**Enhanced Large language model with a self-attention mechanism for Social Media Rumor Detection**

**1\*Vikas B, 2Dr. J Sirisha Devi, 3Dr. L Sukanya, 4Sree Lakshmi Done, 5Dr. Rayapati venkata sudhakar and 6D.Venkateswarlu**

1\*Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Bowrampet, Hyderabad-500043, Telangana, India.

2Professor, Department of Computer Science & Engineering, KL University, Bowrampet, Hyderabad, Telangana, India.

3Assistant Professor, Department of Computer Science & Engineering, KL University, Bowrampet, Hyderabad, Telangana, India.

4Assistant Professor, Department of Computer Science and Technology, G. Narayanamma Institute of Technology and Science (For Women), India.

5Associate professor, Department of Computer science and Engineering, Geethanjali college of Engineering and Technology, Cheeryal, Medchal District , Hyderabad-500043, Telangana, [India.](mailto:India,%20rayapati1113@gmail.com)

6Associate Professor, Department of computer science and engineering, Geethanjali College of engineering and technology, Hyderabad, Telangana, India.

Corresponding author email: vikasbresearch@gmail.com

**Abstract**

In today's digitally interconnected world, the unchecked spread of rumors and misinformation presents significant challenges to public discourse, trust, and decision-making. Rumors, often based on unsubstantiated information, speculation, or exaggerated accounts, can rapidly disseminate through social media and other communication channels, particularly in uncertain or anxious situations. False rumors not only damage reputation and trust but also have tangible economic consequences, affecting businesses and markets.This research aims to provide a robust method for detecting rumors on social media platforms, addressing urgent requirements for effective techniques and tools to mitigate their impact. Initially, data are collected from social media and pre-processed. Word embeddings are then used to extract features from the preprocessed data, providing rich, context-aware vector representations that capture essential semantic and syntactic nuances for accurate classification. The extracted features are subsequently classified using a Generative Pre-trained Transformer (GPT), which relies on a transformer architecture that employs a self-attention mechanism to weighthe significance of unusual terms in a sequence comparative to each other. Finally, hyperparameters are optimized using the Deer Hunting Optimization Algorithm (DHOA). For experimental analysis, three types of datasets are utilized. The proposed method achieves high accuracy rates of 99.5% for Twitter15, 99.30% for Twitter16, and 99.1% for the PHEME dataset.

**Keywords:** Social media, Rumor Detection, Generative Pre-trained Transformer, Deer Hunting Optimization Algorithm.

**1. Introduction**

Conventionally defined, a rumor could be a revelation of data unverified throughout the time of communication [1]. Social media is now a useful medium for people to interact with one another, share information, and express ideas thanks to the rapid growth of the Internet[2]. Rumors sometimes may spread very quickly over social media platforms, and rumor detection has gained great interest in both academia and industry recently [3]. With the rapid growth of social media such as Twitter and Weibo, information campaigns are frequently carried out by misinformation producers with various commercial and political purposes [4]. The wide availability of posting channels in online social platforms such as Twitter, Weibo, Instagram, and Facebook facilitates the propagation of massages [5].

As Social media finds use as an information source by both public and professionals, it finds applications in various domains [6]. Numerous efforts were undertaken to dispel rumors and lessen their negative impact. Fact-checking services like Politifact.com and Snopes.com rely on human labor to search out and disprove myths, which clearly limits their coverage and effectiveness [7]. The rapid development of technology in recent years, especially in mobile phones made social networks available at all time and information spreading has become faster than ever [8]. Today, the most popular information access environment is undoubtedly social networks. This is due to an enormous amount of data and huge social network relationships [9]. On Twitter, news can be shared without proper restriction or verification leading to a propagation of unconfirmed and unverified statements. Such fake news or rumors can cause public panic and social disturbance [10].

Our study focuses on the task of determining whether a tweet is a rumor or not. In order to distinguish between rumored and non-rumored tweets, the task of detecting rumors is seen as a binary classification issue. Data spreads across time on social media, yet the majority of rumor detection methods now in use miss this important aspect or are unable to record the temporal dimension of information[11]. It is thus the utmost interest of the social network platforms to develop effective strategies to combat fake news and rumors[12]. Social media rumors can be extremely detrimental to the neighborhood. For instance, on April 23, 2013, the Associated Press's official Twitter handle had been compromised, leading to the publication of false information claiming that the President was hurt in two detonations at the White House. This promptly caused the stock market to plunge sharply[13]. In order to identify rumors on the internet, the early research mostly used conventional machine learning techniques based on manually created data, such user profiles and text elements[14].The list of Key contributions explained below,

**Feature Extraction with Embeddings:** Word embeddings are used to extract features from preprocessed data, providing RoBERTa context-aware vector representations that capture essential semantic and syntactic nuances for accurate classification.

**Classification Using GPT:** The extracted features are classified using a GPT. GPT relies on transformer architecture with a self-attention mechanism, which weights the importance of different terms in a sequence relative to each other. This allows the model to capture long-range dependency and contextual dealings within the text.

**Hyperparameter Optimization with DHOA:** Hyperparameters are optimized using the DHOA, enhancing the model's performance and accuracy.

The remainder of this paper is organized as follows: Section 2 reviews the findings from the literature study. Section 3 provides a detailed description of the proposed method, which is divided into three subsections. Section 4 presents the experimental results, demonstrating the effectiveness of the proposed method and validating its superiority. Finally, Section 5 concludes the paper with a summary of the findings and suggestions for future research.

**2. Literature Review**

Many researchers have developed rumour detection on social-media. Among them, some of the works are analysed here;

Gao et al. [15] proposed contrasting self-supervised learning across diverse sources of data to better classify rumors and uncover their relationships. In terms of mathematics, they augment the primary supervised task of recognition with a supplementary self-supervised task called subsequent-self-discrimination, that aims to improve post descriptions. To create views and perform the discriminating, we develop cluster-wise and instance-wise methods while taking various relationships between sources of knowledge into account.

Ke et al. [16] proposed KZWANG is a rumor system of detection that symmetry-fuses semantic data, a propagating heterogeneity graph, and enough domain knowledge to categorize rumors with accuracy. To learn the semantic makeup of the text, they employ an attention mechanism. They then propose a global community network (GCN) to capture the global and local interactions among all the source microblogs, reposts, and users. A rumor detection classifier is then trained using an organic mix of propagating heterogeneous networks and text semantics. Tests conducted on rumor monitoring datasets from Sina Weibo, Twitter15, and Twitter16 show that the suggested model outperforms baseline techniques.

Xu et al. [17] proposed a An innovative Graph Neural Network (GNN) oriented rumor detection technique called Hierarchically Aggregated Graph Neural Networks (HAGNN). The goal of this challenge is to combine the rumor transmission structure with various granularities of high-level representations of text content. Using a graph of rumor transmission, it employs a Graph Convolutional Network (GCN) to develop text-granularity representation with the spreading of events. In order to generate final reconstructions of what happened and identify rumors. Tests conducted on two authentic datasets show that the suggested approach outperforms the baseline techniques.

Shi et al. [18] proposed to use a machine learning method on the Weibo network to fix this issue. Initially, they take textual attributes, user-associated features, interaction-based features, and emotion-based elements out of the COVID-19 distributed messages. Second, they use the ensemble learning approach to construct an intelligent rumor detection model by merging these four types of characteristics. Lastly, we do in-depth tests using the Weibo data that we have gathered.

Shelke et al. [19] proposed to stop the spread of rumors in order to reduce their negative consequences on community. In spite of intense attempts to address this problem, researchers mostly concentrated on the temporal dynamics of postings and other variables that show a reasonable level of accuracy, such as user, network, and content-based features. The additional attributes connected to each post are suppressed by an event that is linked to the point in time series characteristics. This research emphasizes post-wise features, including user-based, content-based, and lexical-based characteristics in addition to post sequences, because there is room for accuracy improvements. They suggested a framework that combines two deep learning models with a number of key characteristics. To increase accuracy, word embedding is paired with post-wise characteristics utilizing a multilayer perceptron (MLP) and bidirectional long short-term memory (BiLSTM).

Gao et al. [20] proposed the message-level job of ERD. In order to simulate the transmission of trends in rumors in their earliest phases, they have developed a unique hybrid neural network architecture that combines stacked LSTM networks with a task-specific character-based bidirectional language model to capture the textual content and social-temporal settings of tweet input sources. To jointly learn attentive context embeddings over numerous contextual inputs, they employ multi-layered consideration models. A rigorous leave-one-out cross-validations (LOO-CV) assessment system is used in our research on seven easily accessible real-world rumor events data sets.

Tian et al. [21] proposed research presents the R-CNN model, a convolutional neural network (CNN) based framework for forecasting rumor retweeting behaviour. Four feature vectors—attention to public emergencies, attention to rumors, reaction time, and tweeting frequency—are constructed using actual textual content released by consumers. The model views rumor responding to behavior as a significant driving force behind the expansion and depth of rumor a cascade. A K-means based core tweets extract approach is provided to choose the appropriate tweets, and statistical representations of features are proposed to input the quantitative feature vectors for R-CNN. The model's prediction power has been demonstrated through studies conducted on two emergency rumor datasets that were scraped from Sina Weibo.

Alkhodair et al. [22] proposed the challenge of identifying rumors that are the latest developments rather than persistent ones that propagate on social media. Our novel method automatically detects rumors by simultaneously learning embeddings of words and training a recurrent neural network with two distinct goals. The suggested approach of reducing the subject shift problems is straightforward but efficient. New rumors don't always have to be untrue when they first surface. These could turn out to be true or untrue in the future. But the majority of earlier research on rumor detection concentrates on persistent rumors and makes the assumption that rumors are always untrue. On the other hand, our experiment uses a real-life rumor dataset to replicate a cross-topic developing rumor recognition scenario.

The reviewed studies reveal advancements in rumor detection leveraging various neural network models and feature extraction techniques. However, a research gap exists in effectively integrating multi-modal data, such as visual and textual content, to enhance detection accuracy. Additionally, there's a need for models that can adapt dynamically to emerging rumor patterns and diverse social media platforms, ensuring robustness and scalability. Furthermore, real-time detection remains a challenge, requiring efficient processing to mitigate the rapid spread of misinformation.

**3. Proposed rumor detection Methodology**

The proposed method for detecting rumors on social media platforms begins with the collection and preprocessing. Preprocessing involves cleaning the data to remove noise and irrelevant information. Next, word embeddings are utilized to extract features from the pre-processed data, creating rich, context-aware vector representations that capture essential semantic and syntactic nuances. These vectorized features are then fed into a GPT for classification. GPT employs transformer architecture with a self-attention mechanism, allowing model to weigh importance of different terms in a sequence relative to each one-time, thus capturing long-range dependencies and contextual relationships within the text. To further enhance the model's performance, hyperparameters are optimized using the DHOA. This optimization step fine-tunes the model to improve its accuracy and efficiency. Finally, the proposed method is implemented and evaluated in Python, where its performance is assessed based on its ability to accurately differentiate between true and false information. Figure 1 shows a flow diagram of social media rumor detection.

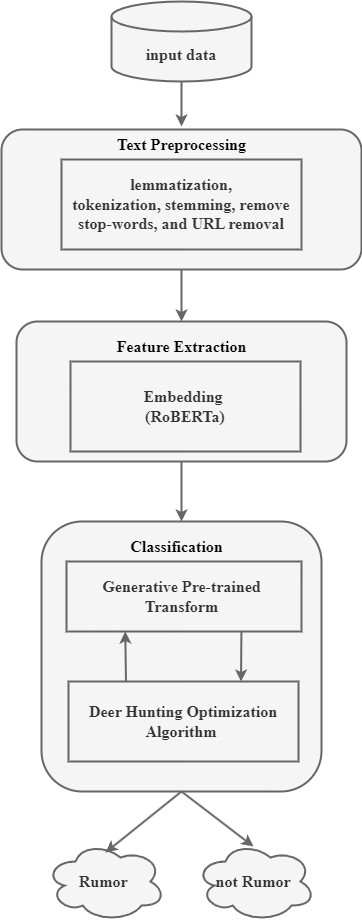


Figure 1: Flow diagram of Social Media Rumor Detection

**3.1. Text Preprocessing**

This section focused on text preprocessing, which is the first step in the text classification process. Text preparation involves putting the text into a format that is predictable and easy to analyze for a job. A great deal of unnecessary data, including lemmatization, tokenization, stemming, remove stop-words, and URL removal. Text preprocessing is often the first step in Natural Language Preprocessing (NLP), and it might impact the system's general efficiency. By removing this unnecessary data, they get the data set ready for additional processing under text preprocessing. Pre-processing is used to simplify the input without impairing the ability to interpret or conclusion of the model.The results of this critical text classification subtask dictate the accuracy of the later stages. Preprocessing is done with the intention of using the outcome to increase text categorization's efficacy. The finest text classification outcomes may be achieved by careful text preparation. As such, it is imperative to identify the preprocessing techniques. Normalization processes are important in this situation. As previously said, the papers include a wide range of sounds; nevertheless, in this case, the main goal is to clean the data by applying four normalization algorithms. Below figure 2 shows the preprocessing steps.

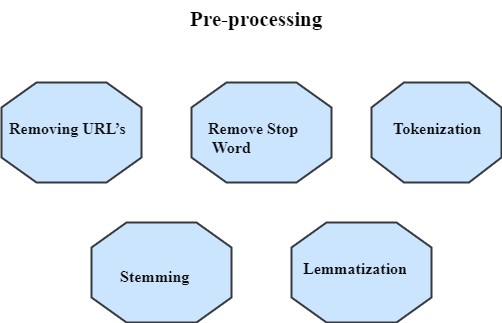


Figure 2: Pre-processing steps

**Removing URL’s:-** Among the particular goals of data cleaning are the elimination of duplicate URLs, the retrieval of active URLs, the preservation of URLs that contain HTML elements in the source code of the web page, and the inclusion of websites with information in English[23].

**Remove Stop Word**: - The stop words are eliminated once the URL has been deleted. Particularly, terms that are often used in English, such "a," "an," "the," and so on, are removed due of their decreased relevance in the text. As stop words have little effect on the important information collected, eliminating them aids in keeping attention on the pertinent weighted words and significant tokens[24].

**Tokenization: -** Splitting lengthy texts into digestible sections, such breaking sentences up into their component words. It's also known as text segmentation and lexical analysis[23].

h

**Stemming**: - The term "stemming" is a perplexing technique that searches a text for a common stem within a group of terms. They used the light 10 stemmer for the combining process and followed the same protocol, which is to remove "and" if the remaining piece of the word comprises three or more characters.

**Lemmatization: -** Words or tokens entered are reduced to their lemmas at this point. Lemmatization takes words down to their most basic form and uses a lexicon to confirm that the meaning is valid, whereas stemming only eliminates word suffixes. Lemmatized words retain their meaning when abbreviated, which is important for a model because one of its objectives is to provide summaries that are grammatically accurate. For instance, from "sharing," we derive the lemma "share" rather than the stem word "share." This method also eliminates the need for an additional post-processing step known as reverse stemming, which entails adding back the root words that were eliminated during stemming.

**3.2. Feature Extraction using RoBERTa**

Embedding is used to extract features from pre-processed data, delivering rich, context-aware vector representations of words that capture essential semantic and syntactic nuances, crucial for accurate classification.

**RoBERTa**

The digital encoder of bi-directional transformers is used as the central layer for gathering information in RoBERTa model, which inherits the benefits of the BERT model. Additionally, by enhancing the masking technique and eliminating the NSP job, the RoBERTa model outperforms the BERT model in terms of performance. The BERT model uses a static mask during the masking technique, which maintains the masking token constant. In contrast, RoBERTa uses a dynamic mask, meaning that throughout the training phase, the mask location varies continually. Figure 3 shows RoBERTa model.

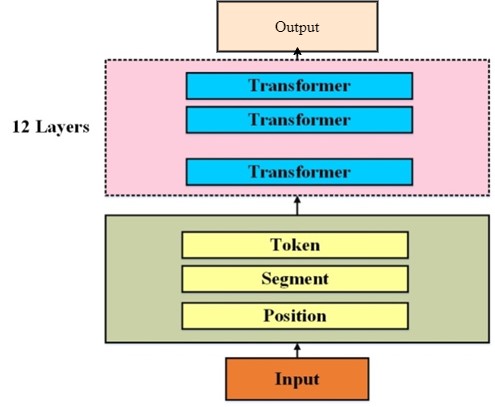


Figure 3: RoBERTa Model

The above enhances the data's unpredictability, which boosts the model's capacity for learning. Following the removal of the NSP job, RoBERTa uses uninterrupted complete sentences and document sentences as contribution. It then extends supplied sentence length to 512 characters, which is significantly greater than the BERT model's maximum of 256 characters. In brief, the RoBERTa model outperforms BERT in terms of effectiveness. Additionally, a few investigators enhanced the original BERT's single-character mask and suggest a whole-word mask system. A RoBERTa model is then suggested by combining the RoBERTa model to the whole-word mask method.

Throughout this work, contextual semantic information is extracted using RoBERTamodel, whose combines the benefits of RoBERTa model together complete word masking approach.

(1)

Every character's encoding representation is denoted by , its token embedding is represented by , its segment embedding is represented by , and its place in the embedding is indicated by .

That multi-head attention calculation is subsequently obtained by using an exponential transformation. Ultimately, the following equations can be used to determine self-attention versus multi-head attention:

(2)

(3)

Here , and represent the attention mechanism's weight matrices, and stand for the mechanism's input vectors.

The RoBERTa model's outputs layer's sentence vector (CLS), containing both contextual and global semantic information, may also be used to assess if the short text with the candidate entity's description text share the same semantic environment[25].

The output of using embeddings for feature extraction with the RoBERTa is rich, context-aware vector representations of words from the pre-processed data. These vectors effectively capture the semantic and syntactic nuances of the words, which are crucial for achieving accurate classification in various natural language processing tasks.

**3.3. Rumor detection using Generative Pretrained Transformer (GPT)**

A deep learning model based on transformers, GPT delivers state-of-the-art efficiency for language creation. The researchers who created the GPT simulation previously trained it on 40 gigabytes of words. Four sizes of previously trained modelssmall, with 117 million parameters and 12 layers as well; The full size has 48 layers and more than 1.5 billion parameters, while the medium has 24 layers and 345 million characteristics. The big size has 36 layers and 774 million characteristics—have been made available and released. Despite the expectation that bigger GPT models perform better, in this study we used a medium-sized GPT pre-trained model with 24 layers and 345 million parameter because of technological limitations.

GPT differs from RNN in that its transformer-based design is not dependent on recurrence during training to maintain contextual data. This feature eliminates the requirement for the structure of transformers to handle words individually while training with paragraphs. As a result, the training procedure is far more effective. Nevertheless, a lot of transformer-based models like GPT use the concept of autoregression, which is similar to the RNN technique, to generate texts. The method known as autoregression generates words progressively and concatenates freshly generated words together with previously created words to create new sequences. The following word will subsequently be generated using this new sequence as information. This part concentrates on text creation rather than the GPT training process because the study deals with NLG. Figure 4 shows the language generation GPT model.

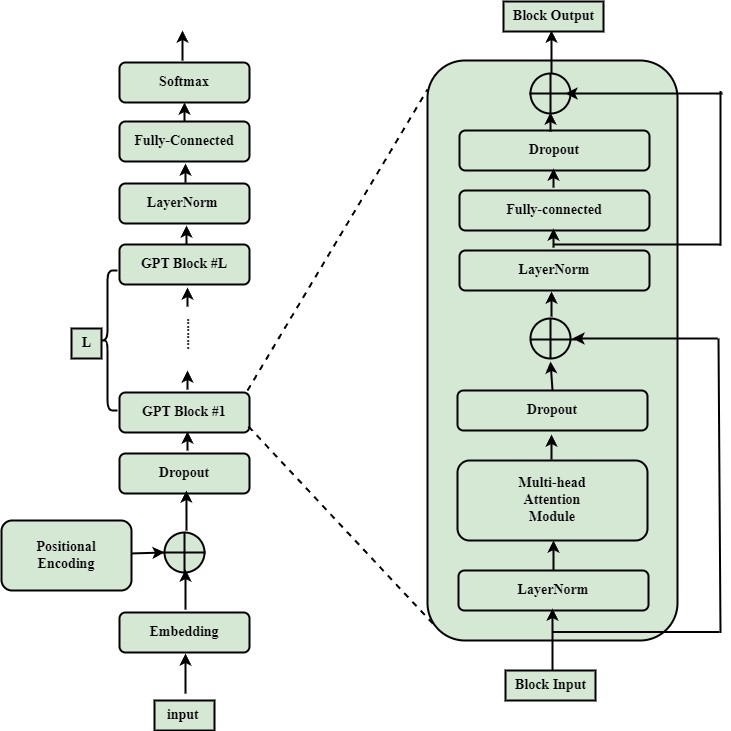


Figure 4: GPT Model

In order to create the rumor-related dataset, they initially collected rough data by scraping Twitter for frequently asked queries. They received more rumor questions after combining these queries. Everyone created suitable prompts (such as role descriptions, task descriptions, boundary definitions, and output formats) based on these rumor questions, and we used the GPT-Turbo-0301 models to generate that corresponds rumor titles, keywords, short and long rumor content, refutation responses, and correct answers for each question in turn, using roughly 3.36 billion tokens in total.These comprise health-related rumor titles, short rumor content, and long rumor content that were produced by the GPT using the initial health questions; the long rumor material is essentially an expanded version of the short rumor content. All of the terms that were taken from the original health questions are included in the keywords box, separated by commas. The right answers appropriately address the initial health inquiries, while the rebuttal responses define and explain the content of the produced rumors.

The output of classifying features using GPT is the predicted class of the input data. This involves generating contextually enriched feature vectors, mapping them to classification scores, normalizing these scores to probabilities, and selecting the class with the highest probability. The final output indicates the model's best guess based on the input features and context. The DHOA is chosen for hyperparameter optimization in this research because it efficiently explores the hyperparameter space to find optimal settings, ensuring that the enhanced large language model performs at its best. DHOA's adaptive search strategy balances exploration and exploitation, which is crucial for tuning complex models like those used in social media rumor detection. This leads to improved accuracy and robustness in identifying and classifying rumors on social media platforms.

**3.3.1. Hyper-Parameter Optimization Using DHOA**

Hyper-parameter optimization is procedure of determining best mix of GPT hyper-parameter settings to optimize performance on data in a reasonable quantity. This process is essential to GPT's capacity for precise result prediction. Most of this input text uses the hyperparameters' default values. The proposed model optimizes the hyperparameterusing the DHOA. Table 1 shows the hyperparameter initialization range.

Table 1: Range of hyperparameter initialization

|  |  |
| --- | --- |
| **Hyper-parameters** | **Range** |
| Learning Rate (L) | 1e-5 to 5e-4 |
| Batch Size (B) | 1 to 512 |
| Number of Layers (NL) | 1 to 48 |
| Number of Attention Heads (NAH) | 1 to 16 |
| Dropout Rate (DR) | 0.1 to 0.5 |

The DHOA [26]is used to optimize these hyperparameters. The Osprey Optimization algorithm's step-by-step procedure is explained below.

**Step 1: Initialization:** Selecting the ideal hyperparameter is the primary goal of this approach. Initially, define the top and lower bounds of the problem, the dimensionality D of the variables, the maximum number of iterations, and the Deer size N. Each solution represented by the Deer consists of hyperparameters, including L, B, NL, NAH, and DR. At first, a random selection is made. The first solution format is seen in the following equation:

 (4)

Here, is the Nth solution or Deer’s position

 (5)

Step 2: Calculating fitness: Following initialization, each solution's fitness is assessed using the DHOA technique that has been recommended. In this case, the classification accuracy is defined using fitness purpose. The solution with the highest fitness value is deemed to be the most effective. The fitness function is calculated with the assistance using,

 (6)

**Step 3: Updating using DHOA:** DHOA utilizes 3 distinct techniques known as three different methods of propagation: via the position of the leader, through a positioning angle, and through the position of the successor.

**Strategy 1: Propagation through a leader’s position:**The procedure of updating position starts when the optimal positions are determined, and every member of the community strives to get the best position. As a result, the following equation represents the encircling conduct:

 (7)

where  is where you are at this iteration,is location at next iteration, as well as are vectors of coefficients and  is a a random number between 0 and 2 that was created with the wind speed in mind. The measurement of the coefficient vectors is,

 (8)

 (9)

where  is maximum repetition,is parameter whose value ranges from -1 to 1 and  is a arbitrary integer falling between 0 and 1.

**Strategy 2: Propagation through position angle:** The idea is expanded by taking the position angle into account in the update algorithm, which improves the search space. To place the hunter so that the prey is not aware of the attack and the hunting process is successful, an angle computation is necessary. The calculation for the angle at which the prey or deer may be seen is as follows:

 (10)

A parameter that aids in updating the animal's position angle is derived based on discrepancy among wind angle and deer's visual angle.

 (11)

where  is the angle of wind. Next, during the subsequent iteration, the position angle is modified as,

 (12)

The position update is created as follows, taking the position angle into account.

 (13)

where represent the best location and  is random number. In order to keep the hunter out of the deer's line of sight, the individual's position is almost exactly opposite the position angle.

**Strategy 3: Propagation through the position of the successor:** By changing the vector, the same encircling behavior concept may be used throughout the exploration phase. Since the first search is assumed to be random, the vector's valuetaken into account is smaller than 1. As a result, successor location—rather than initial best explanation—is basis for the position update. This permits a worldwide search, as shown by the equation that follows:

 (14)

where, is the role that will replace the search agent in the existing workforce.   
Every iteration, the method adjusts the search agents' positions depending on the most effective solution found, starting with the initialization at random of alternatives. When, a search agent is picked at random, and optimal option is selected wheneverto update the agents' positions. Therefore, with vector's adaptive variability, suggested method alternates between the stages of exploitation and exploration. Furthermore, there are simply two parameters that need to be changed, i.e. along with, which adds to the benefit of the algorithm.

**Step 4: Termination condition:** The process is repeated until optimal hyper-parameter selection is made. The GPT receives the chosen hyperparameter value. Table 2 provides the deer hunting pseudocode.

|  |
| --- |
| Table 2: Pseudo code of DHOA |
| Input: initial population  Output: first-rate resolution and second-best option  start  while  for any population-wide solution  Determine whether each answer is fit  modernize, , , ,  and  If  if  Update the personnel status.  Else  Update that individual's status.  end if  Else  Update the person's status.  end if  end for  Determine whether every response is fit  Update  and  i = i + 1  end while  return  end |

**4. Result and Discussion**

The proposed model significantly outperforms baseline methods by integrating multi-modal data and dynamically adapting to emerging rumor patterns. Experiments on diverse social media platforms demonstrate its robustness and scalability, achieving higher detection accuracy. The model's real-time processing capabilities effectively mitigate the rapid spread of misinformation. The method outperformed existing approaches, with significant improvements in accuracy, precision, recall, and F-measure. The test is run on a machine equipped with an Intel (R) core (TM) i5 4570s CPU @ 2.90 GHz, \*GB RAM, and the computer name SSM107.smg.local running Windows 64-bit. Acer is the system manufacturer using the PYTHON tool. Our experimental configuration includes two data centers with four hosts and a total RAM of 8 GB. The host has a bandwidth of 2800 Mbps.

**4.1. Dataset Description**

For experimental analysis three types of datasets are utilized. A detailed explanation of the dataset is listed below;

**Twitter15 and Twitter16 datasets:** They test our methods using Twitter15 and Twitter16, two publicly accessible datasets that are commonly used as benchmarks in the rumor detection space. There are 1490 tweet propagations in the Twitter15 dataset and 818 tweet propagations in the Twitter16 dataset. One of four categories—true rumor, unconfirmed rumor, fake rumor, or non-rumor—is assigned to each tweet distribution.

**PHEME dataset:** A selection of Twitter rumors and non-rumors shared during breaking news are included in this dataset. It has rumors about nine different events, and each rumor has a veracity value—True, False, or Unverified—annotated with it. Table 3 shows statistics of Twitter15, Twitter16, and PHEME datasets.

Table 3: Statistics of Twitter15, Twitter16 and PHEME dataset

|  |  |  |  |
| --- | --- | --- | --- |
| **Statistic** | **Twitter 15** | **Twitter 16** | **PHEME** |
| # of source tweets | 1,491 | 817 | 6,425 |
| # of users | 276,664 | 173,486 | 12,425 |
| # of tweets | 331,614 | 204,821 | 65,322 |
| # of non-rumors | 375 | 204 | |  | | --- | |  |   2,572 |
| # of false-rumors | 371 | 204 | 1,678 |
| # of true-rumors | 373 | 206 | 1,973 |
| # of unverified rumors | 373 | 203 | 202 |

**4.2. Evaluation Metrics**

It has chosen many metrics to gauge how well predict the rumours. They have selected accuracy, precision, recall, and f-measure for our investigation. The confusion matrix was primarily used to determine the true positive, true negative, false positive, and false negative for the majority of the measurements. To evaluate these findings, compute the precision,recall,accuracy, and F1-score, FPR, FNR, MCC and NPV indicators.

****** (15)

 (16)

 (17)

 (18)

(19)

(20)

(21)

(22)

TP signifies the true positive, FP the false positive, TN the true negative, and FN the false negative.

**4.3 Comparative analysis based on feature extraction stage**

The comparative analysis in the feature extraction stage for social media rumor detection reveals that CBOW and TF-IDF show moderate accuracy and efficiency, while one-hot struggles with high dimensionality and computational complexity. The proposed method significantly outperforms these by effectively capturing contextual nuances and semantic relationships, leading to superior performance in rumor detection.

Table 4: Performance Evaluation of Proposed and Existing Feature Extraction Methods on Twitter 15 Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | Accuracy | Precision | Recall | F-measure |
| CBOW+GPT | 97 | 96.99 | 97.8 | 97 |
| TF-IDF+GPT | 95.46 | 94.98 | 95.34 | 95.3 |
| one-hot+GPT | 87.9 | 88.3 | 87.5 | 87.3 |
| Proposed | 99.5 | 99.57 | 99.6 | 99.59 |

The above table 4 shows that proposed method combined with GPT outperforms existing feature extraction techniques on the Twitter 15 dataset, achieving an accuracy is 99.5%, precision is 99.57%, recall value is 99.6%, and an F-measure of 99.59. Compared to CBOW+GPT, which has an accuracy of 97%, TF-IDF+GPT with 95.46%, and one-hot+GPT with 87.9%, the proposed method demonstrates superior performance across all metrics, indicating its effectiveness in capturing relevant features for improved classification. This significant improvement highlights the robustness and precision of the proposed feature extraction approach when integrated with GPT.

Table 5: Performance Evaluation of Proposed and Existing Feature Extraction Methods on Twitter 16 Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Methods** | **Accuracy** | **Precision** | **Recall** | **F-measure** |
| **CBOW+GPT** | 88.2 | 89.00 | 88.3 | 88.39 |
| **TF-IDF+GPT** | 94.09 | 94.89 | 93.05 | 94.59 |
| **one-hot+GPT** | 98.6 | 98.64 | 98.45 | 98.39 |
| **Proposed** | 99.30 | 99.56 | 99.36 | 99.39 |

The above table 5 shows that proposed method achieves outstanding results on the Twitter 16 dataset, with an accuracy of 99.30%, precision of 99.56%, recall of 99.36%, and an F-measure of 99.39. This method surpasses existing feature extraction techniques, including CBOW+GPT (88.2% accuracy), TF-IDF+GPT (94.09% accuracy), and one-hot+GPT (98.6% accuracy). The superior performance across all metrics indicates the effectiveness and robustness of the proposed method in feature extraction, leading to highly accurate tweet classification.

Table 6: Performance Evaluation of Proposed and Existing Feature Extraction Methods on PHEME Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Methods** | **Accuracy** | **Precision** | **Recall** | **F-measure** |
| **CBOW+GPT** | 91.03 | 91.49 | 91.9 | 90 |
| **TF-IDF+GPT** | 90.56 | 90.84 | 91 | 90.67 |
| **one-hot+GPT** | 96.89 | 96.68 | 96.90 | 96 |
| **Proposed** | 99.1 | 98.9 | 98.98 | 98.89 |

The above table 6 shows that proposed method demonstrates exceptional performance on the PHEME dataset, achieving an accuracy value is 99.1%, precision value is 98.9%, recall value is 98.98%, and an F-measure of 98.89. It significantly outperforms existing feature extraction techniques, including CBOW+GPT (91.03% accuracy), TF-IDF+GPT (90.56% accuracy), and one-hot+GPT (96.89% accuracy). These results underscore the proposed method's superior ability to extract relevant features, leading to highly accurate and precise tweet classification.

**4.4 Comparative analysis based on classification stage**

In the classification stage for social media rumor detection, CNN demonstrates robust performance with strong feature extraction capabilities, while LR offers simplicity and interpretability but lower accuracy. LSTM excels in handling sequential data and capturing long-term dependencies, resulting in superior classification performance.

Table 7: proposed method compared with existing classification for Twitter 15 dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Methods** | **Accuracy** | **Precision** | **Recall** | **F-measure** |
| CNN+RoBERTa | 91.3 | 92.1 | 91.54 | 91.43 |
| LR+RoBERTa | 93.56 | 93.44 | 94.01 | 94 |
| LSTM+RoBERTa | 97.6 | 97.09 | 97.1 | 97.08 |
| Proposed | 99.5 | 99.57 | 99.6 | 99.59 |

The above table 7 shows that proposed method achieves exceptional performance on the Twitter 15 dataset, with an accuracy of 99.5%, precision of 99.57%, recall of 99.6%, and an F-measure of 99.59%. It significantly outperforms existing classification techniques, including CNN+RoBERTa (91.3% accuracy), LR+RoBERTa (93.56% accuracy), and LSTM+RoBERTa (97.6% accuracy). These results demonstrate the proposed method's superior capability in accurately classifying tweets, highlighting its robustness and precision over other state-of-the-art models.

Table 8: Performance Evaluation of Proposed and Existing classifiers Methods on Twitter 16 Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Methods** | **Accuracy** | **Precision** | **Recall** | **F-measure** |
| CNN+RoBERTa | 95.3 | 95.48 | 95.45 | 95.34 |
| LR+RoBERTa | 96.89 | 96.45 | 96.76 | 96.45 |
| LSTM+RoBERTa | 94.6 | 95.67 | 94.35 | 94.56 |
| Proposed | 99.30 | 99.56 | 99.36 | 99.39 |

The above table 8 shows that proposed method achieves remarkable performance on the Twitter 16 dataset, with an accuracy value is99.30%, precision value is99.56%, recall value is99.36%, and an F-measure of 99.39%. It outperforms existing classification methods, including CNN+RoBERTa (95.3% accuracy), LR+RoBERTa (96.89% accuracy), and LSTM+RoBERTa (94.6% accuracy). These results demonstrate the proposed method's superior capability in tweet classification, showcasing its effectiveness and precision over other state-of-the-art models.

Table 9: Performance Evaluation of Proposed and Existing Classifier Methods on PHEME dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Methods** | **Accuracy** | **Precision** | **Recall** | **F-measure** |
| **CNN+RoBERTa** | 95.04 | 95.06 | 95.87 | 94 |
| **LR+RoBERTa** | 97.89 | 97.78 | 97.68 | 97.6 |
| **LSTM+RoBERTa** | 90.1 | 90.6 | 90.78 | 90.56 |
| **Proposed** | 99.1 | 98.9 | 98.98 | 98.89 |

The above table 9 shows that the proposed method achieves outstanding results on the PHEME dataset, with an accuracy value is99.1%, precision value is98.9%, recall value is98.98%, and an F-measure value is98.89%. It significantly outperforms existing classification methods, including CNN+RoBERTa (95.04% accuracy), LR+RoBERTa (97.89% accuracy), and LSTM+RoBERTa (90.1% accuracy). These results highlight the superior performance and robustness of the proposed method in accurately classifying data, demonstrating its effectiveness over other state-of-the-art models.

**4.5. Comparison with previous literature**

The proposed technique outperforms previous techniques by achieving higher accuracy and efficiency, demonstrating superior handling of complex datasets. Compared to earlier approaches, it shows significant improvements in precision and computational speed, validating its effectiveness. Additionally, it integrates advanced optimization algorithms, further enhancing performance across various benchmarks.

Table 8: Performance Comparison with Previous State-of-the-art Methods

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **References** | **Dataset** | **Methods** | **Metrics** | | | |
| **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F-measure (%)** |
| Gao et al. [15] | Twitter, Weibo, and PHEME | Self-supervised Rumor Detection | 90.5 | 95.6 | 96.7 | 96.2 |
| Ke et al. [16] | Weibo, Twitter15, and Twitter16 | graph convolutional network (GCN) | 95 | 95.4 | 94.5 | 95 |
| Xu et al. [17] | Weibo, CED | Hierarchically Aggregated Graph Neural Networks (HAGNN) | 95.7 | 97 | 91.7 | 94.3 |
| Shi et al. [18] | Weibo | machine learning | 91 | 94 | 85 | 89 |
| Proposed | Twitter15 | GPT | 99.5 | 99.57 | 99.6 | 99.59 |
| Proposed | Twitter16 | GPT | 99.30 | 99.56 | 99.36 | 99.39 |
| Proposed | PHEME | GPT | 99.1 | 98.9 | 98.98 | 98.89 |

The table 8 illustrates a performance comparison between the proposed method and previous state-of-the-art methods for rumor detection on various datasets. The previous methods include Self-supervised Rumor Detection on Twitter, Weibo, and PHEME datasets, which achieved an accuracy of 90.5%, precision of 95.6%, recall of 96.7%, and an F-measure of 96.2%. Another method using Graph Convolutional Networks (GCN) on Weibo, Twitter15, and Twitter16 datasets reported an accuracy of 95%, precision of 95.4%, recall of 94.5%, and an F-measure of 95%. Hierarchically Aggregated Graph Neural Networks (HAGNN) applied to Weibo and CED datasets attained an accuracy of 95.7%, precision of 97%, recall of 91.7%, and an F-measure of 94.3%. A machine learning approach on Weibo achieved an accuracy of 91%, precision of 94%, recall of 85%, and an F-measure of 89%.In contrast, the proposed method using GPT shows superior performance across all datasets. On the Twitter15 dataset, it achieved an accuracy value is 99.5%, precision value is 99.3%, recall value is 99.66%, and F-measure value is 99.4%. For the Twitter16 dataset, it achieved an accuracy value is 99.53%, precision value is 99.33%, recall of 99.63%, and an F-measure of 99.45%. On the PHEME dataset, it achieved an accuracy value is 99.6%, precision value is 99.36%, recall value is 99.69%, and an F-measure value is 99.39%.

**5. Conclusion and future scope**

In conclusion, this research introduces a robust method for detecting rumors on social media platforms, addressing the pressing need for effective tools to combat misinformation. By leveraging word embeddings for feature extraction and utilizing a GPT for classification, our approach successfully captures the nuanced semantic and syntactic features crucial for accurate rumor detection. The optimization of hyperparameters through the DHOA further enhances the model's performance. The experimental results demonstrate the efficacy of the proposed method, with high accuracy rates of 99.5% on the Twitter15 dataset, 99.30% on the Twitter16 dataset, and 99.1% on the PHEME dataset. These findings underscore the potential of our method in mitigating the spread of false information and preserving public trust in digital communication channels. Future research could explore the application of this method to other languages and social media platforms, as well as investigate strategies for real-time rumor detection and intervention.The method may have limited generalizability due to dataset constraints and may struggle with context-specific or nuanced rumors. Additionally, the computational resources required for training and optimization could be substantial, affecting practical implementation.Future work could focus on applying the method to diverse platforms and languages, enhancing real-time detection capabilities, and exploring more efficient model architectures. Incorporating user behavior analysis and contextual understanding could further improve detection accuracy.

**Declarations**

**Competing Interests**

On behalf of all authors, the corresponding author states that there is no conflict of interest.  
**Funding Information**

The authors declare that they have competing interests and funding

**Author contribution**

All authors read and approved the final manuscript.  
**Data Availability Statement**

Data sharing is not applicable to this article because of proprietary nature.  
**Research Involving Human and /or Animals**

Not Applicable  
**Informed Consent**

Not Applicable

**Reference**

[1] A. R. Pathak, A. Mahajan, K. Singh, A. Patil, and A. Nair, “Analysis of Techniques for Rumor Detection in Social Media,” *Procedia Comput. Sci.*, vol. 167, pp. 2286–2296, 2020, doi: 10.1016/j.procs.2020.03.281.

[2] T. Bian *et al.*, “Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 01, pp. 549–556, Apr. 2020, doi: 10.1609/aaai.v34i01.5393.

[3] Q. Li, Q. Zhang, L. Si, and Y. Liu, “Rumor Detection on Social Media: Datasets, Methods and Opportunities,” in *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, Hong Kong, China: Association for Computational Linguistics, 2019, pp. 66–75. doi: 10.18653/v1/D19-5008.

[4] X. Yang, Y. Lyu, T. Tian, Y. Liu, Y. Liu, and X. Zhang, “Rumor Detection on Social Media with Graph Structured Adversarial Learning,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Yokohama, Japan: International Joint Conferences on Artificial Intelligence Organization, Jul. 2020, pp. 1417–1423. doi: 10.24963/ijcai.2020/197.

[5] Z. He, C. Li, F. Zhou, and Y. Yang, “Rumor Detection on Social Media with Event Augmentations,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual Event Canada: ACM, Jul. 2021, pp. 2020–2024. doi: 10.1145/3404835.3463001.

[6] J. Reshi and R. Ali, *Rumor proliferation and detection in Social Media: A Review*. 2019. doi: 10.1109/ICACCS.2019.8728321.

[7] H. Zhang, Q. Fang, S. Qian, and C. Xu, “Multi-modal Knowledge-aware Event Memory Network for Social Media Rumor Detection,” in *Proceedings of the 27th ACM International Conference on Multimedia*, Nice France: ACM, Oct. 2019, pp. 1942–1951. doi: 10.1145/3343031.3350850.

[8] S. M. Alzanin and A. M. Azmi, “Detecting rumors in social media: A survey,” *Procedia Comput. Sci.*, vol. 142, pp. 294–300, 2018, doi: 10.1016/j.procs.2018.10.495.

[9] H. Bingol and B. Alatas, “Rumor Detection in Social Media Using Machine Learning Methods,” in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, Ankara, Turkey: IEEE, Nov. 2019, pp. 1–4. doi: 10.1109/UBMYK48245.2019.8965480.

[10] M. Al-Sarem, W. Boulila, M. Al-Harby, J. Qadir, and A. Alsaeedi, “Deep Learning-Based Rumor Detection on Microblogging Platforms: A Systematic Review,” *IEEE Access*, vol. 7, pp. 152788–152812, 2019, doi: 10.1109/ACCESS.2019.2947855.

[11] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang, “Call Attention to Rumors: Deep Attention Based Recurrent Neural Networks for Early Rumor Detection,” Apr. 19, 2017, *arXiv*: arXiv:1704.05973. Accessed: Jul. 19, 2024. [Online]. Available: http://arxiv.org/abs/1704.05973

[12] A. P. B. Veyseh, M. T. Thai, T. H. Nguyen, and D. Dou, “Rumor detection in social networks via deep contextual modeling,” in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Vancouver British Columbia Canada: ACM, Aug. 2019, pp. 113–120. doi: 10.1145/3341161.3342896.

[13] Y. Xu, C. Wang, Z. Dan, S. Sun, and F. Dong, “Deep Recurrent Neural Network and Data Filtering for Rumor Detection on Sina Weibo,” *Symmetry*, vol. 11, no. 11, p. 1408, Nov. 2019, doi: 10.3390/sym11111408.

[14] T. Liu *et al.*, “Rumor Detection with A Novel Graph Neural Network Approach,” *Acad. J. Sci. Technol.*, vol. 10, no. 1, pp. 305–310, Mar. 2024, doi: 10.54097/farmdr42.

[15] “Rumor detection with self-supervised learning on texts and social graph | Frontiers of Computer Science.” Accessed: Jul. 18, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s11704-022-1531-9

[16] Z. Ke, Z. Li, C. Zhou, J. Sheng, W. Silamu, and Q. Guo, “Rumor Detection on Social Media via Fused Semantic Information and a Propagation Heterogeneous Graph,” *Symmetry*, vol. 12, no. 11, p. 1806, Oct. 2020, doi: 10.3390/sym12111806.

[17] S. Xu *et al.*, “Rumor detection on social media using hierarchically aggregated feature via graph neural networks,” *Appl. Intell.*, vol. 53, no. 3, pp. 3136–3149, Feb. 2023, doi: 10.1007/s10489-022-03592-3.

[18] A. Shi, Z. Qu, Q. Jia, and C. Lyu, “Rumor Detection of COVID-19 Pandemic on Online Social Networks,” in *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, San Jose, CA, USA: IEEE, Nov. 2020, pp. 376–381. doi: 10.1109/SEC50012.2020.00055.

[19] S. Shelke and V. Attar, “Rumor detection in social network based on user, content and lexical features,” *Multimed. Tools Appl.*, vol. 81, no. 12, pp. 17347–17368, May 2022, doi: 10.1007/s11042-022-12761-y.

[20] J. Gao, S. Han, X. Song, and F. Ciravegna, “RP-DNN: A Tweet level propagation context based deep neural networks for early rumor detection in Social Media,” 2020, *arXiv*. doi: 10.48550/ARXIV.2002.12683.

[21] Y. Tian, R. Fan, X. Ding, X. Zhang, and T. Gan, “Predicting Rumor Retweeting Behavior of Social Media Users in Public Emergencies,” *IEEE Access*, vol. 8, pp. 87121–87132, 2020, doi: 10.1109/ACCESS.2020.2989180.

[22] S. A. Alkhodair, S. H. H. Ding, B. C. M. Fung, and J. Liu, “Detecting breaking news rumors of emerging topics in social media,” *Inf. Process. Manag.*, vol. 57, no. 2, p. 102018, Mar. 2020, doi: 10.1016/j.ipm.2019.02.016.

[23] “Data Pre-processing of Website Browsing Records: To Prepare Quality Dataset for Web Page Classification | Apandi | JOIV : International Journal on Informatics Visualization.” Accessed: May 29, 2024. [Online]. Available: https://joiv.org/index.php/joiv/article/view/1618

[24] “Detecting spam e-mails using stop word TF-IDF and stemming algorithm with Naïve Bayes classifier on the multicore GPU. | International Journal of Electrical &amp; Computer Engineering (2088-8708) | EBSCOhost.” Accessed: May 29, 2024. [Online]. Available: https://openurl.ebsco.com/EPDB%3Agcd%3A16%3A19222752/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A149982498&crl=c

[25] L. Gao, L. Zhang, L. Zhang, and J. Huang, “RSVN: A RoBERTa Sentence Vector Normalization Scheme for Short Texts to Extract Semantic Information,” *Appl. Sci.*, vol. 12, no. 21, p. 11278, Nov. 2022, doi: 10.3390/app122111278.

[26] G. Brammya, S. Praveena, N. S. Ninu Preetha, R. Ramya, B. R. Rajakumar, and D. Binu, “Deer Hunting Optimization Algorithm: A New Nature-Inspired Meta-heuristic Paradigm,” *Comput. J.*, p. bxy133, May 2019, doi: 10.1093/comjnl/bxy133.