International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I9P230
Volume 71 Issue 9, 341-355, September 2023

Original Article

Equanimous Intelligent Water Drop Algorithm-Based Feed-Forward Neural Network (EIWDA-FFNN) for Improving Sentiment Analysis

P. Radha[1], N. Sudha Bhuvaneswari[2]

[1]*Department of Computer Science, Sri Krishna Arts and Science College, Tamil Nadu, India.*
[2]*Department of Computer Science, Dr. G.R Damodaran College of Science, Tamil Nadu, India.*

[1]*Corresponding Author : radhaharismita@gmail.com*

Received: 13 May 2023 Revised: 29 July 2023 Accepted: 15 August 2023 Published: 03 September 2023

Abstract - *Sentiment analysis is critical in natural language processing, particularly online shopping. With the rise of e-commerce platforms and social media, customers generate enormous amounts of data through their reviews and feedback. The sentiment analysis of this data provides valuable insights into customer opinions and preferences, enabling companies to improve their products and services. However, sentiment analysis faces several challenges, including the ambiguity of the language used in customer reviews and the lack of labelled data. Researchers have developed various techniques to overcome these challenges, including machine learning and deep learning models. This research paper presents a new approach to sentiment analysis in online shopping reviews, using the Equanimous Intelligent Water Drop Algorithm-based Feed-Forward Neural Network (EIWD-FFNN). The proposed algorithm combines the Equanimous Intelligent Water Drop (EIWD) algorithm with the Feed-Forward Neural Network to optimize the network's hyperparameters for accurate sentiment classification. The EIWD algorithm maximizes the number of hidden layers, the learning rate, and the number of neurons in each hidden layer of the FFNN. This optimization approach reduces the dependency on labelled data and enhances the performance of the sentiment analysis task. The experiments are conducted on a dataset of customer reviews from Amazon. The results demonstrate that the proposed EIWD-FFNN algorithm outperforms other state-of-the-art sentiment analysis techniques, achieving an accuracy of 90.17%. Our study highlights the importance of developing accurate sentiment analysis techniques to understand customer feedback and improve products and services. The proposed algorithm offers a new approach that can overcome the challenges of ambiguous language and lack of labelled data, providing valuable insights into customer opinions. Overall, the proposed EIWD-FFNN algorithm offers a promising approach for sentiment analysis and could be a useful tool for businesses to improve their understanding of customer feedback in online shopping platforms.*

Keywords - *Sentiment analysis, Classification, Amazon, Neural network, Intelligent Water Drop, Feed-Forward.*

1. Introduction

Sentiment analysis, also known as opinion mining, is the process of extracting emotions, attitudes, and opinions expressed in text data. Sentiment analysis in opinion mining provides a computational approach to understanding the sentiments expressed in a large corpus of text data [1]. This allows organizations to gain valuable insights into consumer opinions and preferences, which can be used to make informed business decisions.

Sentiment analysis is a crucial tool for businesses as it allows them to measure the success of their products and services. By analyzing the sentiment expressed in customer reviews, companies can determine whether their products and services meet their customers' needs and expectations. This information can be used to improve existing products and services or to develop new products and services that better meet the needs and expectations of customers [2].

Another critical application of sentiment analysis in opinion mining is in the field of marketing. Companies use sentiment analysis to understand the attitudes and opinions of customers towards their products and services and their competitors. This information can be used to create targeted marketing campaigns that better resonate with customers and improve brand image [3]. Additionally, sentiment analysis can be used to monitor the effectiveness of marketing campaigns, allowing companies to make real-time adjustments to optimize their campaigns.

In addition to its applications in business and marketing, sentiment analysis is also used in politics and media. Political organizations use sentiment analysis to gauge public opinion on important issues, such as political candidates or policies. This information can be used to inform political strategy and decision-making. Similarly, media organizations use sentiment analysis to understand the public's opinions on current events and monitor news articles' tone and sentiment [4].

The accuracy of sentiment analysis is of utmost importance, as incorrect sentiment analysis results can lead to wrong business decisions and harm a company's reputation. The accuracy of sentiment analysis depends on various factors, including the quality of the data being analyzed, the algorithms used to process the data, and the skill of the sentiment analyst. To ensure the accuracy of sentiment analysis results, it is essential to use high-quality data sources, such as customer reviews and social media posts, and sophisticated algorithms that consider the context of the text being analyzed [5].

Sentiment analysis is a powerful tool in opinion mining, providing organizations valuable insights into consumer opinions and preferences. Its business, marketing, politics, and media applications demonstrate its versatility and importance in today's data-driven world. However, the accuracy of sentiment analysis results is crucial, and organizations must take steps to ensure that the data and algorithms used are of the highest quality. As sentiment analysis evolves and improves, it will likely play an even more critical role in shaping business and social decision-making [6].

The significant feature of sentiment analysis is [3, 7, 8]:

- Sentiment analysis of customer surveys: The ability to analyze customer survey data and understand customer opinions and preferences.

- Sentiment analysis of customer complaints: The ability to analyze customer complaints data and understand customer opinions and preferences.

- Sentiment analysis of customer churn data: The ability to analyze customer churn data and understand customer sentiment towards the company.

- Sentiment analysis of customer lifetime value: The ability to analyze customer lifetime value data and understand customer sentiment towards the company.

- Sentiment analysis of customer loyalty: The ability to analyze customer loyalty data and understand customer sentiment towards the company.

1.1. Problem Statement

Irony and sarcasm often pose a significant challenge for sentiment analysis algorithms as they can completely change the intended meaning of a sentence. Enhancing irony and sarcasm detection in sentiment analysis involves developing algorithms that can accurately identify and categorize the sentiment expressed in texts that contain irony or sarcasm. This requires algorithms to consider the contextual cues and linguistic patterns commonly associated with irony and sarcasm and the ability to distinguish between literal and figurative expressions. Improving the detection of irony and sarcasm is essential in applications such as sentiment analysis of social media posts, product reviews, and customer feedback, where accurate identification of the sentiment expressed can significantly impact decision-making processes. The objective is to enhance the ability of sentiment analysis algorithms to classify sentiments accurately, even in the presence of irony or sarcasm, improving their overall performance and effectiveness.

1.2. Objective

Developing Multilingual Models: To overcome the challenge of sentiment analysis in different languages, developing multilingual models is an essential objective for this field of research. To achieve this objective, researchers may focus on developing models capable of sentiment analysis in multiple languages and exploring transfer learning techniques to adapt existing models to new languages. Additionally, researchers may investigate using cross-lingual word embeddings and other methods to better incorporate language-specific information into the model.

2. Literature Review

"Heterogeneous Graph Convolution" [9] presents a novel approach to multimodal SA. The authors propose using graph convolutional networks on heterogeneous graphs to capture the relationships between different modalities and to use self-supervision to improve the model's performance. This approach can effectively model the complex relationships between text, image, and audio modalities, improving SA performance. "Multimodal Sentiment Analysis" [10] provides a comprehensive overview of multimodal SA. The authors survey the field's history, discuss various multimodal fusion methods, and provide an overview of the challenges and future directions for research. This article offers a valuable resource for researchers and practitioners in sentiment analysis. It allows for a comprehensive overview of the field and highlights the current state-of-the-art. "Hybrid Sentiment Analysis" [11] explores using textual and interactive information for SA. The authors proposed a novel hybrid model that combines the strengths of both types of information to improve the performance of SA, can capture the exact sentiment expressed in text and the implicit sentiment conveyed through user interactions, can improve SA performance, especially in great interactive information domains.

Table 1. Comparison of related literature

Related Literature	Merits	Demerits
Heterogeneous Graph Convolution	• It uses heterogeneous graph convolution, which enhances the in-domain self-supervision capability. • It has improved accuracy and robustness.	• Computational complexity affects scalability. • The method is evaluated on a limited number of datasets.
Multimodal Sentiment Analysis	• The multimodal fusion method helps identify the potential way of increasing classification accuracy.	• The review focuses primarily on state of the art in multimodal sentiment analysis.
Hybrid Sentiment Analysis	• It combines textual and interactive information for improved sentiment analysis. • It leverages the strengths of both types of information to provide improved sentiment analysis.	• It is not suitable for all sentiment analysis tasks.
Transfer-based Adaptive Tree	• Transfer-based adaptive tree for multimodal sentiment analysis that leverages latent user aspects. • It can adapt to new domains and improve sentiment analysis performance.	• It increased false positives and false negatives. • Performance is evaluated only on limited datasets.
Hybrid Deep Learning Model	• Effectively capture the aspect-level sentiments. • Improved sentiment analysis results.	• It is unsuitable for all sentiment analysis tasks, and its performance is evaluated only on specific datasets.
Hybrid Feature Extraction	• Can effectively extract features from various sources. • The hybrid approach combines the strengths of multiple feature extraction methods.	• The complex model design may require specialized knowledge and expertise to implement and train. • High computational resources required
Enhancement of the KNN Model	• Uses WordNet semantic relations to enhance the KNN model • Resulting in improved implicit aspect identification • The KNN model is simple and easy to implement	• It may not perform as well as more advanced deep-learning models • Limited to the semantic relations present in WordNet, it may not capture all relevant implicit aspects.
Multi CNN-Based Deep Learning Model	• Bi-GRU and CNN components capture different aspects of the data. • It can handle large amounts of data, making it suitable for social media sentiment analysis.	• The complex model design may require specialized knowledge and expertise to implement and train. • High computational resources are required.
Emotional Sentiment Analysis	• Can provide valuable insights into the emotional states of individuals on social media • Can help identify early warning signs for mental health issues	• This may be subject to privacy concerns and ethical issues surrounding collecting and analysing personal data. • It may not accurately capture all emotions, leading to false positive or false negative results.
Ensemble Transformer-Based Model	• Can handle the complexities of the Arabic language, including multiple scripts and dialects • The ensemble approach combines the strengths of multiple models, leading to improved accuracy.	• Implementing and training the complex model design may require specialized knowledge and expertise. • High computational resources are required.
Heterogeneous Graph Convolution	• Can handle large-scale sentiment analysis. • Aids in group decision-making through sentiment analysis.	• The accuracy of the sentiment analysis may be limited by the quality and amount of data used. • It may not generalize well to other domains or products.

Multimodal Sentiment Analysis	• Incorporates multiple modalities in sentiment analysis. • Utilizes information relevance to improve performance.	• The complexity of the model may limit its scalability and generalizability. • The model may not perform well on tasks with limited data.

"Transfer-based Adaptive Tree" [12] proposes a transfer-based adaptive tree approach to SA incorporating latent user aspects. This approach leverages pre-trained SA models to perform SA on new domains while also adapting to the specific characteristics of the target domain. This approach can effectively transfer knowledge from pre-trained models to new fields while also adapting to the particular characteristics of the target domain, leading to improved performance. "Hybrid Deep Learning Model" [13] presents a novel hybrid deep learning approach to aspect-based SA. The authors propose a hybrid model that combines convolutional neural networks and recurrent neural networks to capture the text's local and global context. This approach can effectively model the relationships between aspect terms and sentiment, improving performance on aspect-based SA tasks [14]. "Hybrid Feature Extraction" [15] presents a novel deep learning model for SA of consumer feedback. The model uses a hybrid feature extraction approach combining word embeddings and hand-crafted features such as part-of-speech tags [16]. The experiments show that the proposed model outperforms traditional machine learning approaches and deep learning models using only word embeddings. The results demonstrate the effectiveness of the hybrid feature extraction approach in capturing complex sentiment information in consumer feedback.

"Enhancement of KNN Model" [17] proposes an enhancement of the K-Nearest Neighbors (KNN) model for implicit aspect identification in SA by incorporating semantic relationships from WordNet. The experiments show that the proposed approach outperforms the traditional KNN and other state-of-the-art models for implicit aspect identification. The results suggest that integrating semantic relationships from WordNet can effectively capture implicit sentiment information in the text. "Multi CNN-Based Deep Learning Model" [18] proposed for SA of social media content. The model combines the strengths of bi-directional GRU and multi-channel CNN to effectively capture the sequential and semantic information in a social media text. The experiments show that the proposed model outperforms state-of-the-art models for SA of social media content. The results demonstrate the effectiveness of the MBi-GRUMCONV model in capturing both sequential and semantic information for SA. "Emotional Sentiment Analysis" [19] focuses on the importance of emotional SA for mental health safety in social media. The authors discuss the challenges in detecting and analyzing emotions in social media text and propose using deep learning models to address these challenges. The experiments show that the proposed deep learning models can effectively detect and analyze emotions in a social media text, providing valuable insights into users' mental health. The results highlight the importance of emotional SA for mental health safety in social media. "Ensemble Transformer-Based Model" [20] presents an ensemble transformer-based model for SA of Arabic text. The authors propose combining multiple transformer models with different architectures to achieve improved performance compared to individual models. The experiments show that the ensemble model outperforms state-of-the-art models for Arabic SA. The results demonstrate the effectiveness of the ensemble approach in achieving improved performance for SA of Arabic text. Optimization plays a significant role in Different research domains [21-35], [36].

"Large Scale Group Decision-Making System (LSGDMS)" [37] proposes a new approach to group decision-making that considers the sentiment individuals express. The authors claim that based on SA clustering, their system can effectively capture the sentiment expressed by individuals in a group and use it to make accurate group decisions. "Joint Multimodal Sentiment Analysis (JMSA)" [38] proposes a new approach to multimodal SA that considers the relevance of the information contained in different modalities. The authors claim that their system, based on information relevance, can effectively capture the sentiment expressed in multimodal data, as it considers the significance of the information contained in each modality.

3. Equanimous Intelligent Water Drop Algorithm-Based Feed-Forward Neural Network

A feed-forward neural network employs a forward pass algorithm to generate output from an input. These are the stages that make up the algorithm:

Step 1: Setup the Network
Network parameters such as layer depth, the total number of neurons, and activation function are set during the initialization phase. Every neuron's starting weight and bias are entirely arbitrary.

Step 2: Compute the first hidden layer's output.
The data is sent into the first hidden layer. To generate an output, the neurons within the initial hidden layer first calculate a weight value of the input, then add a correction factor, and then apply the activation function. Every neuron's output becomes part of the input for the following layer.

Step 3: Compute the results of the final hidden layers.
Similarly, the output of one hidden layer becomes the input for the next hidden layer's neurons, and so on.

Step 4: Computation of the output layer's output
At the output layer, which receives the network's final output from the last hidden layer, each neuron calculates a weight value of the input, inserts a bias term, and then uses the activation function.

Step 5: Determine the loss function.
To quantify how much of a discrepancy there is between the anticipated and actual result for a particular input, the lost function is employed. Mean squared error is the most used loss function (MSE).

Step 6: Update the biases and weights.
By calculating the loss function gradient about the weights and biases, the backpropagation method updates the network's neuronal weights and biases. The loss function is minimized by adjusting the weights and preferences as reversely as the gradient.

Step 7: Repeat steps 2 through 6 for every input in the practice data.
For each input in the training data, the forward pass and backpropagation are performed until the network converges; at this point, the loss function has achieved a minimum.

3.1. Activation Mechanisms

The activation function is critical in a feed-forward neural network since it defines how a neuron responds to its input. In practice, these activation functions are the most popular:

3.1.1. Signature Function

A non-linear function, the sigmoid function converts any real number to an integer range between 0 and 1, and it is defined in Eq.(1).

$$f(x) = 1/\left(1 + e^{(-x)}\right) \tag{1}$$

With x serving as the neuron's input.

3.1.2. ReLUFunction

Non-linear functions such as the Rectified Linear Unit (ReLU) function return the inputs if it is affirmative and 0 otherwise. Eq.(2) characterizes the same.

$$f(x) = max(0, x) \tag{2}$$

The input to the neuron is x.

3.1.3. SoftmaxFunction

Feed-forward neural networks frequently utilize the non-linear softmax function in the output layer to calculate the probabilistic model of the output classes, which is expressed in Eq.(3).

$$f(x_i) = e^{(x_i)}/sum\ je^{(x_j)} \tag{3}$$

Here ith is the output layer neuron that received input from x_i.

3.2. Functions of Loss

The loss function is a metric that quantifies the deviation between the predicted and observed outputs for a given input. Popular loss functions include (i) Mean Squared Error and (ii) Loss of Cross-Entropy.

3.2.1. Mean Squared Error (MSE)

A popular loss function for regression issues is the mean squared error; Eq. (4) provides the same.

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

3.2.2. Loss of Cross-Entropy

In the context of classification issues, the cross-entropy loss is a frequently employed loss function, and Eq.(5) characterizes the same.

$$L = \frac{1}{n}\sum_{i} log(y' - i) + (1 - y_i)log(1 - y_i') \tag{5}$$

For some input ith, the actual label is y_i, the projected label is y', and the input is n.

3.3. Backpropagation

A particular procedure called backpropagation keeps a feed-forward neural network's weights and biases current. A gradient of the error function about the preferences and weights is computed using the following equation in calculus. The steps of the backpropagation algorithm are as follows.

Step 1: Calculate the output layer error.
The output layer error is determined by comparing the projected and actual outputs.

$$\delta = y - y' \tag{6}$$

In which the actual result is y, and the expected result is y'.

Step 2: Compute the error in the hidden layers.
The error in the hidden layers is calculated by propagating the error in the output layer backwards through the network. The error at each neuron is the weighted sum of the errors in the neurons it connects to in the next layer.

$$\delta = f'(z) * w * \delta \tag{7}$$

Wherein $f(z)$ is the activation function's derivative, w is the connection's weight, and delta output is the output layer's error.

Step 3: Revisit the biases and weights.
Following are some equations used to update the neuronal weights and biases:

$$\begin{gathered} w_{new} = w_{old} - learning_{rate} * \delta * x \\ b_{new} = b_{old} - learning_{rate} * \delta \end{gathered} \tag{8}$$

Wherein x is the input to the neuron, delta is the neuron's error, $learning_{rate}$ is the rate at which the neuron is learning, and w_{old} and b_{old} are the previous weights and biases.

Step 4: Repeat steps 1-3 for each input in the training data
The backpropagation algorithm is repeated for each information in the training data until the network reaches convergence.

3.4. Training a Feed-forward Neural Network

To train a feed-forward neural network, the following steps are typically followed:

Step 1: Prepare the data
Preprocessing involves separating incoming data into training, validation, and testing.

Step 2: Initialize the network
The network is initialized with a random set of weights and biases.

Step 3: Forward pass
The input data is fed through the network to generate a predicted output.

Step 4: Determine the loss.
A loss function may be determined by comparing the expected and actual output.

Step 5: Backpropagation
The network's neuronal weights and biases are adjusted using the backpropagation method.

Step 6: Repeat steps 3-5 for multiple epochs.
After the network has converged, a forward passage and backpropagation are stopped.

Step 7: Evaluate the performance on the test set.
The accuracy and generalization abilities of the neural model are assessed on a test set.

3.5. Regularization Techniques

Regularization techniques are used to prevent overfitting in a feed-forward neural network. Some commonly used regularization techniques are:

3.5.1. L1 Technique

After applying L1 regularisation, the loss function will now include a penalty term that scales inversely with the weights' absolute values.

$$L = loss + \lambda \sum_i |w_s| \tag{9}$$

Where lambda is the regularization strength.

3.5.2. L2 Technique

After applying L2 regularisation, the loss function will now include a penalty term that scales inversely with the square of the weights.

$$L = loss + \lambda \sum_i w_i^2 \tag{10}$$

Where lambda is the regularization strength.

3.5.3. Dropout

With the dropout method, some neurons are randomly removed from the training process. The network is effectively forced to learn redundant representations, which helps to reduce overfitting.

3.6. Hyperparameters

Before a neural network is trained, its performance is heavily influenced by its hyperparameters, which are specified parameters. Examples of frequently used hyperparameters include (i) Learning rate, (ii) Total Number of hidden layers, (iii) Number of neurons per layer, and (iv) Activation function.

3.6.1. Learning Rate

The learning rate determines how much the weights and biases are updated during backpropagation. A high learning rate can cause the consequences to oscillate and prevent convergence, while a low learning rate can result in a slow conjunction.

3.6.2. Total Number of Hidden Layers

Complexity and the network's capacity to model intricate interactions are both functions of the hidden layers. However, adding too many hidden layers can result in overfitting and slow training.

3.6.3. Number of Neurons Per Layer

How well a network can represent complex functions depends on how many neurons are in each layer. However, adding too many neurons can also result in overfitting.

3.6.4. Activation Function

The network's efficiency may change drastically depending on the activation function selected. Standard activation functions include sigmoid, tanh, and ReLU.

3.7. Classification

Feed-forward neural networks (FFNN) can be used for classification tasks by mapping input data to output classes through a series of interconnected layers of neurons. The classification process in FFNN involves three key steps: forward propagation, computation of error, and backward propagation of error.

3.7.1 Forward Propagation

Data is sent into the network at the input layer, and then each layer's output is produced to use a non-linear activation function as part of the forward propagation process. The kth layer's production, represented by the notation h_k, may be calculated mathematically as:

$$h_k = f(W_k h_{k-1} + b_k) \tag{11}$$

If w_k and b_k are the kth layer's weights and biases, the output of the $\{k-1\}$ layer is h_{k-1}, and the activation function is f.

The activation function makes the output of each layer non-linear, which lets the network pick up subtleties in the data. The sigmoid, ReLU, and tanh activation functions are only a few of the most popular ones. Predicted class probabilities are produced as h_n from the last layer, the class with the most significant probability is chosen as the predicted class.

3.7.2. Computation of Error

A loss function calculates the network's error by comparing the anticipated class probabilities to the correct class labels. The cross-entropy loss is the most popular loss function for classification tasks since it quantifies the dissimilarity between the predicted and observed class probabilities. The formula for calculating the cross-entropy loss for a given training example is:

$$L(y,\hat{y}) = -\sum\nolimits_{-} \{i=1\}\Lambda\{C\} y_i log\{\hat{y}_i\} \tag{12}$$

Where y is the accurate class label, $\Lambda\{Y\}$ is the predicted class probability vector, C is the number of classes, and a log is the natural logarithm.

The cross-entropy loss penalizes the network more when it makes confident but incorrect predictions and less when it makes uncertain or correct predictions.

3.7.3. Backward Propagation of Error

The mistake is then re-routed back through the network to adjust the weights and biases, a process known as backward propagation. Weights and biases are updated based on the loss gradient relative to each layer's weights and biases, calculated using the chain rule of differentiation. Precisely, the kth layer's weight and bias update may be calculated as:

$$W_k = W_k - \alpha(\partial L/\partial W_k)$$
$$b_k = b_k - \alpha * \frac{L}{b_k} \tag{13}$$

Where alpha determines how much the weights and biases are changed.

At each iteration, the backpropagation method fine-tunes the network's weights and biases to reduce the loss of the training data. This procedure is repeated until convergence or the maximum number of iterations is reached.

Feed-forward neural networks perform classification by mapping input data to output classes through a series of interconnected layers of neurons using non-linear activation functions. A loss function compares the predicted class probabilities to the actual class labels. The error is sent back into the network to adjust the weights and biases using the gradient descent method.

3.8. Equanimous Intelligent Water Drop Algorithm

IWD is inspired by how natural rivers work and the relationships between individual water droplets. It mimics the biological interactions between river water and soil. Large-scale swarm movement may be seen in nature, specifically in the water droplets of rivers in motion. Indeed, a river's courses are built by the persistent cooperation of a swarm of water droplets and the surrounding environment. The river bed, over which the water trickles, is a significant feature of the landscape that has far-reaching implications for the routes. In reality, a natural river's course is determined by water droplets in a swarm competing with various features of its surrounding environment.

Three notable changes may occur to a water drop as it travels from the upper point of a river to the succeeding places downstream:

- The water drop's soil will increase.
- The water drop's velocity will rise.
- The soil of the river's bed will decrease.

Some of the river bed's dirt gets washed away and incorporated into the water droplets as they travel through the river. In addition, the high-velocity water drops travel away more dirt from the riverbed than the low-velocity water drops. Consequently, the rate at which water descends is proportional to the amount of removed dirt. Also, a path with little soil will enhance a water drop's velocity more than a rich soil path. Therefore, a route (or path) with less dirt allows the water drop to accumulate more soil, increasing its speed, while a path with more soil opposes the drop more vigorously, allowing it to collect less soil and decrease its speed. Water drops tend to pick the one with less dirt while deciding between many forks in the river on its way from upstream to downstream.

Proposed Algorithm

Input:

- Training data X and their corresponding labels Y
- Network architecture (count of neurons in each layer)
- Its alpha rate of learning
- Number of epochs

Output:

- Network biases and trained weights

Procedure:

Initialization:

- Set the network's weights and biases at random.
- Choose a training rate as well as the number of iterations.

Training:

For epoch in 1 to epochs:

For each training example x and its corresponding label y:

Forward Propagation:

Compute the output of each layer using the following equations:

$h_1 = f(w_1 * x + b_1)$

$h_2 = f(w_2 * h_1 + b_2)$

...

$h_n = softmax(W_n * h_{n-1} + b_n)$

Where f is the activation function (such as ReLU or sigmoid), and softmax is the output activation function that converts the outputs of the last layer into a probability distribution over the classes.

Compute the loss $L(y, h_n)$ Using the information loss due to cross-entropy:

$L(y, h_n) = -sum\ (y_i * log h_{n,i})$

Backward Propagation:

Compute the error of the last layer as:

$\delta_n = h_{n-y}$

For each layer k in reverse order (from $n - 1$ to 1):

Compute the error delta_k using the following equation:

$\delta_k = f'(z_k * (W_{k+1})^T * \delta_{k+1})$

where f' is the derivative of the activation function, and $z_k = W_k * h_{k-1} + b_k$ is the input to the activation function.

Compute the gradients of the weights and biases using the following equations:

$\delta W_k = h_{k-1} * \delta_k^T$

$$\delta b_k = \sum_{s=1}^{N} \delta_{k_i}$$

Update the weights and biases using the gradients and the learning rate:

$W_k = W_k - \alpha \delta W_k$

$b_k = b_k - \alpha * \delta b_k$

Prediction:

- For each test example p:
- Feed the input p into the network and compute the output of the last layer.
- Select the class with the highest probability as the predicted class.

Evaluation:

- Figure out the network's performance on the test set.

EIWD uses two crucial features, which are (i) soil present in water droplets and (ii) speed it travels are two distinct factors. It is possible that the values of soil and velocity will dynamically alter the surroundings while it travels. From point s to point w, the velocity increases by a velocity value. Its growth is often proportional to the $soil(s,w)$, the reverse of the soil at points s and w. Eq.(14) can give a mathematical expression for the connection between velocity and $soil(s,w)$.

$$\Delta velocity(IWD) \propto \frac{1}{soil(s,w)} \tag{14}$$

One possible formulation between Δvelocity and $soil(s,w)$ is where the velocity of IWD gets updated by the $soil(s,w)$ as a non-linear function, expressed in Eq.(15).

$$\Delta vel^{IWD}(f) = \frac{d_r}{v_r + u_r \times soil^{2\delta}(s,w)} \tag{15}$$

Wherein δ, d_r, v_r and u_r does the user choose parameters based on demand. It is also expected that the amount of soil added to the IWD denoted as $\Delta soil(s,w)$, is inversely proportional to the time spent travelling from point s to point w. Eq.(16) offers a viable alternative to Δsoil(IWD).

$$\Delta soil(s,w) = \frac{d_e}{b_e + u_e \times time^{2\rho}(s,w{:}\,vel^{IWD})} \tag{16}$$

Where $time^{2\rho}(s,w{:}\,vel^{IWD})$ is the amount of time it takes to keep moving from position s to location w at a speed of vel^{IWD} and, ρ, d_e, v_e and u_e does it choose any real integer values based on the demand.

In Eq.(17) $time(s,w;vel^{IWD})$ is determined as a function of velocity using standard mechanics for linear motion.

$$time(s,w;vel^{IWD}) = \frac{1}{vel^{IWD}} \tag{17}$$

In Eq.(17), $soil(s,w)$ should also be updated as some soil is removed from the traversed route as described in Eq.(18).

$$soil(s,w) = \varphi_k \times soil(s,w) - \varphi_t \times \Delta soil(s,w) \tag{18}$$

Parameters for updating the local soil are φ_k and φ_t, both of which must be positive. Thus, the following formula is used to get the soil volume, expressed in Eq.(19).

$$soil^{IWD} = soil^{IWD} + \Delta soil(s,w) \tag{19}$$

The significant characteristic of IWD is that it favours routes with less dirt on the way to their final destination. Using this property as inspiration, this research uses a probability function for picking position w from a random value when the water drop is placed at location s, and Eq.(20) expresses the same.

$$m(s,w;IWD) = \frac{g\big(soil(s,w)\big)}{\sum_{a \notin ru(IWD)} g\big(soil(s,a)\big)} \tag{20}$$

Where $ru(IWD)$ stands for the problem's infeasible nodes due to environmental limitations, and Eq.(21) expresses how $g\big(soil(s,w)\big)$ is computed.

$$g\big(soil(s,w)\big) = \frac{1}{\sigma_e + j\big(soil(s,w)\big)} \tag{21}$$

Wherein σ_e is a constant positive integer used to avoid the issue that arises during the calculation in Eq.(21) (i.e., division by zero) and $j\big(soil(s,w)\big)$ is a function that shifts the $soil(s,w)$ towards positive integer values. Eq.(22) expresses the same.

$$j\big(soil(s,w)\big) = \begin{cases} soil(s,w) & if\, min_{z \notin ru(IWD)}\big(soil(s,z)\big) \geq 0 \\ soil(s,w) - min_{z \notin ru(IWD)}\big(soil(s,z)\big) & Otherwise \end{cases} \tag{22}$$

3.8.1. Scheduler Rejection

Each IWD's visited node list is generated at the beginning of each iteration and is initially an empty value. Each IWD's initial velocity is set to InitVel, and the initial dirt value is set to 0. Starting with a node at random, an IWD is placed to choose neighbouring nodes to build the solution.

During this process, IWD verifies the validity of all environmental restrictions, and it would assist in solving the problem. While it completes each iteration, it builds its solutions parallelly. These are the major features of the proposed IWD:

Stage 1. Initialization

An encoding strategy is a mandatory methodology in initialization for giving a solution in an optimum manner. In classical scheduling issues, a permutation list is used, and it is a standard encoding approach. A permutation list represents every drop of IWD. Random keys (RK) are a relatively better encoding method for IWD to achieve a better result. It is commonly used to describe the recommended sequence of tasks. By relaxing the restrictions of the RK technique, each drop can be an actual number in the range $(0, c+1)$. The water drop should be denied entry to the overall water if it exceeds the value 1. In an IWD graph, a node is an actual number, and an edge is a schedule that repeats itself in two different orders. Moreover, this research assumes that the first population is generated randomly, which is expected to be a popular method for initializing the solutions.

Stage 2. Initialization of the following Static Parameters

- Criteria for termination ($MaxIter$)
- The total amount of water droplets ($EIWD$)
- First soil ($InitSoil$)
- Initial speed ($InitVel$)
- Updated velocity parameters (δ, d_r, v_r and u_r)
- Parameters for updating the soil (ρ, d_e, v_e and u_e)

It is up to the general research issue identified to determine the optimal maximum number of iterations. Here, this research work utilizes a value of 500 for $MaxIter$. The $EIWD$ population water-drop target is set to 500. An individual water drop with $EIWD$ indicates a potential answer to the issue. The parameters for updating the velocity are determined to be $\delta = 1, d_r = 1, v_r = 0.01$ and $u_r = 1$, where all these are based on the findings of a main experiment. The settings for the soil update are $\rho = 1, d_e = 1, v_e = 0.01$ and $u_e = 1$. Within the range [0, 1], this research selects the value of φ_t for the local soil update parameter and φ_{IWD} for the soil surface updating parameter. The optimal values for both the local soil update variable φ_t and the overall soil update variable φ_{IWD} are 0.53.

Stage 3. Beginning Values

- EIWD's speed of flow
- Ground of EIWD
- List of nodes EWID visited

Each route will have 50 initial soil ($InitSoil$), and it is applied to it. When $InitVel$ = 4, the speed of every EIWD is set to 4.5. A solution for each $EIWD$ is built using a visited node list, $R_u(IWD)$, which is initially empty.

Stage 4**.** Random initial hops are necessary for IWDs to make and put them in an optimum place.

Stage 5**.** Each $EIWD$'s $R_u(EIWD)$ of visited nodes should be refreshed to include the most recently visited nodes.

Stage 6. Steps 6.1 and 6.2 are repeated until the EIWD achieves the target solution.

Stage 6.1. When an $EIWD$ is moved to a position $R_u(EIWD)$, this research work performs calculations to determine the likelihood that node w_{best} will be selected.

Stage 6.2**.** Eq.(23) is used to calculate the velocity of each $EIWD$ as it travels from node s to node w_{best}.

$$vel^{IWD}(f+1) = vel^{IWD}(f) + \frac{d_r}{v_r + u_r \times soil^{2\delta}(s,w)} \quad (23)$$

Where the new velocity is denoted by $vel^{IWD}(f+1)$.

$$soil(s,w) = \varphi_k \times soil(s,w) - \varphi_t \times \Delta soil(s,w) \quad (24)$$

and

$$soil^{IWD} = soil^{IWD} + \Delta soil(s,w) \quad (25)$$

When $\varphi_k = 1 - \varphi_t$, then $\Delta soil(s,w)$ is calculated using Eq.(26).

$$\Delta soil(s,w) = \frac{d_e}{v_e + u_e \times time^2\big(s,w{:}\,vel^{IWD}(f+1)\big)} \quad (26)$$

$time^2\big(s,w{:}\,vel^{IWD}(f+1)\big)$ calculated using Eq.(27) during which

$$time\big(s,w{:}\,vel^{IWD}(f+1)\big) = \frac{1}{vel^{IWD}(f+1)} \quad (27)$$

*Stage 7***.** *Lexicographic Utility Function*

This step sets solutions F^{IWD} set by $EIWD$s in the threshold state and rate their overall quality. To account for the weights placed on the two objectives, the Lexicographic technique generates two models, denoted by $M1$ and $M2$, respectively.

$$M1{:}\, MinFU = \sum_{s=1}^{t} (1 - Q_s)_{\vartheta_s} \quad (28)$$

Subject to $\sum_{s=1}^{t} \theta_s \le T_F^*$

$$M2{:}\, Min\ T_F = \sum_{s=1}^{t} \theta_s \quad (29)$$

Subject to $\sum_{s=1}^{t} (1 - Q_s)_{\vartheta_s} \le FU^*$

Stage 8**.** Eq.(30) finds the iteration's best answer (i.e., solutions) F^{SV} out of all the solutions F^{IWD}.

$$F^{SV} = \arg min_{allIWDS}\, q(F^{IWD}) \quad (30)$$

Stage 9**.** By replacing F^{TB} with F^{IB}, the best solution from the current iteration can be obtained, and Eq.(31) expresses the same.

$$F^{TB} = \begin{cases} F^{IB} & if\ x(F^{IB}) \le x(F^{TB}) \\ F^{TB} & else \end{cases} \quad (31)$$

Step 10**.** Adjust the border soil around the F^{IB}, i.e., the iteration that reflects the solution's improved quality where Eq.(32) expresses the same.

$$soil(s,w) = (1 + \varphi IWD).soil(s,w) - \varphi IWD.\frac{1}{(T_{IB})}.soil_{IB}^{IWD} \text{ for } (s,w)\omega\ F^{IB} \quad (32)$$

Where T_{IB} is the total number of water drops and $soil_{IB}^{IWD}$ is the soil of the best iteration so far.

Stage 11**.** If $Iter$ is less than $MaxIter$, set $Iter = Iter + 1$, and go to Step 4 and follow still Step 11.

Stage 12**.** Stop the EIWD and give the best answer of F^{TB}.

3.9. Iterated Local Search (ILS)

A random search paradigm called iteration-based threshold local search (ITLS) was used to improve the efficiency of EIWD. It will provide simplicity, efficacy, and efficiency in tackling combinatorial optimization issues. ITLS is a straightforward stochastic local search strategy due to its basic tenets. ITLS relies heavily on (i) geotargeting and (ii) perturbation. The perturbation causes the algorithm to move away from the local extremum and into the newly accessible area of the search space. A local search technique based on the surrounding area is also planned to provide better results, l_z. It starts with the present solution and descends to a local optimum. Furthermore, the specified is an accepted metric for finalizing the solution chosen for further development. If the new solution is of higher quality, it will replace the old one. There is no end to the ITLS process unless a specific condition is met.

3.9.1. Perform Local Search and Identify Neighborhood Features

The local search involves discovering a locally optimal solution by probing the area around a given one. Local search iteratively progresses towards an optimum solution or meets some other requirement by making small, incremental adjustments to each potential solution in the feasible space. If a neighbourhood connection can be found on the solutions inside the feasible space, a local search can begin at a solution and progress iteratively to a neighbour one. The layout of the neighbourhood structure is a key challenge in developing a local search. The neighbourhood structure dictates how to get to a new solution by tweaking the old one. An important job is creating a neighbourhood structure that facilitates the transition from one answer to a better one. Several neighbourhood structures are used to steer the algorithm towards better outcomes in less time because the local solution acquired by one neighbourhood is not always the local ideal achieved by another.

3.9.2 Perform ITLS-based Perturbation

The neighbourhood movements in the local search described address RKs that are accessible and rearrange the components of the present solution. In actuality, local search allows looking for viable places in the closest vicinity to the existing solution. This kind of movement is concerned with the search process's intensification approach. It will enable us to find a different solution in a diversified manner. The perturbation is used to prevent the algorithm from becoming stuck in local optimization and to explore new spaces of viable areas. This is precisely what a perturbation approach aims to achieve. The high level of diversity makes the ITLS act like a random restart mechanism, which makes it very unlikely that reasonable solutions will be found. Also, if the disturbance is not severe enough, the ITLS cannot leave the current local region and will return to the current calculation extremum.

4. About the Dataset

The Amazon product review dataset is a comprehensive customer feedback collection that offers insights into product performance, user preferences, and customer satisfaction levels. The dataset is diverse and spans various categories, including books, electronics, clothing, and more, making it a valuable resource for businesses and researchers. One of the most notable features of the Amazon product review dataset is its large size, with millions of reviews and ratings available for analysis. This abundance of data makes it possible to identify trends and patterns across various product categories, which can be used to inform marketing and product development strategies. In addition to the core features of product ID, reviewer ID, review text, rating, helpful votes, total votes, review date, and product category, the dataset also includes product metadata. This information provides additional context for the reviews, such as the brand, model number, and price of the reviewed product. The Amazon product review dataset can be used for various purposes, including sentiment analysis, recommendation systems, and market research. Its accessibility and size make it popular for machine learning, data science projects, and academic research. The Amazon product review dataset is valuable for businesses looking to understand their customers better and researchers looking to gain insights into user behaviour and preferences.

5. Performance Metrics

- True Positive (TP): A sentiment analysis system accurately detects and labels the presence of positive sentiment in a text.
- True Negative indicates an accurate text classification without negative sentiment in sentiment analysis.
- In sentiment analysis, a False Positive occurs when a model incorrectly classifies a text with neutral sentiment as negative.
- In sentiment analysis, a False Negative occurs when the model incorrectly classifies a text with negative sentiment as neutral.

The abovementioned variables are used to calculate four standard performance metrics in this research.

- The accuracy of a classification model is a measure of how well the model can correctly predict the class labels of the instances in a dataset.
- The F-measure is a statistical measure that combines precision and recall into a single score to provide a balanced evaluation of a model's performance.

- The Fowlkes-Mallows Index is a statistical measure that evaluates the degree of agreement between two clusterings of the same dataset.
- The Matthews Correlation Coefficient (MCC) is a statistical measure that evaluates the performance of a binary classifier by measuring the correlation between the predicted and actual binary classifications.

6. Results and Discussion

6.1. CA and FM Analysis

The graph presents a comparison of the classification accuracy (CA) and F-measure (FM) of three different sentiment analysis classifiers: Equanimous Intelligent Water Drop Algorithm-based Feed-Forward Neural Network (EIWD-FFNN), Large-Scale Group Decision-Making System based on Sentiment Analysis Cluster (LSGDMS), and Joint Multimodal Sentiment Analysis (JMSA). The CA and FM values of each classifier are shown on the graph. The proposed EIWD-FFNN classifier achieved the highest CA value of 90.173%, indicating that it could classify the input data accurately. Similarly, the EIWD-FFNN classifier also achieved the highest FM value of 91.632%, indicating that it could classify the data with high precision and recall. On the other hand, the LSGDMS classifier had the lowest CA value of 52.366%, while JMSA had a slightly higher CA value of 61.424%.

Regarding FM, LSGDMS had the lowest value of 51.744%, while JMSA had a higher value of 63.285%. It is important to note that the results were based on a specific dataset and evaluation method, and the performance of the classifiers may vary depending on the data and evaluation criteria used. The EIWD-FFNN classifier is based on an intelligent water drop algorithm and feed-forward neural network, commonly used machine learning techniques for various classification tasks. The graph results indicate that the proposed EIWD-FFNN classifier outperforms the existing LSGDMS and JMSA classifiers regarding CA and FM, making it a promising option for sentiment analysis applications requiring high accuracy and precision. The result values of Figure 1 are provided in Table 2

6.2. FMI and MCC Analysis

The graph displays a comparison of the Fowlkes-Mallows Index (FMI) and Matthews Correlation Coefficient (MCC) of three sentiment analysis classifiers: Equanimous Intelligent Water Drop Algorithm-based Feed-Forward Neural Network (EIWD-FFNN), Large-Scale Group Decision-Making System based on Sentiment Analysis Cluster (LSGDMS), and Joint Multimodal Sentiment Analysis (JMSA). The FMI and MCC values for each classifier are presented on the graph.

Table 2. Result values of CA and FM

	LSGDMS	JMSA	EIWD-FFNN
CA	52.366	61.424	90.173
FM	51.744	63.285	91.632

Table 3. Result values of FMI and MCC

	LSGDMS	JMSA	EIWD-FFNN
FMI	51.747	63.286	91.635
MCC	4.727	22.649	79.747

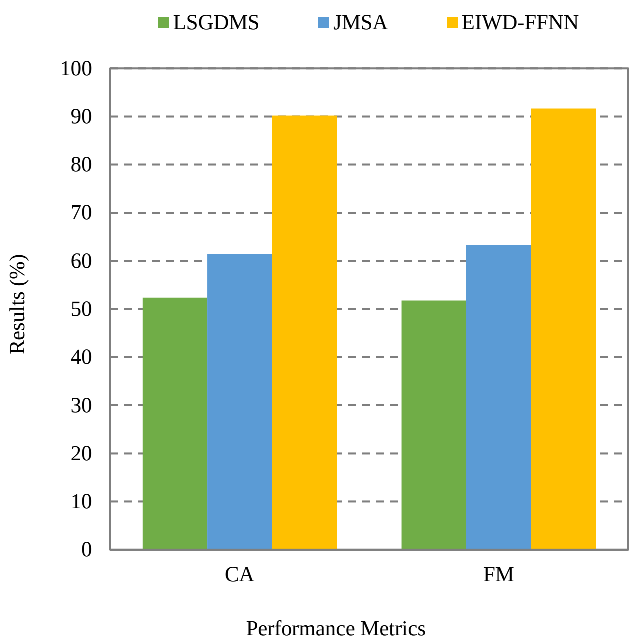

Fig. 1 CA and FM Analysis

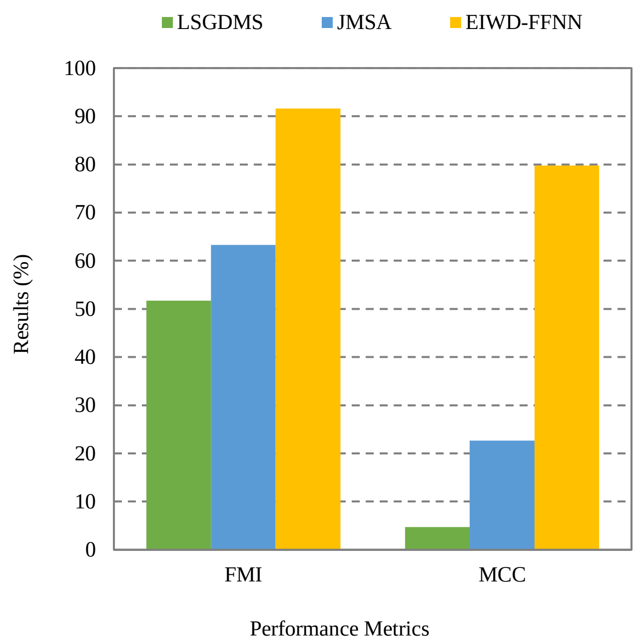

Fig. 2 FMI and MCC Analysis

The proposed EIWD-FFNN classifier achieved the highest FMI value of 91.635%, indicating that it could classify the input data with high precision and recall. Similarly, the EIWD-FFNN classifier also achieved the highest MCC value of 79.747%, showing a strong correlation between the predicted and actual sentiment labels. In contrast, the LSGDMS classifier had the lowest FMI value of 51.747%, while JMSA had a slightly higher FMI value of 63.286%. For MCC, LSGDMS had the lowest value of 4.727%, while JMSA had a higher value of 22.649%. It is essential to note that the results were based on a specific dataset and evaluation method, and the performance of the classifiers may vary depending on the data and evaluation criteria used. The EIWD-FFNN classifier uses an intelligent water drop algorithm and a feed-forward neural network.

The graph results indicate that the proposed EIWD-FFNN classifier outperforms the existing LSGDMS and JMSA classifiers regarding both FMI and MCC, making it a promising option for sentiment analysis applications that require high precision and strong correlation between predicted and actual sentiment labels.

7. Conclusion

In this research paper, we proposed a new approach to sentiment analysis in online shopping reviews using the Equanimous Intelligent Water Drop Algorithm-based Feed-Forward Neural Network (EIWD-FFNN). Our proposed algorithm is a combination of the EIWD algorithm and the FFNN, which optimizes the hyperparameters of the network to achieve better classification accuracy. The results of our experiments on a dataset of customer reviews from Amazon demonstrate the effectiveness of the proposed algorithm. The proposed EIWD-FFNN algorithm outperforms other state-of-the-art sentiment analysis techniques, achieving an accuracy of 90.17%.

Our study highlights the importance of developing accurate sentiment analysis techniques to help businesses understand customers' feedback and improve their products and services. The proposed algorithm offers a new approach to enhance sentiment analysis accuracy, even in ambiguous language and a lack of labelled data. Future work may focus on applying the proposed algorithm to other domains and expanding the range of applications.

Additionally, the proposed algorithm can be extended to analyze different aspects of customer reviews, such as identifying the most common complaints or compliments, which could provide further insight into customer feedback. Overall, our proposed EIWD-FFNN algorithm offers a promising approach to sentiment analysis. It could be an essential tool for businesses to improve their understanding of customer feedback in online shopping platforms.

References

[1] Nawapon Kewsuwun, and Siriwan Kajornkasirat, "A Sentiment Analysis Model of Agritech Startup on Facebook Comments using Naive Bayes Classifier," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, pp. 2829–2838, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Maritza Bustos López et al., "EduRP: An Educational Resources Platform Based on Opinion Mining and Semantic Web," *Journal of Universal Computer Science*, vol. 24, no. 11, pp. 1515-1535, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Youness Madani et al., "Social Collaborative Filtering Approach for Recommending Courses in an E-Learning Platform," *Procedia Computer Science*, vol. 151, pp. 1164–1169, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sharath Narayana, and Sushmita Manikandan, "Smart Learning through Enhanced Concentration and Contextual Search using Eyeball Tracker," *International Conference on Energy, Communication, Data Analytics and Soft Computing*, pp. 532–536, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Meifang Chen, and Fei Fan, "Application of LCF Model based on Deep Learning in Campus Network Platforms," *IEEE 5th International Conference on Information Systems and Computer Aided Education*, pp. 113–117, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Carlos Arcila-Calderón, "Communication, Education and Social Media," *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pp. 920–922, 2018. [CrossRef] [Publisher Link]

[7] Garima Chanana, V. Vijaya Kumar, and M. Geetha, "Understanding Students' Learning Experiences by Mining Social Media Data," *International Journal of Recent Technology and Engineering*, vol. 7, no. 6, pp. 1392–1398, 2019. [Publisher Link]

[8] Robin G. Qiu et al., "A Big Data Based Smart Evaluation System using Public Opinion Aggregation," *Proceedings of the 18th International Conference on Enterprise Information Systems*, pp. 520–527, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Yufei Zeng et al., "Heterogeneous Graph Convolution Based on in-Domain Self-Supervision for Multimodal Sentiment Analysis," *Expert Systems with Applications*, vol. 213, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ankita Gandhi et al., "Multimodal Sentiment Analysis: A Systematic Review of History, Datasets, Multimodal Fusion Methods, Applications, Challenges and Future Directions," *Information Fusion*, vol. 91, pp. 424–444, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jiahui Wen et al., "Hybrid Sentiment Analysis with Textual and Interactive Information," *Expert Systems with Applications*, vol. 213, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Sana Rahmani et al., "Transfer-Based Adaptive Tree for Multimodal Sentiment Analysis Based on User Latent Aspects," *Knowledge-Based Systems*, vol. 261, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Mouthami Kuppusamy, and Anandamurugan Selvaraj, "A Novel Hybrid Deep Learning Model for Aspect Based Sentiment Analysis," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Abdussalam Mamoon, Adamu Umar Alhaji, and Ibrahim Abdullahi, "Application of Neur*al Network for Material Selection: A Review,*" *SSRG International Journal of Material Science and Engineering*, vol. 7, no. 2, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Gagandeep Kaur, and Amit Sharma, "A Deep Learning-Based Model using Hybrid Feature Extraction Approach for Consumer Sentiment Analysis," *Journal of Big Data*, vol. 10, no. 5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] R. Bharathi, R. Bhavani, and R. Priya, "Leveraging Deep Learning Models for Automated Aspect Based Sentiment Analysis and Classification," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 10, no. 5, pp. 120-130, 2023. [CrossRef] [Publisher Link]

[17] Halima Benarafa, Mohammed Benkhalifa, and Moulay Akhloufi, "WordNet Semantic Relations Based Enhancement of KNN Model for Implicit Aspect Identification in Sentiment Analysis," *International Journal of Computational Intelligence Systems*, vol. 16, no. 3, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Muhammet Sinan Başarslan, and Fatih Kayaalp, "MBi-GRUMCONV: A Novel Multi Bi-GRU and Multi CNN-Based Deep Learning Model for Social Media Sentiment Analysis," *Journal of Cloud Computing*, vol. 12, no. 5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Ferdaous Benrouba, and Rachid Boudour, "Emotional Sentiment Analysis of Social Media Content for Mental Health Safety," *Social Network Analysis and Mining*, vol. 13, no. 17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Omar Mohamed et al., "An Ensemble Transformer-Based Model for Arabic Sentiment Analysis," *Social Network Analysis and Mining*, vol. 13, no. 11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] A. Senthilkumar et al., "Minimizing Energy Consumption in Vehicular Sensor Networks using Relentless Particle Swarm Optimization Routing," *International Journal of Computer Networks and Applications*, vol. 10, no. 2, pp. 217–230, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] J. Ramkumar et al., "IoT-Based Kalman Filtering and Particle Swarm Optimization for Detecting Skin Lesion," *Soft Computing Applications in Modern Power and Energy Systems*, vol. 975, pp. 17-27, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] R. Vadivel, and J. Ramkumar, "QoS-Enabled Improved Cuckoo Search-Inspired Protocol (ICSIP) for Iot-Based Healthcare Applications," *Incorporating the Internet of Things in Healthcare Applications and Wearable Devices*, pp. 109–121, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[24] Ramkumar Jaganathan, and Vadivel Ramasamy, "Performance Modeling of Bio-Inspired Routing Protocols in Cognitive Radio Ad Hoc Network to Reduce End-to-End Delay," *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 1, pp. 221–231, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[25] Ramkumar Jaganathan, and Vadivel Ramasamy, "Improved Frog Leap Inspired Protocol (IFLIP) – for Routing in Cognitive Radio Ad Hoc Networks," *World Journal of Engineering*, vol. 15, no. 2, pp. 306–311, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[26] Ramkumar Jaganathan, and Vadivel Ramasamy, "CSIP—Cuckoo Search Inspired Protocol for Routing in Cognitive Radio Ad Hoc Networks," *Advances in Intelligent Systems and Computing*, vol. 556, pp. 145–153, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[27] Lingaraj Mani, Senthilkumar Arumugam, and Ramkumar Jaganathan, "Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol," *Proceedings of the 4th International Conference on Information Management & Machine Intelligence*, no. 70, pp. 1-5, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] D. Jayaraj et al., "AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network," *International Journal of Computer Networks and Applications*, vol. 10, no. 1, pp. 119–129, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[29] Ramkumar Jaganathan et al., "Energy Consumption Minimization in Cognitive Radio Mobile Ad-Hoc Networks using Enriched Ad-hoc on-demand Distance Vector Protocol," *International Conference on Advanced Computing Technologies and Applications*, pp. 1–6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[30] Ramkumar Jaganathan, and Vadivel Ramasamy, "Robust Optimization Based Extreme Learning Machine for Sentiment Analysis in Big Data," *International Conference on Advanced Computing Technologies and Applications*, pp. 1–5, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[31] Ramkumar Jaganathan, Vadivel Ramasamy, and B. Narasimhan, "Constrained Cuckoo Search Optimization Based Protocol for Routing in Cloud Network," *International Journal of Computer Networks and Applications,* vol. 8, no. 6, pp. 795–803, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] Ramkumar Jaganathan, and Vadivel Ramasamy, "Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks," *Wireless Personal Communications*, vol. 120, pp. 887–909, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[33] Ramkumar Jaganathan, and Vadivel Ramasamy, "Whale Optimization Routing Protocol for Minimizing Energy Consumption in Cognitive Radio Wireless Sensor Network," *International Journal of Computer Networks and Applications*, vol. 8, no. 4, pp. 455–464, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[34] Ramkumar Jaganathan, and Vadivel Ramasamy, "Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks," *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 1063–1074, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[35] Ramkumar Jaganathan, and Vadivel Ramasamy, "Improved Wolf Prey Inspired Protocol for Routing in Cognitive Radio Ad Hoc Networks," *International Journal of Computer Networks and Applications*, vol. 7, no. 5, pp. 126–136, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[36] Ramkumar Jaganathan, and Vadivel Ramasamy, "Meticulous Elephant Herding Optimization based Protocol for Detecting Intrusions in Cognitive Radio Ad Hoc Networks," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 8, pp. 4548–4554, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[37] José Ramón Trillo et al., "A Large Scale Group Decision Making System Based on Sentiment Analysis Cluster," *Information Fusion*, vol. 91, pp. 633–643, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[38] Danlei Chen et al., "Joint Multimodal Sentiment Analysis Based on Information Relevance," *Information Processing and Management*, vol. 60, no. 2, 2023. [CrossRef] [Google Scholar] [Publisher Link]