_i represent adversaries. Additionally, due to mutations in learning processes, dragonflies learn about their own and global best solutions in searches. Figure 3 provides an illustration of the permutation learning method's fundamental idea. The dragonfly tries to imitate its best prior experiences as well as those from around the world, rather than altering the location in accordance with swarming habits.

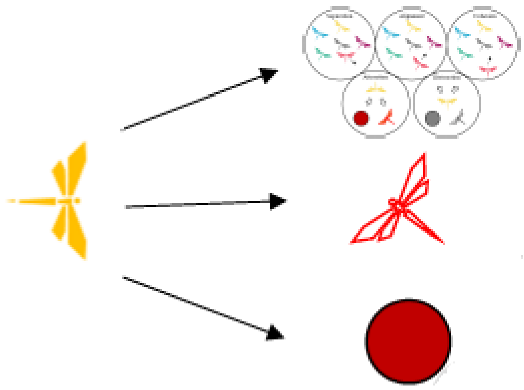

Fig. 3. General principle of the permutation based learning mechanism.

In the proposed PLS-DFO, the position of a dragonfly is revised as below:

$$X_i^d(t+1) = \begin{cases} \bar{X}_i^d & 0 \leq r_1 < pl \\ Xpb_i^d(t) & pl \leq r_1 < gl \\ Xf^d(t) & gl \leq r_1 < 1 \end{cases} \quad (13)$$

$$\bar{X}_i^d = \begin{cases} 1 - X_i^d(t) r_2 < TF(\Delta X_i^d(t+1) \\ X_i^d(t) r_2 \geq TF(\Delta X_i^d(t+1) \end{cases} \quad (14)$$

where X stands for dragonfly positions, Xpb represent best dragonfly positions, Xf implies food sources, i represents dragonfly's order, d stands for dimensions (decision variable counts), t represents current iterations, and r1 and r2 represent independent random values in the interval [0,1]. The variables pl and gl signify personal and global learning rates and their values range between 0 and 1. Equations (13 and 14) show the importance of the pl and gl in the learning process. When pl and glare levels are very low, it is easy to slip into the local optima since the algorithm often looks for the individual and collective optimal solutions. In contrast, the position update method will develop to be equivalent to DFO when the values of pl and glare are extremely high. The choice made by the pl and gl is consequently critical.

3.4. FEATURE SELECTION USING FPSO

Fuzzy matrices with n rows and c columns where n stands for data piece counts and c represents counts of clusters, used to describe fuzzy characteristics in behavioural patterns. The element in the ith row and jth column, designated by the symbol ij, μ denotes the level of membership function or correlation between the ith item and the jth cluster [26]. The suggested technique eliminates the shortcomings of fuzzy based features by using the power of global search in PSO algorithm. PSO (Particle Swarm Optimisation) is a population-based optimisation method that is normally relatively simple to use and apply in order to solve various functional optimisation issues or problems that may be converted into functional optimisation problems.

PSO is a population-based stochastic optimisation approach that depends on iterations and generations and is inspired by the characteristics of fish schools and bird flocks. In search spaces, velocities are randomly initialised, and in PSO, algorithmic executions begin with particle populations whose placements hint to potential difficulties. Iterations are used to update particle velocities and positions to find the best sites. Additionally, fitness values corresponding to particle s positions are determined by fitness functions utilised during iterations [27]. Each particle's velocity is updated using its own best position and the world's best position. The personal best position (pbest) of a particle indicates its best location, whereas the gbest of a swarm indicates its best location as of the first time step. A particle's location and velocity are updated as necessary.

$$V(t+1) = w.V(t) + c_1 r_1 \big(pbest(t) - X(t)\big) + c_2 r_2 \big(gbest(t) - X(t)\big); \quad k = 1,2,\ldots..P \quad (15)$$

$$X(t+1) = X(t) + V(t+1) \quad (16)$$

Where, X stands for particle's location and V denote represents particle's speed, while P implies swarm's particles counts, w denotes inertia weight, c_1 and c_2 are positive constant values for acceleration coefficients governing imp[acts of pbest and gbest on searches, and r_1, r_2 depict randomized values in the interval [0,1[. Positions and velocities of the particles are recast in the proposed FPSO technique to describe the fuzzy relationship between variables.

In FPSO algorithm X, the position of particle, defines the fuzzy association between a group of data objects, $o = \{o_1, o_2, \ldots \ldots, o_n\}$, and set of cluster centers, $Z = \{z_1, z_2, \ldots \ldots, z_c\}$. X. the expression is as below:

$$X = \begin{bmatrix} \mu_{1l} & \cdots & \mu_{1c} \\ \vdots & \ddots & \vdots \\ \mu_{nl} & \cdots & \mu_{nc} \end{bmatrix} \quad (17)$$

In which μ_{ij} stands for membership functions of ith object with jth clusters with conditions and hence, position matrices of particles and fuzzy matrices are similar. Moreover, matrices of [-1, 1]-ranged entries and n rows and c columns in sizes determine particle velocities. The following definitions specify the phrase to update the particle's locations and speeds in accordance with matrix operations..

$$V(t+1) = w \otimes V(t) \oplus (c_1 r_1) \otimes \big(pbest(t) \ominus X(t)\big) \oplus (c_2 r_2) \otimes \big(gbest(t) \ominus X(t)\big) \quad (18)$$

$$X(t+1) = X(t) \oplus V(t+1) \quad (19)$$

The requirements outlined in (18) and (19) may be violated once the position matrix was modified. The position matrix must be normalised as a result, which is crucial. First, the matrix's negative members are all made to equal zero. If all elements of rows in matrices become zero, they need to be evaluated again using randomized values in the range (0, 1), and then the matrix must go through the following alterations without breaking the rules:

$$Xnormal = \begin{bmatrix} \mu_{11}/\sum_{j=1}^{c} \mu_{1j} & \cdots & \mu_{1c}/\sum_{j=1}^{c} \mu_{1j} \\ \vdots & \ddots & \vdots \\ \mu_{n1}/\sum_{j=1}^{c} \mu_{nj} & \cdots & \mu_{nc}/\sum_{j=1}^{c} \mu_{nj} \end{bmatrix} \quad (20)$$

FPSO method also uses fitness functions like other parallel algorithms to assess generic solutions. Equation (21) can be used to examine solutions.

$$f(X) = \frac{K}{J_m} \quad (21)$$

Here, J_m stands for the objective function, while K denotes a constant. Clustering effects are higher individual fitness f(X) get maximised when J_m is lower.

Algorithm 1. Fuzzy PSO for feature selection

Input: Extracted features

Output: optimized features

1. Initialization of the parameters, which includes the largest iteration count, population size P, 1c, 2c, and w.
2. Create a swarm of P particles using the matrices X, pbest, gbest, and V.
3. Specify swarm's X, V, pbest, and gbest for particles.
4. Obtain cluster centres of particles.
5. Use Eq. (21) to obtain particle fitness values.
6. Determine the best for every particle.
7. Use the swarm's finest computing.
8. Apply Eq. (18) for updating particles' velocity matrices.
9. Using Eq. (19), for updating particles' location matrices.
10. If the stop criteria are not met, move on to step 4.

The suggested technique's stop criterion is determined by the maximum counts of iterations or by a decrease in counts of gbest in iterations.

The collection of all the typical features for each group is obtained according to the customary feature set for each cardholder:

$$G_j = \cup_{id \in j} G^{id}, \forall j \in V \qquad (22)$$

Classifying these common feature sets to specific patterns (words) is difficult in practise. This is because, even if patterns (or articles) are supported by data from the human domain, their specification may not be apparent. However, it's really simple to use a classification strategy to address this supervised learning issue, which can aid in the automated structuring of high-level abstract knowledge.

3.5. CLASSIFICATION USING ELM

The main idea behind the ensemble technique is to combine a counts of models that each find a solution to the same real-world problem in order to create a more accurate and reliable set of predictions or decisions than would be possible with just one model. The ensemble model in this part includes the following neural networks: MENN (Modified Elman Neural Network), GNN (Granular Neural Network), and KSVM (Kernel SVM).

3.5.1. ENN (Elman Neural Network)

Figure 4 displays the basic configuration of ENN which have input, hidden, context and output layers. Flexible weights link adjoining layer pairs [28]. The self-connections of context nodes are responsible for ENN's sensitivity to historical inputs which make it useful for dynamic system models and discussed in full below:

$w1_{ij}$: Weights connecting nodes i in input layers and nodes j of hidden layers.

$w2_{ij}$: Weights connecting nodes i in hidden layers and nodes j of output layers.

$w3_{ij}$: Weights connecting context nodes i and nodes j of hidden layers.

m, n, r : Counts of nodes in input, output, and hidden layers respectively.

$u_i(k), y_i(k)$: ENN's Inputs and outputs, where i=1,2,.....m, and j=1,2,.......n.

$x_i(k)$: Outputs of hidden nodes i, where i=l, 2,......r.

$c_i(k)$: Output of context nodes i or output of hidden nodes i of last time. z^{-1}**:** A unit delay.

For modules in hidden layers, second modules known as context modules are provided. Context modules are directly related in forward directions to hidden modules, suggesting context modules have weights that are transferred to hidden modules. Recurring linkages from context modules to hidden modules are also present. Figure 2 depicts the single link between each hidden module and the relevant context module.

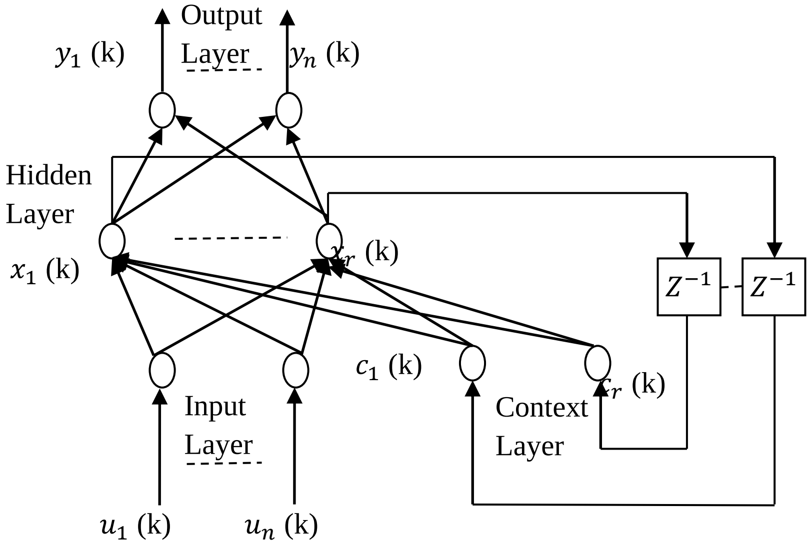

Fig. 4. Structure of the ENN model.

The forward weights of recurrent connections may be trained via back-propagation, and their weights are constant. The context modules behave similarly to input modules during the forward phase. Similar to feedforward networks, the values for the hidden modules and the output modules are calculated. After the outputs of the hidden modules have been computed, the current values are replicated (with a unit delay) into the relevant context modules through the recurrent connections. These values must be initialised with a few values at the first time step before they may be used in the following time step. Target values for the outputs are used during the backward phase of the training, and the back-propagation aids in modifying the forward weights. The network inputs are given by$(k) \in R^m y(k) \in R^n$, $x(k) \in R^r$, so the outputs in every layer can be formulated as

$$x_j(k) = f(\sum_{i=1}^{m} w2_{i,j}u_i(k) + \sum_{i=1}^{r} w1_{i,j}c_i(k)) \quad (23)$$

$$c_i(k) = x_i(k-1) \quad (24)$$

$$y_j(k) = g(\sum_{i=1}^{r} w3_{i,j}x_i(k) \quad (25)$$

where, f(.) and g(.) entail linear, nonlinear outputs from hidden, output layers. Because internal connections only have dynamic properties of ENNs, it is critical to use states as inputs or training signals. Concept of ENN's is advantageous over static feed-forward networks. Due to unstable learning processes caused by lack of complete gradient information, ENN can only find one order linear dynamic system using typical backpropagation (BP) learning techniques, where only first-order gradients are accessible. These issues are handles by employing dynamic BP algorithm and another technique that modifies core ENN's architecture.

Modified ENN

To augment the dynamic features and convergence speed of the actual ENN, two improved network models are introduced. One is the modified ENN **(MENN)** illustrated in Fig. 3.

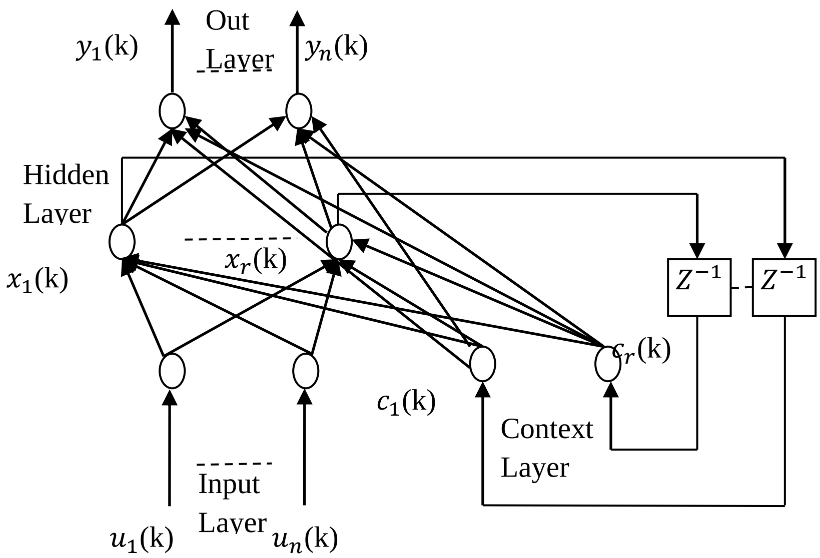

Fig 5. Modified architecture of ENN

The two networks may be distinguished when comparing Figs. 4 and 5 because the updated ENN has an auto feedback link with a constant gain in the context module. The output from the context layer at the kth iteration is therefore identical to the output from the hidden layer at the k-1st iteration, with the exception that the context layer's output is multiplied at the very end. The enhanced ENN reverts to its original configuration if the fixed gain is set to zero.

The enhanced ENN -based definition of the preceding equation's nonlinear state space is provided as:

$$x_j(k) = f(\sum_{i=1}^{m} w2_{i,j} u_i(k) + \sum_{i=1}^{r} w1_{i,j} c_i(k)) \quad (26)$$

$$c_i(k) = x_i(k-1) + \alpha \times c_i(k-1) \quad (27)$$

$$y_j(k) = g(\sum_{i=1}^{r} w3_{i,j} x_i(k)) \quad (28)$$

Gaussian weight updation function based ENN

The output and context nodes are separated by additional adjustable weights (w4ij) in this newly built MENN. The link in Fig. 2 between the context node i and the node j in the output layer stands in for the weight. It seems as though the output of the context layer serves as an input for the output layer. The suggested modified network model uses novel, adaptable weights known as the gaussian weight updation function (w4ij) between the context and output nodes. The nonlinear state space equation is presented in the figure below.

$$x_j(k) = f(\sum_{i=1}^{m} w2_{i,j} u_i(k) + \sum_{i=1}^{r} w1_{i,j} c_i(k)) \quad (29)$$

$$c_i(k) = x_i(k-1) \quad (30)$$

$$y_j(k) = g(\sum_{i=1}^{r} (w3_{i,j} x_i(k) + w4_{i,j}\, c_i(k)) \quad (31)$$

It can be inferred from above evaluation that improved ENN exhibits proportional and integral characteristics where proportional gains and integral coefficients are changed by adaptive weights. In contrary to the common PID algorithm, the regulated increase of the improved network is not just modified as the input module changes, but also causes reinforce mentor restrain on the output at the finalized moment with a + wl weight function. Ifa + wl> 1, it will extend the control output at the final time; when a + wl<1 , it will decrease the control output at the finalized time; and when a + wl = 1, it becomes the standard variable-parameters. If a =0, and w4 f 0 (respective to the second improved ENN), or when a f 0 and w4=0(with respect to the first improved ENN), the network maintains the proportional-integral properties. Also if -0 and w4=0, the improved ENN deteriorates to the fundamental Elman one, so the expression becomes:

$$y(k) = w1 \times y(k-1) + w2 \times w3 \times u(k) \quad (32)$$

This results in the usual integral equation. Its dynamic reaction will thus become slower, which will reduce the pace of convergence. According to the aforementioned theoretical analysis, the upgraded ENN exhibits superior dynamic properties than the basic ENN.

3.5.2. GNN

The foundation of the GNN concept is mostly provided by artificial neural networks that are capable of processing input that is either numerical or granular in nature [29]. The GNN approach emphasises the use of data sequences for live incremental learning. As shown in Fig. 6, learning in GNN and GNN follows a generic notion with two phases. Information granules are first constructed in accordance with a real numerical representation. These granules might be intervals or, more typically, fuzzy sets.

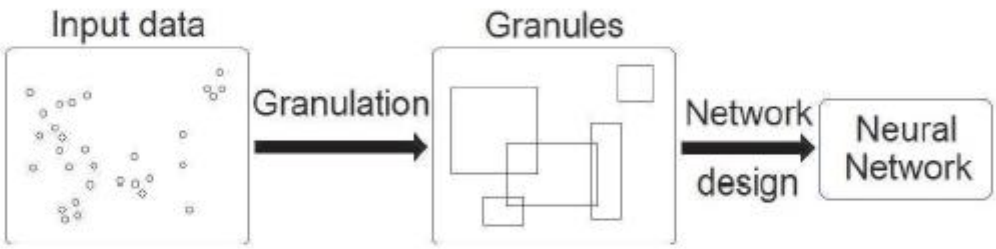

Fig 6. Two-step structure of GNN

In essence, GNN helps handle data sequences by using a rapid incremental one-pass data learning method. For the GNN to begin learning, no prior understanding of the statistical characteristics of data and classifications is required. The technique involves using fuzzy hyper boxes to generate decision boundaries across classes by granulating the feature space. The following are the main traits listed. A GNN:

- can handle labelled and unlabeled samples simultaneously;
- can adjust its architecture and settings for learning new paradigms, and eliminates from memory what no longer has relevance.
- the ability to recognise changes and handle unpredictability in the data; the capacity for nonlinear separation;
- the presentation of perpetual learning using both constructive bottom-up and destructive top-down tactics;

GNN Structure and Processing

GNN learns from datasets $x[h], h = 1, 2, \ldots$ Class label $C[h]$ and accompanying training samples may or may not exist. Information granules γ_i of finite sets of granules $\gamma = \{\gamma 1, \ldots, \gamma c\}$ present in feature spaces $X \subseteq Rn$ corresponds to classes Ck of finite sets of classes $C = \{C1, \ldots, Cm\}$ present in output spaces $Y \subseteq N$. The GNN helps in the connection between the feature and output spaces with the help of granules obtained using the data sequence along with a layer of T-S neurons [30]. The neural network includes a 5-layer architecture as depicted in Fig. 7. The input layer in facts ends the feature vectors $x[h] = (x1, \ldots, xj, \ldots, xn)[h], h = 1, \ldots$ into the network; the granular layer includes the set of information granules $\gamma i \forall i$ created within the field of the feature space. Granules are let to have partial overlapping; the aggregation layer includes null neurons $TSni\forall i$. They aggregate the membership values for generating values $oi\forall i$ indicating the compatibility of class between examples and granules; the decision layer performs the comparison between the compatibility values o_i, and the class $\bar{C}_k$ related to the granule γ_i having the maximum compatibility value becomes the output; the output layer constitute the class label markers. All the layers, with the exception being the input layer, develops as $x[h], h = 1, \ldots$, forms the input.

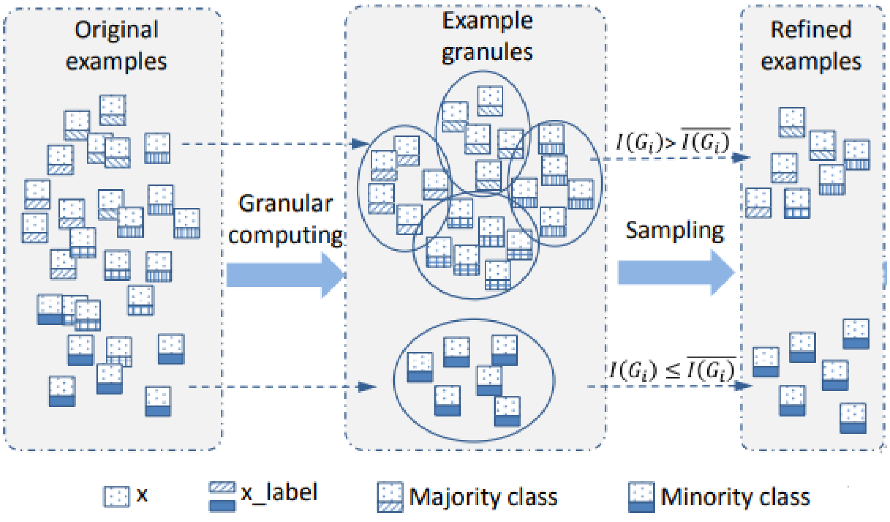

Fig. 7. Block diagram of the developing granular neural network used for classification

There are several approaches to structurally and parametrically tailor the GNN classifier to the needs of the application. For instance, it may be automatically regulated if course counts are known in advance. Granule counts used in the model design may be constrained if memory and processing time are the constraints.

3.5.3. Kernel based Support Vector Machine (KSVM)

An SVM model is typically a MLT, which depends on statistical learning principles. It helps classifying the data using a bunch of support vectors representing the data patterns. A common two-class classification problem involves finding a discriminant function f(x), such that $y_i = f(x_i)$ with N data samples $(x_1, y_1)\ldots(x_i, yi)\ldots(xN, yN)$ provided. A likely linear discriminant function is defined by f(x) = sgn(w • x b), where w • x b = 0 is a hyperplane that separates in the data space. As a result, choosing a discriminant function entails locating a hyperplane with the greatest separating margin between the two classes [31]. The final linear discriminant is calculated as f(x) = sgn(Pli=1 αiyi(xi · x − b), where l stands for counts of training records, yi∈ {−1, +1} indicates the label corresponding to the training data, 0 ≤ αi ≤ C (constant C > 0), and xi refers to the support vectors. If the surface separating the two classes is non-linear, the data points can be changed to a higher dimensional space where they will be separable linearly. SVM's nonlinear discriminant function is written as:

$$f(x) = sgn(\textstyle\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b), \quad (33)$$

where K(xi, x) denotes the kernel operation that was used to convert the data points. The linear function, polynomial function, sigmoid function, and radial basis function are a few of the well-known kernel functions. These kernel functions don't take into consideration variations across data characteristics. Standard SVM kernel function format K(xi, x), with equal treatment of all relevant training and test dataset properties. valuing each quality equally might have a negative impacts on SVM accuracies, which could be hazardous. By adding weights into kernel functions, it is possible to account for saliency of different features [32]. Weights aid in determining relative importance of properties. K(wxi, wx) are common expressions for new kernel functions, where w is a vector carrying weights for data set characteristics. A nonlinear discriminant function with feature weights is constructed as follows.:

$$f(x) = sgn(\textstyle\sum_{i=1}^{l} \alpha_i y_i K(wx_i, wx) + b), \quad (34)$$

This kernel doesn't display any dependence on certain kernel operations. The ideal kernel function may be chosen for applying the feature weights on in the case of varied applications. Using rough set theory, the training data is utilised to create and calculate these weights. The following are the fundamental concepts behind weight computation: 1) When a feature is absent from all reducts, its weight is set to zero; 2) the more times a feature occurs in a reduct, the more essential it becomes; and 3) the fewer features in a reduct, the more important the features that are there. When a reduct has only one feature, that feature is the most important.

As a result, the ELM based classification strategy is a very useful method for accurately recognising bogus news.

4. Results and Discussion

The efficiency of the suggested model is evaluated by a series of experiments carried out on the FNC-1 datasets provided below. This section describes these trials and contrasts the results with those of other cutting-edge techniques. Several datasets have been made available for the goal of identifying fake news. One of the key conditions for adopting neural networks is the availability of a substantial dataset for model training. In this study, a dataset made up of several texts was taken from Kaggle and used to train the deep models.

The dataset that is provided is used to evaluate this system's performance, and it is compared to recent techniques such as CSI (Capture, Score, and Integrate), CNN (Convolution Neural Network) with LSTM (Long Short Term Memory), ELM, and proposed MBDFO with ELM. According to true positive (TP), false positive (FP), true negative (TN), and false negative (FN) rates, the following section provides a list of experiment-specific assessment metrics. The percentage of pertinent events among those returned is the first performance metric, called accuracy. The second performance metric, recall, is the proportion of relevant and returned instances. Despite the fact that they frequently have opposing qualities, the measurements of accuracy and recall are both necessary for evaluating the efficacy of a prediction strategy. Since these two measurements may be joined with the same weights, the F-measure is created. The accuracy component of the final performance metric is calculated as the ratio of accurately predicted events to all anticipated occurrences.

Precision is given by the rightly obtained positive observations divided by all the predicted positive observations.

$$\text{Precision} = TP/(TP+FP) \tag{35}$$

Sensitivity or Recall is given by the proportion between the rightly predicted positive observations and the total observations.

$$\text{Recall} = TP/(TP+FN) \tag{36}$$

F - measure yields the weighted average of Precision and Recall. Consequently, it uses false positives and false negatives

$$\text{F1 Score} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \tag{37}$$

Accuracy is computed using positives and negatives as below:

$$\text{Accuracy} = (TP+FP)/(TP+TN+FP+FN) \tag{38}$$

Table 1. results of the Performance comparison analysis between proposed and available techniques for the considered FNC-1 dataset

Performance Metrics	CSI	CNN-LSTM	ELM	MBDFO-ELM
Accuracy	85.15	97	98	98.64
Precision	82	96	98	99
Recall	84.23	96.70	97.90	98.14
F-measure	83	96	97.5	99

The performance comparison study between the suggested and recent methodologies for the given FNC-1 dataset is summarised in Table 1 along with the findings. The table shows that, when compared to current false news detection methods, the suggested ELM model has the highest detection accuracy.

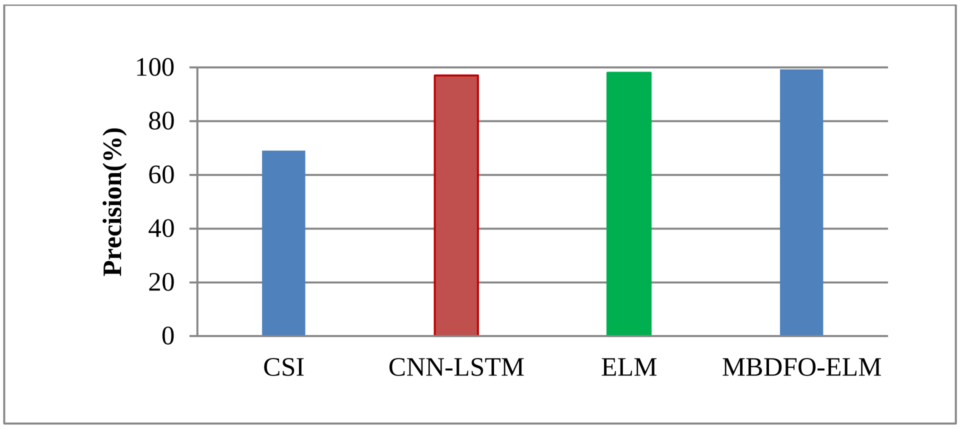

Fig.8. Precision comparison results between the proposed and available fake news detection model

The fig.8. illustrates that the precision comparison results between the proposed and existing fake news detection model. By analyzing all the results, one can conclude that using MBDFO is more effective for severe dimensionality reduction as it significantly improves the accuracy. The presented model outperforms all other models by producing a precision of 99%.From the results it concludes that the proposed MBDFO-ELM technique has high precision results compared to the existing classification techniques.

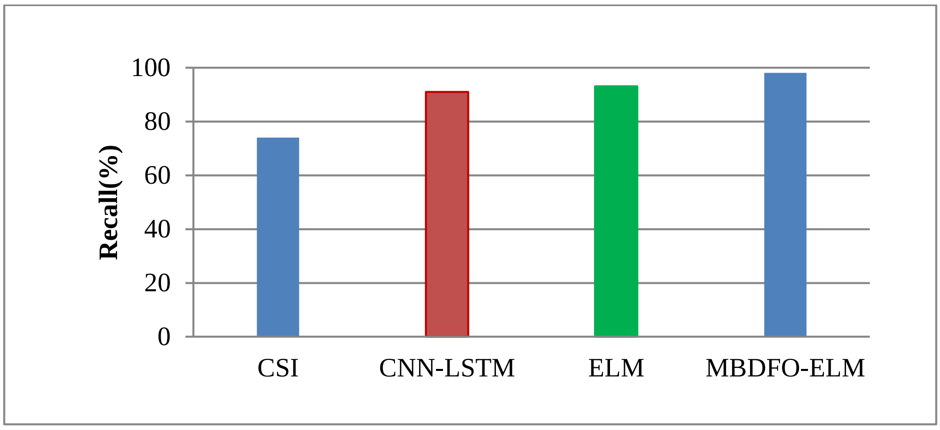

Fig.9. Results of the recall comparison between the proposed and recent fake news detection models

Figure 9 shows the outcomes of a comparison analysis of the proposed and existing fake news detection methods in terms of recall. The statistical significance guarantees that the suggested approach is successful in categorising any news as true or fake. The results show that, when compared to the existing models, the suggested MBDFO-ELM model has a 98.14% recall value.

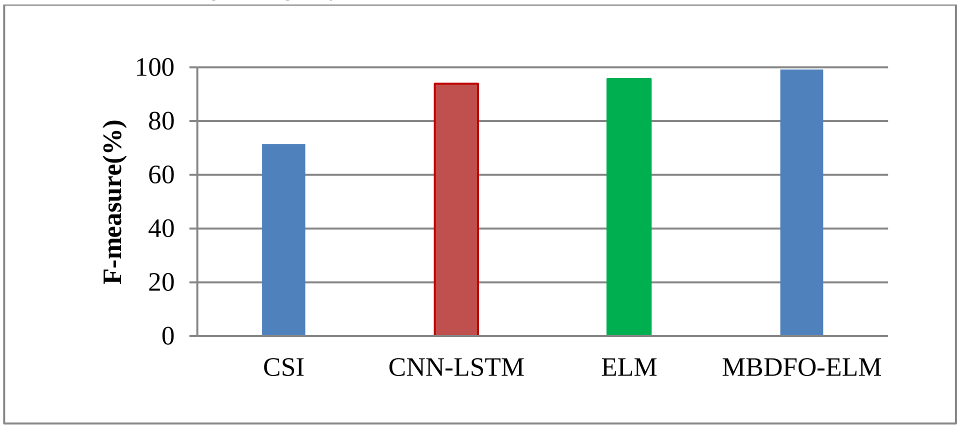

Fig.10. F-measure comparison results between the proposed and existing fake news detection model

In terms of F-measure, Figure 10 displays the results of a comparison analysis of the recommended and most recent fake news detection algorithms. It is feasible to infer a significant improvement in F1-score, accuracy, and recall.

The time needed to finish a prediction is also greatly decreased when employing the proposed MBDFO-based feature extraction method. The findings indicate that the proposed MBDFO-ELM methodology outperforms the existing applied classification strategies in terms of F-measure values.

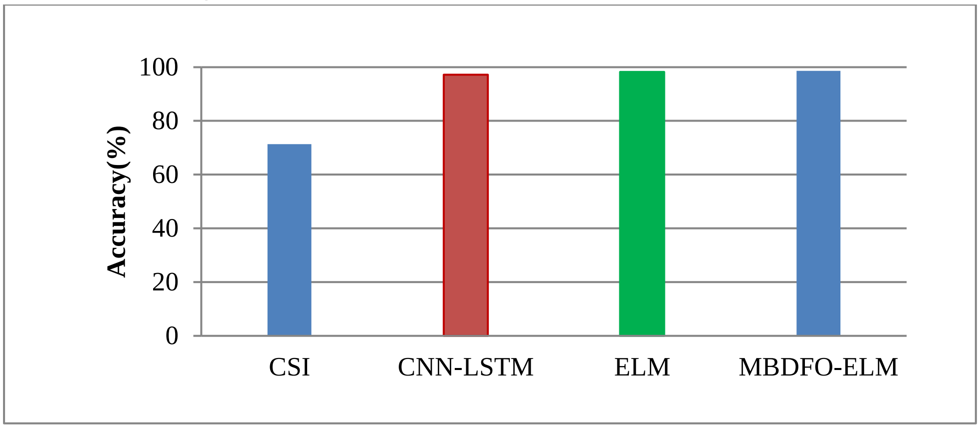

Fig.11. Accuracy comparison results between the proposed and existing fake news detection model

The findings of a comparison examination into the efficacy of the present and proposed fake news detection methods are shown in Figure 11. According to the findings, the accuracy achieved when the characteristics are employed without data cleansing or preprocessing is a significantly lower 78%. It is an indication that the actual dataset is full of noisy, redundant, and discontinuous data. The accuracy increases to 98.64% when the preparation procedures are finished and extraneous data is removed. The findings demonstrate that the proposed MBDFO-ELM approach is capable of greater accuracy values in comparison to existing classification strategies.

5. Conclusion

The identification of fake news is still a major worry and has a significant impact on our modern culture. Even when MLTs are taken into account, the prediction and detection of false news was proven to be a challenging problem. An intelligent feature extraction and ensemble learning system for successfully identifying bogus news is provided in this technical paper. This innovative algorithm uses a four-step technique to identify bogus news on social media. The first step in the procedure pre-processes the data set to turn it from an unorganised set into an ordered set. The second phase employs Feature Extraction using MBDFO, which is brought on by the unidentified traits of false news and the variety of linkages between news pieces. The final stage offers feature reductions depending on FPSO choices. Finally, an ELM is developed to teach students how to successfully identify false news and represent news stories. A series of tests on the aforementioned FNC-1 datasets are used to evaluate the efficiency attained with the suggested model. This research suggests that while the suggested model's performance is not as effective as that of a single model, it offers greater resilience and adaptability than the models that are currently available. Therefore, it can be said that the suggested model is an effective method for identifying and estimating fake news based on the dataset that was selected. Future study will involve evaluating the proposed false news detection algorithm using larger datasets.

References

[1] Parikh, S. B., &Atrey, P. K. (2018, April). Media-rich fake news detection: A survey. In 2018 IEEE conference on multimedia information processing and retrieval (MIPR) (pp. 436-441). IEEE.

[2] Oshikawa, R., Qian, J., & Wang, W. Y. (2018). A survey on natural language processing for fake news detection. arXiv preprint arXiv:1811.00770.

[3] Zhou, X., &Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. ACM Computing Surveys (CSUR), 53(5), 1-40.

[4] Cao, J., Qi, P., Sheng, Q., Yang, T., Guo, J., & Li, J. (2020). Exploring the role of visual content in fake news detection. Disinformation, Misinformation, and Fake News in Social Media: Emerging Research Challenges and Opportunities, 141-161.

[5] Alonso, M. A., Vilares, D., Gómez-Rodríguez, C., &Vilares, J. (2021). Sentiment analysis for fake news detection. Electronics, 10(11), 1348.

[6] Zhang, X., &Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management, 57(2), 102025.

[7] Awan, M. J., Yasin, A., Nobanee, H., Ali, A. A., Shahzad, Z., Nabeel, M., ... & Shahzad, H. M. F. (2021). Fake news data exploration and analytics. Electronics, 10(19), 2326.

[8] Bhutani, B., Rastogi, N., Sehgal, P., &Purwar, A. (2019, August). Fake news detection using sentiment analysis. In 2019 twelfth international conference on contemporary computing (IC3) (pp. 1-5). IEEE.

[9] Kula, S., Choraś, M., Kozik, R., Ksieniewicz, P., &Woźniak, M. (2020). Sentiment analysis for fake news

detection by means of neural networks. In Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part IV 20 (pp. 653-666). Springer International Publishing.

[10] Bondielli, A., &Marcelloni, F. (2019). A survey on fake news and rumour detection techniques. Information Sciences, 497, 38-55.

[11] Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. In Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments: First International Conference, ISDDC 2017, Vancouver, BC, Canada, October 26-28, 2017, Proceedings 1 (pp. 127-138). Springer International Publishing.

[12] Amer, E., Kwak, K. S., & El-Sappagh, S. (2022). Context-based fake news detection model relying on deep learning models. Electronics, 11(8), 1255.

[13] Seddari, N., Derhab, A., Belaoued, M., Halboob, W., Al-Muhtadi, J., &Bouras, A. (2022). A hybrid linguistic and knowledge-based analysis approach for fake news detection on social media. IEEE Access, 10, 62097-62109.

[14] Akinyemi, B., Adewusi, O., &Oyebade, A. (2020). An improved classification model for fake news detection in social media. Int. J. Inf. Technol. Comput. Sci, 12(1), 34-43.

[15] Kaliyar, R. K., Goswami, A., & Narang, P. (2021). EchoFakeD: improving fake news detection in social media with an efficient deep neural network. Neural computing and applications, 33, 8597-8613.

[16] Albahar, M. (2021). A hybrid model for fake news detection: Leveraging news content and user comments in fake news. IET Information Security, 15(2), 169-177.

[17] Bauskar, S., Badole, V., Jain, P., & Chawla, M. (2019). Natural language processing based hybrid model for detecting fake news using content-based features and social features. International Journal of Information Engineering and Electronic Business, 11(4), 1-10.

[18] Reis, J. C., Correia, A., Murai, F., Veloso, A., &Benevenuto, F. (2019). Supervised learning for fake news detection. IEEE Intelligent Systems, 34(2), 76-81.

[19] Zhou, X., Jain, A., Phoha, V. V., &Zafarani, R. (2020). Fake news early detection: A theory-driven model. Digital Threats: Research and Practice, 1(2), 1-25.

[20] Jarrahi, A., & Safari, L. (2023). Evaluating the effectiveness of publishers' features in fake news detection on social media. Multimedia Tools and Applications, 82(2), 2913-2939.

[21] Sitaula, N., Mohan, C. K., Grygiel, J., Zhou, X., &Zafarani, R. (2020). Credibility-based fake news detection. Disinformation, misinformation, and fake news in social media: Emerging research challenges and Opportunities, 163-182.

[22] Yuan, H., Zheng, J., Ye, Q., Qian, Y., & Zhang, Y. (2021). Improving fake news detection with domain-adversarial and graph-attention neural network. Decision Support Systems, 151, 113633.

[23] D. Pomerleau and Rao. (2017). Fake News Challenge Dataset. Accessed:Oct. 29, 2019. [Online]. Available: http://www.fakenewschallenge.org/.

[24] Meraihi, Y., Ramdane-Cherif, A., Acheli, D., &Mahseur, M. (2020). Dragonfly algorithm: a comprehensive review and applications. Neural Computing and Applications, 32, 16625-16646.

[25] Mafarja, M., Aljarah, I., Heidari, A. A., Faris, H., Fournier-Viger, P., Li, X., &Mirjalili, S. (2018). Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowledge-Based Systems, 161, 185-204.

[26] Bai, Q. (2010). Analysis of particle swarm optimization algorithm. Computer and information science, 3(1), 180.

[27] Lingras, P., & Jensen, R. (2007, July). Survey of rough and fuzzy hybridization. In 2007 IEEE International Fuzzy Systems Conference (pp. 1-6). IEEE.

[28] Ren, G., Cao, Y., Wen, S., Huang, T., & Zeng, Z. (2018). A modified Elman neural network with a new learning rate scheme. Neurocomputing, 286, 11-18.

[29] Song, M., &Pedrycz, W. (2013). Granular neural networks: concepts and development schemes. IEEE transactions on neural networks and learning systems, 24(4), 542-553.

[30] Pedrycz, W., &Vukovich, G. (2001). Granular neural networks. Neurocomputing, 36(1-4), 205-224.

[31] Kobayashi, K., & Komaki, F. (2006). Information criteria for support vector machines. IEEE transactions on neural networks, 17(3), 571-577.

[32] Chung, K. M., Kao, W. C., Sun, C. L., Wang, L. L., & Lin, C. J. (2003). Radius margin bounds for support vector machines with the RBF kernel. Neural computation, 15(11), 2643-2681.

[33] Gopalakrishnan, S. & Ganeshkumar, P. (2015). Secure and Efficient Transmission in Mobile Ad hoc Network to Identify the Fake ID's Using Fake ID Detection Protocol. Journal of Computer Science, 11(2), 391-399. https://doi.org/10.3844/jcssp.2015.391.399

[34] Salman Al-Nuaimi, M. A. ., & Abdu Ibrahim, A. . (2023). Analyzing and Detecting the De-Authentication Attack by Creating an Automated Scanner using Scapy. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2), 131–137. https://doi.org/10.17762/ijritcc.v11i2.6137

[35] María, K., Järvinen, M., Dijk, A. van, Huber, K., & Weber, S. Machine Learning Approaches for Curriculum Design in Engineering Education. Kuwait Journal of Machine Learning, 1(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/111