Journal of Theoretical and Applied Information Technology

15th December 2023. Vol.101. No 23
© 2023 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HARMONY SEARCH OPTIMIZATION-BASED GENERATIVE ADVERSARIAL NETWORKS (HSO-LGAN) FOR ENHANCING SENTIMENT ANALYSIS IN AUGMENTED REALITY-ENABLED APPLICATIONS

PRAGATHI ARAVABOOMI[1], T.K.SARAVANAN[2]

[1] Department of Journalism & Communication, SRM Institute of Science & Technology, Kattankulathur, India

[2] Department of Visual Communication, SRM Institute of Science & Technology, Kattankulathur, India

[1]pa8399@srmist.edu.in, [2]saravank1@srmist.edu.in

ABSTRACT

This paper explores the transformative dynamics of online shopping, emphasizing the pivotal role of augmented reality (AR) in reshaping the customer experience. In the digital marketplace, customer product reviews serve as a rich source of feedback, providing valuable insights for businesses. However, the nuanced sentiments within these reviews pose a significant challenge for effective analysis. Acknowledging this complexity, the proposed work introduces "Harmony Search Optimization-Based Generative Adversarial Networks (HSO-LGAN)" as an innovative solution. HSO-LGAN leverages harmony search optimization algorithms within generative adversarial networks to translate textual sentiments from product reviews into immersive AR experiences. The working mechanism involves a synergistic approach, enhancing the generative adversarial network's capabilities to create personalized and emotionally resonant AR content. Through comprehensive evaluation, the paper assesses the effectiveness of HSO-LGAN in improving customer engagement, satisfaction, and emotional attachment to brands. Results and discussions from experimental studies across diverse industries highlight the potential of HSO-LGAN to significantly impact customer loyalty and advocacy in the evolving landscape of online shopping. This research contributes to the broader understanding of leveraging AR and sentiment analysis to optimize the online shopping experience and bridge the gap between customer feedback and interactive AR elements.

Keywords: *Augmented Reality, Classification, Gan, Harmony Search, Sentiment Analysis, Optimization.*

1. INTRODUCTION

Online shopping is continually evolving, shaped by technological advancements and consumer preferences. Exciting trends and innovations are defining the future of e-commerce. Augmented Reality (AR) shopping allows customers to visualize products in their spaces, revolutionizing the experience, particularly for furniture and clothing [1]. Artificial intelligence (AI) recommendations use past purchase history and browsing behavior to provide personalized product suggestions, simplifying the discovery of new items. Sustainability and eco-friendly shopping are gaining prominence, with many online retailers offering environmentally conscious products and reducing packaging waste [2]. Voice commerce via virtual assistants like Alexa and Siri is rising, enabling consumers to purchase through voice commands. Same-day and drone delivery options are increasingly accessible, ensuring faster shipping choices [3]. Virtual shopping events and livestreams are creating engaging communities for online shoppers. Blockchain technology enhances supply chain transparency, allowing consumers to trace the origin and authenticity of products. These trends are shaping an even more personalized, efficient, and sustainable future for online shopping [4].

Online shopping has undergone a remarkable transformation with the integration of AR technology. This innovation bridges the digital and physical shopping experience gap, offering consumers a new way to interact with products online [5]. AR-enabled online shopping allows customers to visualize and interact with products in real-world environments through smartphones or other devices. This means you can see how that new piece of furniture fits in your living room, how shoes complement your outfit, or even how a cosmetic product looks on your skin, all without leaving your home. This immersive shopping experience enhances convenience and boosts confidence in purchase decisions, making online shopping more interactive and engaging than ever before [6].

Sentiment analysis, or opinion mining, is a powerful tool in product reviews. It employs natural language processing and machine learning to analyze the emotions and attitudes expressed within customer feedback. Beyond categorizing reviews as positive, negative, or neutral, sentiment analysis delves deeper, providing nuanced insights into the reasons behind these sentiments [7]. Companies can understand whether customers are satisfied or dissatisfied and the specific aspects of their products or services that drive these sentiments. This level of granularity empowers businesses to make data-driven decisions to enhance their offerings and strengthen customer relationships [8].

Sentiment analysis is a valuable tool that provides significant advantages to companies and manufacturers across various industries. Businesses gain insights into consumer preferences, satisfaction levels, and pain points by analyzing the sentiments expressed in customer reviews, social media mentions, and other feedback forms [9]. This data-driven approach enables companies to make informed decisions about product development, marketing strategies, and customer service improvements [10]. A smartphone manufacturer can use sentiment analysis to identify common themes in customer feedback, helping them prioritize features and enhancements for their next product release. In essence, sentiment analysis empowers companies to align their offerings more closely with customer expectations, increasing customer satisfaction and loyalty [11].

The significance of bio-inspired optimization in this work lies in its capacity to elevate the accuracy of sentiment analysis within online shopping. Drawing inspiration from natural processes, these optimization methods contribute to the adaptability and efficiency of the sentiment analysis model [12]–[26]. This bio-inspired approach enhances the system's ability to discern subtle nuances in customer sentiments expressed in product reviews. Consequently, the sentiment analysis becomes more robust and effective, providing businesses deeper insights into customer preferences and emotions in online shopping. This heightened accuracy, facilitated by bio-inspired optimization [27], can transform customer-brand interactions, fostering improved engagement, satisfaction, and brand loyalty in the ever-evolving landscape of online commerce.

1.1. Problem Statement (polarity-based problem statement)

Analyzing the polarity of customer sentiments within reviews of AR-enabled shopping apps presents a pressing problem. As AR technology becomes more integrated into the e-commerce sector, understanding how users perceive and react to these apps becomes crucial. The problem statement revolves around developing effective sentiment analysis techniques that accurately classify customer reviews as positive, negative, or neutral. This entails addressing the challenges of parsing complex and nuanced language in user-generated content, identifying sarcasm and context, and dealing with the inherent subjectivity of sentiment. Additionally, the problem extends to the need for domain-specific sentiment analysis models that account for the distinct features and nuances of AR shopping apps, which may not align with standard sentiment analysis models designed for more general text data. Solving this problem is imperative for businesses to make data-informed decisions, enhance user experiences, and stay competitive in the rapidly evolving AR-enabled shopping app market.

1.2. Motivation

The motivation for addressing the problem of analyzing sentiment polarity within reviews of AR-enabled shopping apps is grounded in the evolving landscape of e-commerce and the pivotal role of augmented reality technology. With AR technology increasingly integrated into the e-commerce sector, it has become paramount for businesses to gain insights into how users perceive and interact with these innovative apps. Understanding user sentiment is crucial, as it offers a window into the effectiveness of AR shopping experiences. By accurately classifying reviews as positive, negative, or neutral, we can unlock valuable information for businesses. Parsing the intricate language used in user-generated content, detecting nuances such as sarcasm, and considering the subjective nature of sentiment are all challenges that, when surmounted, empower companies to make informed decisions. Moreover, the motivation stems from the necessity to develop domain-specific sentiment analysis models that cater to the unique characteristics of AR shopping apps, ultimately enabling businesses to maintain their competitive edge in a rapidly evolving market.

1.3. Objectives

The primary objective of this research is to devise an innovative and highly effective sentiment analysis classification algorithm explicitly tailored for the reviews of AR-enabled shopping apps. This algorithm aims to accurately classify customer feedback as positive, negative, or neutral, addressing the complexities inherent in parsing the intricate and nuanced language often present in user-generated content. By developing a specialized domain-specific model, the research seeks to overcome the challenges of standard sentiment analysis techniques that may not fully capture the unique features and context of AR shopping apps. Furthermore, the objective is to provide businesses and app developers with a powerful tool to extract precise sentiment insights from customer reviews, enabling data-driven decision-making and enhancing the overall AR shopping experience. This research contributes to advancing sentiment analysis methodologies in the context of AR-enabled shopping apps, facilitating improved user satisfaction and competitiveness in the ever-evolving landscape of e-commerce and augmented reality technologies.

2. LITERATURE REVIEW

“Hybrid Approach” [28] in sentiment analysis of Amazon user reviews is a promising methodology combining the strengths of rule-based and machine learning techniques. By examining existing research, it becomes evident that this approach leverages predefined rules and machine learning algorithms to achieve more accurate sentiment classification. “Structural Topic Modeling (STM)” [29] offers a novel approach to predicting user sentiments regarding intelligent personal agents like Amazon’s Echo and Google Home. This technique enables a deeper understanding of user-generated content by considering content and its underlying structure. STM leverages probabilistic graphical models to identify critical topics within a dataset, linking them to sentiment scores.

“Fake Review Detection” [30] in e-commerce platforms can benefit significantly from the application of “Aspect-Based Sentiment Analysis” (ABSA). By leveraging ABSA, it becomes possible to identify review anomalies, flagging those that exhibit suspicious sentiment patterns, incoherent aspect sentiments, or disproportionately positive or negative feedback for specific product features. ABSA is pivotal in automating the detection process in e-commerce, where fake reviews can significantly impact consumer trust and purchasing decisions.

“Leveraging User Sentiment Orientation” [31] in the context of sentiment analysis within social networks is a promising avenue for understanding and categorizing user sentiments. It focuses on mining the sentiment orientation of individual users, which refers to their habitual expression of positive, negative, or neutral sentiments across their social network interactions. “Hybrid Optimization Algorithm” [32] incorporates a Bidirectional Long Short-Term Memory (BiLSTM) structure for sentiment analysis. It represents an innovative approach to enhance the accuracy and efficiency of sentiment classification in text data. In this, the BiLSTM structure serves as a deep learning component to model language's sequential nature effectively. “Integrated Deep Learning Paradigm” [33] for document-based sentiment analysis represents a sophisticated and comprehensive approach to extracting sentiment information from textual documents. This methodology combines deep learning techniques to achieve a more nuanced understanding of sentiment within a copy. These deep learning architectures enable the model to capture local and global context within the text, recognizing patterns, linguistic nuances, and sentiment cues.

“Hybrid Recommendation System” [34] incorporates product review sentiment and leverages sentiment analysis to enhance user profiles with emotional preferences and enrich item profiles with sentiment awareness. By fusing traditional recommendation algorithms with sentiment information, this approach delivers personalized recommendations that align with users' historical behaviors and resonate with their emotional responses to products. "Online Product Sentiment Analysis" [35] merges the "Random Evolutionary Whale Optimization Algorithm" for efficient data collection and preprocessing with the "Deep Belief Network" as a powerful sentiment analysis model. This integrated approach enables real-time monitoring and nuanced analysis of product sentiments across online platforms, extracting fine-grained insights from vast streams of user-generated content like reviews and social media comments.

"Topic-Driven Adaptive Network (TDAN)" [36] introduces an innovative approach to sentiment analysis, specifically tailored for the challenges of sentiment classification across diverse domains. This methodology leverages topic-driven adaptive networks to adapt to the unique nuances of sentiment expression in different contexts. The model enhances its accuracy and context awareness in sentiment analysis by considering topic information, making it particularly valuable for market research, product reviews, and social media sentiment-tracking applications.

"Deep Ensemble Learning Technique (DELT)" [37] is a groundbreaking approach poised to transform recommendation systems by integrating sentiment analysis with deep ensemble learning. Understanding user sentiment is paramount for providing personalized and relevant recommendations in recommendation systems. DELT achieves this by leveraging deep ensemble learning techniques, combining multiple models' predictive strengths. This enhances the precision of recommendations and ensures they align with user sentiments. This approach has vast implications for industries such as e-commerce, where product recommendations based on user sentiment can significantly impact sales and user satisfaction.

3.HARMONY SEARCH OPTIMIZATION BASED LAGRANGIAN GENERATIVE ADVERSARIAL NETWORKS (HSO-LGAN)

3.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are machine learning models for generative tasks, such as generating images, music, text, and more. GANs consist of two neural networks, the generator and the discriminator, which are trained simultaneously through a competitive process. The generator tries to produce indistinguishable data from real data, while the discriminator tries to differentiate between real and generated data.

Algorithm 1: Core Generative Adversarial Networks

```
Input:
Real data samples
Random noise vectors
Hyperparameters (e.g., learning rate, batch size, epochs)

Output:
Trained generator network

Procedure:
Step 1: Initialize generator and discriminator networks.
Step 2: Define generator and discriminator loss functions.
Step 3: Training loop for a set number of epochs:
    Train discriminator:
        Sample real data and generate fake data.
        Update discriminator weights.
    Train generator:
        Generate fake data and update generator weights.
Step 4: Monitor training progress.
Step 5: After training, use the generator to produce new data.
Step 6: Evaluate and fine-tune as needed.
Step 7: Deploy the trained generator for generative tasks.
```

3.2. Lagrangian GAN (LGAN)

The primary objective of introducing Lagrange multipliers in GANs is to incorporate additional constraints into the training process to achieve specific goals or control certain aspects of the generated data. A standard GAN has two networks, the generator (G) and the discriminator (D), and two loss functions, one for each network. The generator attempts to minimize its loss, while the discriminator strives to minimize it. These losses are generally independent of each other.

In the Lagrangian GAN $(L-GAN)$, a Lagrange multiplier term is introduced to enforce a constraint in the optimization problem. The Lagrangian function is a combination of the generator loss, discriminator loss, and the constraint term. The Lagrange multiplier is used to adjust the importance of the constraint concerning the losses. The overall objective is to optimize the Lagrangian function.

3.2.1. Initialization

In this step, the initial conditions are set for the key components of the proposed classifier Lagrangian GAN ($L - GAN$): (a) generator network (G), (b) discriminator network (D), and (c) Lagrange multiplier (λ).

(a) Generator Network (G)

The generator network denoted as G, starts with initial parameters represented by θ_G. These parameters determine how the generator transforms random input noise into generated data samples. The initialization is typically performed by sampling θ_G from a probability distribution. Eq.(1) expresses the initialization of the generator network.

$$\theta_G \leftarrow Sample\ From\ Parameter\ Distribution\ (\ ) \quad (1)$$

(b) Discriminator Network (D):

The discriminator network, denoted as D, also begins with initial parameters represented by θ_D. These parameters determine how D distinguishes between real and generated data. Similar to the G, θ_D is initialized by sampling from a probability distribution. Eq.(2) expresses the initialization of the discriminator network.

$$\theta_D \leftarrow Sample\ From\ Parameter\ Distribution\ () \quad (2)$$

(c) Lagrange Multiplier (λ):

The Lagrange multiplier (λ) is a scalar value that regulates the balance between the constraint and the generator and discriminator losses. It begins with an arbitrary initial value, $\lambda_{initial}$. This value sets the initial strength of the constraint. Eq.(3) is applied to initialize the λ.

$$\lambda \leftarrow \lambda_{initial} \quad (3)$$

This research establishes the starting conditions for the $L - GAN$ training process by initializing the generator, discriminator, and Lagrange multiplier. These initializations set the stage for subsequent iterations, where the generator and discriminator learn to compete and optimize their objectives while adhering to the constraint imposed by λ. The random initialization of network parameters ensures that the $L - GAN$ starts without prior knowledge and learns from data during training.

3.2.2. Constraint Function

The constraint function (C) represents the constraint imposed on the $L - GAN$'s behavior. C is the quantitative measure of how well the generated data adheres to the specified constraint. Eq.(4) indicates the constraint function.

$$C(G(z)) \rightarrow \lambda \quad (4)$$

where C represents the constraint function applied to the generated data $G(z)$, where z is a random noise vector sampled from a predefined distribution.

The constraint function C can take various forms depending on the task and constraints to be enforced. It can be a mathematical function, equations, or conditions expressing the desired property of the generated data. During training, the $L - GAN$ aims to minimize this constraint function while simultaneously optimizing the generator and discriminator. The choice of the constraint function C is problem-specific and should align with the goals of the generative task. Defining an appropriate constraint function is crucial for achieving desired constraints during $L - GAN$ training. A clear mathematical representation of the constraint is established by defining the constraint function C, guiding the $L - GAN$'s learning process. This constraint is central to optimizing the Lagrangian function during training, ensuring the generated data meets the specified criteria.

3.2.3. Lagrangian Function

Lagrangian function, denoted as $L(G, D, \lambda)$, is formulated to combine the generator loss, discriminator loss, and the constraint term. The Lagrange multiplier, λ, is introduced to adjust the importance of the constraint relative to the generator and discriminator losses. The Lagrangian function is expressed in Eq.(5).

$$L(G,D,\lambda) = Generator\ Loss,(LG) - \text{Discriminator loss (LD)} + \lambda - C(G(z)) \quad (5)$$

where LG represents the generator loss, which measures how well the generator network G produces data that resembles real data. This loss is typically a function of the generator's parameters θ_G and is aimed at minimizing the difference between the generated and real data. LD represents the discriminator loss, which measures how well the discriminator network D can distinguish between real and generated data. It is typically a function of the discriminator's parameters θ_D and is aimed at maximizing the discriminator's ability to distinguish between real and fake data. λ represents the Lagrange multiplier, introduced to control the trade-off between the constraint $(C(G(z)))$ and the generator and discriminator losses. By adjusting the value of λ, you can control how strictly the constraint is enforced. $C(G(z))$ represents the constraint term, quantifying how well the generated data adheres to the specified constraint. It is typically a function of the generated data $G(z)$ and is intended to be minimized during training while satisfying the constraint.

The Lagrangian function combines these components into a single objective function that the $L-GAN$ seeks to optimize during training. The introduction of λ allows for fine-tuning the balance between generating realistic data, distinguishing real from fake data, and adhering to the imposed constraint. By iteratively adjusting the parameters of the generator, discriminator, and λ, the $L-GAN$ aims to find a balance that meets the desired generative constraints.

Algorithm 2: Lagrangian Function

Input:
Generator Loss (LG)
Discriminator Loss (LD)
Lagrange Multiplier (λ)
Constraint Term $(C(G(z)))$

Output:
A trained generator (G) that can produce data meeting the given constraint and looking realistic.

Procedure:
Initialization: Start with random parameters for the generator, discriminator, and λ.
Training Loop: Repeat the following steps a set number of times or until the model improves sufficiently.
Update Discriminator: Train the discriminator to better distinguish real data from generated data.
Update Generator: Train the generator to make its generated data more similar to real data.
Adjust λ: Tune λ to find the right balance between the constraint and the generator/discriminator losses.
Evaluation (Optional): Occasionally check how well the generated data meets the constraint.
Completion: Keep repeating the training loop until you are satisfied with the results.
Result: Once the training is complete, you can use the trained generator to create data that adheres to the given constraint while appearing realistic.

3.2.4. Training Loop

The L-GAN begins the training loop, where the generator, discriminator, and Lagrange multiplier are updated iteratively. The training loop consists of the following key actions. In each iteration, a batch of random noise vectors (z) is sampled from a predefined distribution. These noise vectors serve as input to the generator to produce generated data samples. The discriminator network (D) is updated first to improve its ability to distinguish between real data from the dataset and generated data produced by the generator. This is achieved by minimizing the discriminator loss (LD) and encouraging D to make accurate classifications. The update of D's parameters (θ_D) is performed by gradient descent, and Eq.(6) expresses the same.

$$\theta_D \leftarrow \theta_D - \alpha.\nabla_{\theta_D} LD \quad (6)$$

where θ_D represents the discriminator's parameters, α represents the learning rate, and a small positive value that controls the step size during optimization ∇ represents the gradient operator.

After updating the discriminator, the generator network (G) is updated to produce more realistic data. The generator loss (LG) is minimized, encouraging G to generate data that fools the discriminator. The update of G's parameters (θ_G) is also performed by gradient descent, and Eq.(7) expresses the same.

$$\theta_G \leftarrow \theta_G - \alpha.\nabla_{\theta_G} LG \qquad (7)$$

where θ_G represents the generator's parameters, α represents the learning rate and ∇ the gradient operator.

The Lagrange multiplier (λ) is adjusted to satisfy the constraint while considering the performance of the generator and discriminator. The update of λ is typically based on a gradient-based method or other optimization techniques to find the right balance between the constraint term and the generator and discriminator losses.

3.2.5. End Training

The training process of the $L-GAN$ ends after completing a predefined number of training iterations or when convergence criteria are met. The key actions in this step are as follows: (a) convergence check, (b) Training Termination, (c) Trained $L-GAN$ and (d) fine-tuning.

(a) Convergence Check

During the training loop, the performance of the $L-GAN$ is monitored at each iteration. Convergence criteria are defined to determine when training should stop. Standard convergence criteria include checking whether the LG has stabilized, the LD is no longer improving significantly, or when the generated data meets the desired constraint satisfactorily.

(b) Training Termination

Once the predefined convergence criteria are satisfied, the training process is terminated. At this point, the $L-GAN$ is considered to have learned the generative task and has adjusted the parameters of the generator (θ_G), discriminator (θ_D), and Lagrange multiplier (λ) to achieve the desired balance between generating realistic data and adhering to the constraint.

(c) Trained LGAN

The result of this step is a trained $L-GAN$ model with optimized parameters. The generator can produce data samples that meet the imposed constraint, and the discriminator has learned to distinguish between real and generated data effectively. After training, additional evaluations of the generated samples are expected to ensure that the constraint is satisfied to the desired degree. Depending on the results, further fine-tuning of the Lagrange multiplier, constraint function, or other model parameters may be necessary to improve the quality of generated data and constraint adherence.

Algorithm 3: End of Traning

```
Input:
Convergence criteria.
Generator parameters ($\theta_G$)
Discriminator parameters ($\theta_D$)
Lagrange multiplier ($\lambda$).

Output:
Trained $L-GAN$ model.
Optimized parameters.

Procedure:
Continuously monitor training progress.
Halt training when convergence criteria are satisfied.
The result is a trained $L-GAN$ model with optimized parameters.
Optionally, refine the model further if necessary for improved performance.
```

3.2.6. Generation

The trained $L-GAN$ model generates new data samples that adhere to the specified constraint. This process involves using the generator network to produce data samples from random noise vectors. Given the trained generator network (G) and a set of random noise vectors (z) sampled from a predefined distribution, the $L-GAN$ generates new data samples using Eq.(8).

$$Generated\ Data: G(z) \qquad (8)$$

where $G(z)$ represents the generated data samples produced by the generator network using the noise vectors (z) as input.

The generated data samples are subject to the constraint function (C) to ensure they meet the desired constraint, and Eq.(9) is applied for that.

$$ConstraintSatisfunction : C(G(z)) \qquad (9)$$

where $C(G(z))$ quantifies how well the generated data adheres to the specified constraint. The goal is to produce data that satisfies this constraint.

The generated data samples $(G(z))$ are evaluated to confirm that they align with the desired characteristics or properties defined by the constraint. The extent to which the generated data adheres to the constraint can be assessed mathematically. The generated data samples, which satisfy the imposed constraint, can be utilized for various applications, depending on the specific generative task. These applications may include image generation, text generation, data synthesis, or any other task requiring data to meet specific constraints.

3.2.7. Evaluation and Fine-Tuning

This step evaluates $L-GAN$ generated data samples to satisfy the desired constraint. If necessary, additional fine-tuning may be performed to improve constraint adherence. The two key actions involved in this are:

(a) Data Evaluation:

The generated data samples $(G(z))$ are rigorously evaluated to quantify how well they conform to the specified constraint. The data evaluation is mathematically expressed as Eq.(10).

$$Evaluate \rightarrow (G(z)) \qquad (10)$$

(b) Assessment of Constraint:

The constraint assessment involves analyzing the evaluation results to determine whether the generated data satisfies the constraint satisfactorily. This can be done mathematically by comparing the evaluation results against predefined criteria, and Eq.(11) represents the same.

$$Constraint\ Satisfaction \rightarrow Assess(G(z)) \qquad (11)$$

3.3. Harmony Search Optimization (HSO)

The default Harmony Search (HS) algorithm operates with a core mechanism that generates a new harmony during each iteration to update the Harmony Memory (HM). This newly generated harmony is constructed based on one of three regulations:

- **Random Selection:** A pitch is randomly chosen from the HM.

- **Modification:** An existing pitch from the HM is arbitrarily selected and modified.

- **Random Generation:** A pitch is created entirely at random.

The first regulation is rooted in classic HS memory considerations, heavily relying on knowledge stored in the HM. However, it has limitations as it only retains randomly selected harmonies from the HM, potentially neglecting valuable information that could guide the search process. This limitation results in HS excelling in exploration but falling short in exploitation. To address this imbalance and improve both exploration and exploitation, this research introduces three additional pitch selection and creation regulations based on statistical concepts:

- **Best Regulation (BR):** The BR utilizes knowledge of the harmonies with the highest fitness values in the HM, guiding the search toward chord progressions with superior performance.

- **Suboptimal Regulation (SR):** It leverages the knowledge of the minor fit harmonies in the HM, aiding in identifying areas to avoid or explore further.

- **Mean Regulation (MR):** This regulation calculates the average pitch values from the HM, generating a mean harmony that provides insights into central tendencies within the stored harmonies.

A probabilistic self-adaptive selection process is implemented to select each new harmony note. This process dynamically determines whether to apply the best SR, balancing exploitation and exploration across various search phases. It ensures the algorithm effectively explores the solution space while exploiting valuable information from the HM. The suggested HSO method enhances the standard HS algorithm by generating two new harmonies in each iteration instead of just one. The first harmony is formed using the MR, while the second follows a stochastic approach, alternating between the best and SRs to calculate each pitch. This stochasticity introduces an element of randomness that can aid in exploration.

3.3.1. Best Regulation (BR)

Numerous adaptations of the HS algorithm have incorporated the concept of the global best harmony, drawing inspiration from the Particle Swarm Optimization (PSO) algorithm, a renowned swarm intelligence technique. These variations have introduced the global best harmony directly or in combination with other harmonies from the HM to be strategically employed during improvisation. Among these variations, this research has identified a formula that consistently yields optimal results when utilizing the HSO algorithm. Eq.(12) mathematically expresses the same.

$$P_{new,w} = P_{Best,w} + rand \times (P_{b,w} - P_{Best,w}) \quad (12)$$

In Eq.(12), within the HM, a randomly selected harmony has the w-th component denoted as $P_{b,w}$ the best harmony present in the HM is represented as $P_{Best,w}$, and the newly generated harmony is indicated as $P_{new,w}$. This regulation is designed to identify chords within the HM that fall within an intermediate quality range relative to the current best chord.

This is because, within the HM, a notable diversity of harmonies exists, creating a spacious area for exploration rather than exploitation, particularly during the initial stages of the search process. As the search process progresses, this regulation gradually narrows down the size of this intermediate-quality zone, thereby enhancing the algorithm's capacity for exploitation.

This formula balances the trade-off between exploration and exploitation within the HSO algorithm. Allowing for exploration in regions where intermediate-quality solutions are likely to be found facilitates the discovery of potentially superior solutions. As the search evolves, the algorithm exploits the knowledge accumulated in the HM to refine the solutions further. This dynamic approach ensures that the HSO algorithm can adapt and optimize its search strategy based on the evolving quality landscape of the solutions within the HM.

Algorithm 4. BR

```
Input:
HM contains various harmonies.
Index of the harmony component, $w$.
Maximum iterations ($max_{iterations}$).
Convergence threshold ($convergence_{threshold}$).

Output:
Return the best harmony found in the list of best harmonies.

Procedure:
Initialize an empty list to store the best harmonies.
Initialize a counter for the current iteration, $iteration_{count}$, and set it to 0.
Initialize a convergence flag, convergence, and set it to False.
While ($iteration_{count} < max_{iterations}$) and (convergence is False):
Calculate the global best harmony, $P_{Best,w}$, within the $HM$.
Randomly select a harmony component $P_{b,w}$ from the $HM$.
Generate a new harmony component, $P_{Best,w}$, using Eq.(12).
Quality Regulation:
Adjust $P_{Best,w}$ to ensure it falls within an intermediate quality range relative to the current best chord.
Update the HM with $P_{Best,w}$ while maintaining its size.
Update the best harmonies list with $P_{Best,w}$ if it outperforms the current best.
Check for convergence:
If the algorithm converges based on predefined criteria, set $convergence = True$.
Increment $iteration_{count}$ by 1.
```

3.3.2. Suboptimal Regulation

The harmony produced by the best regulation tends to approach the best harmony available at any given time, regardless of its fitness value. Consequently, the problem of early convergence may arise when updating the HM

Algorithm 5: SR

Input:
HM and its initial harmonies.
$P_{b,w}$, $P_{Worst,w}$, $rand$ function

Output:
Updated HM that maintains diversity and prevents early convergence.

Procedure:

```
Initialize $HM$ with initial harmonies.
Define $P_{new,w}$ and $rand$.
Iterate to update $HM$
Evaluate harmony fitness.
Identify the best and worst harmonies.
Calculate $P_{new,w}$ using Eq.(13).
Generate a new harmony by incorporating $P_{new,w}$ into an existing harmony and update $HM$.
During optimization stage adaptively select best and $SR$ criteria
Continue iterating to update $HM$ while ensuring diversity and preventing early convergence.
```

with new harmonies, as it could restrict its diversity. The SR incorporates information from the worst harmony pitches into generating new harmonies to address this issue. Eq.(13) expresses SR.

$$P_{new,w} = P_{Worst,w} + rand \times (P_{b,w} - P_{Worst,w}) \quad (13)$$

In HM, a random harmony contains a wth component represented by $P_{b,w}$, while a new harmony possesses a wth component represented by $P_{new,w}$, and the worst harmony within HM is denoted by $P_{Worst,w}$. As mentioned, during the redesigned improvisation phase, one of the two new chord progressions is generated by randomly applying the best and worst criteria. Consequently, the new set of information will be nt in this new harmony result from the simultaneous application of both criteria. The probabilistic selection process is also self-adaptive, with an increased probability of selecting the best SRs during the early and late stages. This adaptive approach ensures that the SR maintains HM's diversity sufficiently high, thereby mitigating the early convergence problem in the later optimization stages.

3.3.3. Mean Regulation (MR)

Individual data is considered by the regulations used to rank HM harmonics from best to worst. Conversely, the purpose of the MR is to enable the extraction of benefits from the information stored in HM at the population or entire history level. Eq.(14) calculates the average of HM's decision variables to determine the harmonic center.

$$P_{new,w} = P_{b,w} \pm rand \times (Mean_w - P_{b,w}) \quad (14)$$

where $P_{new,w}$ represents the wth component of the new harmony, $Mean_w$ denotes the average value of the wth component across all HM harmonies and $P_{b,w}$ signifies the wth component of a randomly selected HM harmonic. In the first search space representation, harmony is depicted as a collection of isolated points. This regulation leads to an adventurous behavior characterized by large jumps as it navigates around the existing harmonies. The chords are widely spaced apart from both the mean harmony and each other. Each harmony approaches the ideal harmony and each other, facilitating exploitative behavior. This behavior seeks a new harmony close to the current best harmony through gradual iterative increments.

3.3.4. Adaptive Decision-Making Process

Recent research suggests that the success of a meta-heuristic optimization algorithm hinges on its ability to strike a delicate balance between exploitation and exploration. This equilibrium is achieved by introducing an $Prag$ parameter, which is used to probabilistically select between optimal and suboptimal procedures for generating pitches in new harmonies. According to Eq.(15), as the iteration count increases, $Prag$ exponentially decreases:

$$\text{Prag} = 0.7 \times exp\left(log(1/7) \times (IT_{count}/MaxIT_{count})\right) \quad (15)$$

When the iteration count is determined by its argument, and the maximum iteration count is defined by the $MaxIT_{count}$ parameter, The $Prag$ parameter favors the BR over the SR for selection at the beginning of the search and conversely at the end. HSO adopts a strategy of making substantial leaps in its search for the best harmony, leveraging the knowledge and experience acquired while exploring the best and other harmonies. The issue of early convergence can be mitigated by encouraging variation within HM, which grows as the search lowers the $Prag$ and the likelihood of choosing the SR gets increased.

In every iteration of HSO, the improvisational process generates two new pitches. The $P^2_{new,w}$ produced by the BR (i.e., Eq.(12)) and the SR (i.e., Eq.(13)), the new pitch generated by the MR (i.e., Eq.(14)) is denoted as $P^1_{new,w}$. When applying these criteria, the parameters $P_{Best,w}, P_{Worst,w}, Mean_w$ and $P_{b,w}$ as representing the best, worst, mean, and random harmonies, respectively. These regulations are grouped even though the random harmonies are separate.

Algorithm 6: MR and Adaptive Decision-Making Process

```
Input:
HM, $Mean_w$, $P_{b,w}$, $Prag$, $IT$, $IT_{count}$

Output:
$P_{new}$

Procedure:
Initialize $P_{new}$ as an empty vector to store the new harmony.
Calculate the Prag parameter using Eq.(15).
For each component w in the harmony vector:
Calculate the new component $P_{new,w}$ using MR as per Eq.(14).
Apply an exploration-exploitation strategy based on $Prag$:
If $rand > Prag$, apply an exploration procedure to $P_{new,w}$.
If $rand \leq Prag$, apply an exploitation procedure to $P_{new,w}$.
Append $P_{new,w}$to the P_new vector for each component $w$.
Return $P_{new}$ as the new harmony.
```

3.3. Fusion of HSO and LGAN

This fusion facilitates sentiment analysis in AR and generates content tailored to user emotions, ensuring a captivating and emotionally resonant AR experience. The fusion of HSO and GANs in the HSO-LGAN framework involves an iterative process. As users engage with AR content, sentiment analysis continuously evaluates their emotional responses. The sentiment data is then fed into the HSO, which adapts AR content to align with the detected emotions, ensuring a harmonious experience. Simultaneously, GANs generate AR elements and visuals that amplify the emotional resonance, making the AR engagement more immersive and captivating. Algorithm 7 expresses the overall work process of HSO-LGAN.

4. ABOUT DATASET

The Lenskart AR Experience Sentiment Dataset, comprising 788 records, is a comprehensive collection of responses gathered from Lenskart customers through questionnaires, surveys, or online forms designed to explore customer sentiments, preferences, and interactions with Lenskart eyewear products, mainly focusing on augmented reality (AR) experiences. This dataset encompasses diverse information, including demographic details, ratings, and qualitative sentiment expressions. It provides insights into customer satisfaction with product quality, pricing, customer service, and online platform usage. Moreover, it delves into AR experiences, assessing their engagement, impact on purchasing decisions, customization, recommendations, and suggestions for enhancement. With 21 fields, this dataset offers a multifaceted view of customer feedback and preferences, making it a valuable resource for businesses aiming to tailor their offerings and improve customer engagement. Researchers can use this dataset for sentiment analysis, offering insights into customer experiences and potential marketing strategies. The authors will share the dataset with all researchers on demand. Ethical considerations, such as data privacy and informed consent, are essential in handling this dataset.

5. PERFORMANCE METRICS

The True Positive (TrP), True Negative (TrN), False Positive (FlP), and False Negative (FlN) plays a significant role in the calculation of performance metrics.

$$MCC = \frac{TrP \times TrN \times FlP \times FlN}{\sqrt{(TrP + FlP) \times (TrP + FlN) \times (TrN + FlP) \times (TrN + FlN)}} \times 100 \quad (20)$$

5.1. Precision

Precision ($Prec$) measures the accuracy of the positive predictions made by a model. Eq.(16) is applied to calculate the $Prec$.

$$Prec = \frac{TrP}{TrP + FlP} \times 100 \quad (16)$$

5.2. Recall

Recall ($Recl$) measures the ability of a model to capture all the relevant instances of the positive class. Eq.(17) is applied to calculate the $Recl$.

$$Recl = \frac{TrP}{TrP + FlP} \times 100 \quad (17)$$

5.3. Classification Accuracy

Classification Accuracy (CA) is a performance metric used to evaluate the accuracy of a classification model. It measures the proportion of correctly classified instances from the total number of instances in the dataset. The Eq.(18) expresses the CA calculation.

$$CA = \left(\frac{(Number\ of\ Correct\ Predictions)}{(Total\ Number\ of\ Predictions)}\right) \times 100 \quad (18)$$

5.4. F-Measure

F-Measure (FM) is a performance metric that combines precision and recall to provide a balanced assessment of a classification model's performance. It is beneficial when dealing with imbalanced datasets. The Eq.(19) expresses the formula used for calculating FM.

Algorithm 7: HSO-LGAN

Input:
Maximum number of iterations.
Convergence threshold to determine when to stop.
Initial value for the Lagrange multiplier.
Initialization of the generator and discriminator networks.

Output:
A trained HSO-LGAN model with optimized parameters.
Trained generator and discriminator networks.

Procedure:
Initialization:
Set the maximum number of iterations to $Max_{iterations}$.
Define the convergence threshold as $Convergence_{threshold}$.
Initialize the Lagrange multiplier (λ) with an arbitrary initial value.
Initialize the generator and discriminator networks with their respective parameters.
Main Training Loop:
For iteration in range ($Max_{iterations}$):
Generate a batch of random noise vectors.
Train the Discriminator:
Sample both real data and generated data.
Update the discriminator's weights to distinguish between real and generated data.
Train the Generator:
Generate fake data and update the generator's weights to make it more realistic.
Adjust the Lagrange Multiplier (λ):
Tune λ to find the right balance between the constraint and the generator/discriminator losses.
Check for convergence based on predefined criteria. If converged, exit the loop.
End of Training:
The result is a trained $HSO - LGAN$ model with optimized parameters.
Retrieve the trained generator and discriminator networks.
Data Generation:
Using the trained generator network and random noise vectors, generate new data samples.
Ensure that the generated data adheres to the specified constraint during this process.
Evaluation and Fine-Tuning (Optional):
Assess how well the generated data satisfies the constraint.
If necessary, refine the model further to improve constraint adherence.

$$FM = \left(2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}\right) \times 100 \quad (19)$$

5.5. Matthews Correlation Coefficient

Matthews Correlation Coefficient (MCC) provides a balanced measure of a classification model's performance. It is advantageous when dealing with imbalanced datasets and is less sensitive to class imbalance than accuracy. The Eq.(20) expresses the formula used for calculating MCC.

5.6. Fowlkes-Mallows Index

The FMI is a performance metric used to evaluate the similarity between two clusters or groups, often in the context of clustering algorithms or unsupervised learning. It measures the geometric mean of precision and recall between two clusters. The Eq.(21) expresses the formula used for calculating FMI.

$$FMI = \left(\sqrt{Precision \times Recall}\right) \times 100 \quad (21)$$

6. RESULTS AND DISCUSSION

6.1. $Prec$ and $Recl$ Analysis

Figure 1 represents Prec and Recl values for different classification algorithms, namely TDAN, DELT and HSO-LGAN, and it is based on the data presented in Table 1. Figure 1 serves as a valuable visual aid for decision-makers, allowing them to quickly discern the strengths and weaknesses of these algorithms. The choice between these algorithms will ultimately depend on the specific requirements of the task. A preference for high precision, high recall, or a balance between the two will guide the selection process, considering other factors such as computational efficiency and scalability.

TDAN demonstrates a nearly equal balance between precision and recall. With a precision of approximately 52.859%, TDAN makes accurate positive predictions. At the same time, its recall of roughly 53.740% suggests that it effectively captures a substantial portion of the relevant instances. This balance could be advantageous in scenarios where both high accuracy and broad coverage of relevant data are required. DELT exhibits a notably higher precision of around 64.326% compared to its recall, approximately 65.136%. This indicates that DELT excels in making accurate positive predictions. DELT could be a promising choice in recommendation systems, where precision is paramount to avoid recommending irrelevant items to users. HSO-LGAN outperforms the other algorithms in both precision and recall. It is well-suited for tasks requiring high accuracy and comprehensive coverage with a precision of roughly 80.376% and a recall of approximately 81.538%. In sentiment-based recommendation systems, for instance, HSO-LGAN can make precise recommendations while ensuring a broader range of relevant items are included.

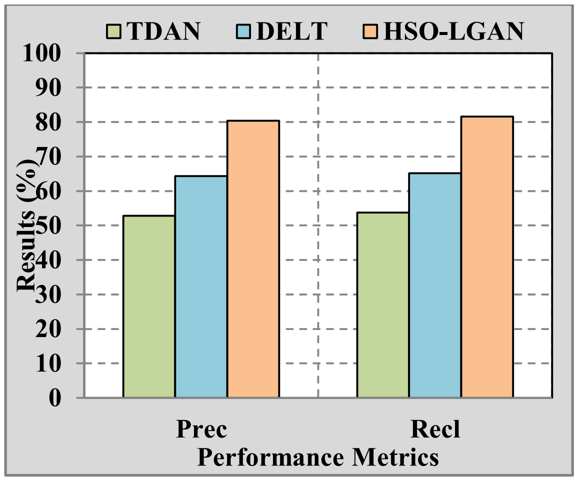

Figure 1. Prec and Recl

Table 1. Prec and Recl

Classification Algorithms	*Prec*	*Recl*
TDAN	52.859	53.740
DELT	64.326	65.136
HSO-LGAN	80.376	81.538

6.2. CA and FM Analysis

Figure 2 presents a comparative analysis of three classification algorithms, TDAN, DELT, and HSO-LGAN, based on their performance metrics. These metrics are represented as CA and FM scores in Table 2. The importance of Figure 2 lies in its ability to provide a quick overview of how these algorithms perform in classification accuracy and feature manipulation. Understanding their relative strengths and weaknesses is crucial for selecting the most suitable algorithm for a given task.

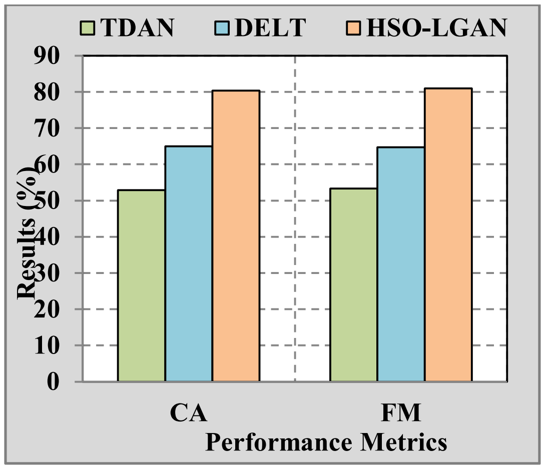

Figure 2. CA and FM

TDAN achieves a CA score of 52.833 and an FM score of 53.296. Its performance can be attributed to its unique mechanism of incorporating temporal dependencies into the classification process. TDAN likely leverages techniques such as recurrent neural networks (RNNs) or autoregressive models to capture sequential patterns in the data, making it suitable for tasks where temporal relationships are essential. DELT, with a CA score of 65.003 and an FM score of 64.728, appears to excel in feature manipulation.

Its performance can be attributed to its ability to preprocess and manipulate data effectively before the classification step. DELT may employ feature engineering, dimensionality reduction, or other data preprocessing techniques to optimize input data, leading to improved accuracy. HSO-LGAN stands out with a CA score of 80.304 and an FM score of 80.953. The exceptional performance of HSO-LGAN can be attributed to its use of Generative Adversarial Networks (GANs) or similar deep learning models. GANs can generate synthetic data samples and refine features, allowing HSO-LGAN to achieve high accuracy while effectively manipulating features.

Figure 2 provides a valuable comparative analysis of these classification algorithms, highlighting their relative strengths and capabilities. TDAN, focusing on temporal dependencies, is suitable for tasks involving time-series data. DELT excels in feature manipulation, making it an excellent choice for improving the quality of input data. Meanwhile, HSO-LGAN combines accuracy and feature manipulation prowess, indicating its versatility for various classification tasks.

The choice of algorithm should ultimately align with the specific requirements of the task at hand, considering factors such as data characteristics and desired outcomes. This comparative analysis aids in making informed decisions when selecting the most appropriate algorithm for a given application.

Table 2. CA and FM

Classification Algorithms	CA	FM
TDAN	52.833	53.296
DELT	65.003	64.728
HSO-LGAN	80.304	80.953

6.3. FMI and MCC Analysis

Figure 3 is a visual guide for comparing the performance of three distinct classification algorithms: TDAN, DELT, and HSO-LGAN. It goes beyond conventional metrics by focusing on two crucial aspects—FMI and MCC. The significance of Figure 3 lies in its ability to provide a concise yet comprehensive overview of how these algorithms fare in terms of feature manipulation and correlation capture. This understanding is essential when choosing the most suitable algorithm for specific tasks.

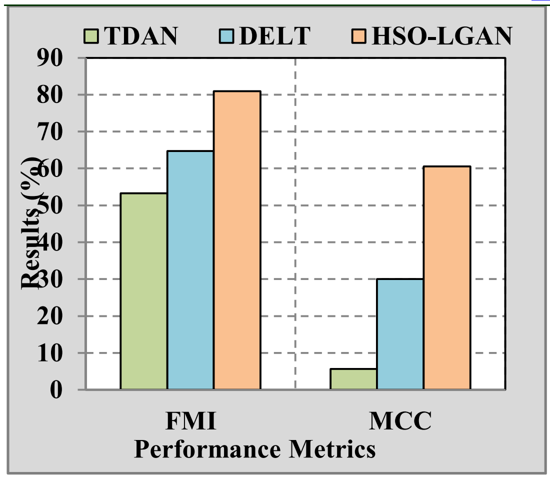

Figure 3. *FMI* and *MCC*

TDAN exhibits an FMI of 53.298 and an MCC of 5.664. These results are indicative of its underlying mechanism. TDAN is designed to excel in tasks involving temporal data. Its strength lies in its ability to recognize patterns and dependencies over time. However, its relatively lower MCC score suggests it may struggle to capture complex correlations within the data. DELT showcases impressive figures, with an FMI of 64.730 and an MCC of 30.007. DELT's mechanism is rooted in advanced feature manipulation and data preprocessing. It excels in optimizing features and capturing meaningful correlations, making it a versatile and effective choice for various classification tasks. HSO-LGAN stands out with an FMI of 80.955 and an MCC of 60.571. Its exceptional performance can be attributed to the use of GANs or similar deep-learning techniques. HSO-LGAN leverages the power of GANs to generate synthetic data, refine features, and effectively model complex correlations, resulting in outstanding scores in both FMI and MCC.

Figure 3 provides a valuable comparative analysis of TDAN, DELT, and HSO-LGAN, emphasizing their feature manipulation and correlation capture proficiency. TDAN specializes in temporal data, DELT excels in feature manipulation and correlation capture, and HSO-LGAN stands out as a versatile powerhouse. The choice of the most appropriate algorithm should align with the specific task requirements, considering factors such as the nature of the dataset and the importance of feature manipulation and correlation detection. This comparative analysis empowers decision-makers to make informed choices when selecting the algorithm best suited for a given application, ensuring optimal results.

Table 3. FMI and MCC

Classification Algorithms	FMI	MCC
TDAN	53.298	5.664
DELT	64.730	30.007
HSO-LGAN	80.955	60.571

7. CONCLUSION

Incorporating "Harmony Search Optimization-Based Generative Adversarial Networks (HSO-LGAN)" into sentiment analysis, as showcased in this research, has yielded remarkable results. The HSO-LGAN approach has demonstrated significant prowess in enhancing classification accuracy (CA) and F-Measure (FM), surpassing other classification algorithms, including TDAN and DELT, with CA values of 80.304% and FM values of 80.953%. These findings underscore the practical viability of this innovative framework in the context of customer sentiment analysis, revealing its potential to empower businesses to understand and respond to consumer feedback more effectively. The results emphasize the tangible benefits that HSO-LGAN offers, not only for businesses but also for customers. Technology equips businesses with the tools to serve and engage their clientele better. The future direction of this work entails refining HSO-LGAN for real-time implementation on dynamic online shopping platforms. Advanced natural language processing techniques and additional bio-inspired optimization methods will be explored to further enhance sentiment analysis accuracy. This ongoing effort aims to continuously innovate personalized augmented reality experiences and deepen insights into customer-brand interactions.

REFERENCES

[1] D. Babichenko, L. Grieve, E. Bilodeau, and D. Koval, "Implementation and assessment of three gamification strategies across multiple higher education disciplines," in *Proceedings of the European Conference on*

Games-based Learning, 2019, vol. 2019-Octob, pp. 41–47. doi: 10.34190/GBL.19.016.

[2] R. A. Chaves, M. J. V. Cueto, E. C. Peñalosa, and I. C. M. Moreno, "Datos y resultados sobre el sentimiento identitario en Andalucía/ Data and results on the feeling of identity in Andalusia," *Rev. Estud. Reg.*, no. 103, pp. 161–187, 2015, [Online]. Available: http://www.ujaen.debiblio.com/login?&url=https://www.proquest.com/scholarly-journals/datos-y-resultados-sobre-el-sentimiento/docview/1762038464/se-2?accountid=14555%0Ahttps://media.proquest.com/media/hms/ORIG/1/s7jwB?_a=ChgyMDIzMTEwNjIyMjE1NTMwODo1NjY3MzQ

[3] B. Omidvar-Tehrani, A. Personnaz, and S. Amer-Yahia, "Guided Text-based Item Exploration," in *International Conference on Information and Knowledge Management, Proceedings*, 2022, pp. 3410–3420. doi: 10.1145/3511808.3557141.

[4] M. P. Geetha and D. Karthika Renuka, "Improving the performance of aspect based sentiment analysis using fine-tuned Bert Base Uncased model," *Int. J. Intell. Networks*, vol. 2, pp. 64–69, 2021, doi: 10.1016/j.ijin.2021.06.005.

[5] L. Min, X. Miao, P. Bi, and F. He, "A Small amount of Labeled Data Chinese Online Course Review Target Extraction via ALBERT-IDCNN-CRF Model," in *Journal of Physics: Conference Series*, 2020, vol. 1651, no. 1. doi: 10.1088/1742-6596/1651/1/012049.

[6] R. Ghasemi, S. A. Ashrafi Asli, and S. Momtazi, "Deep Persian sentiment analysis: Cross-lingual training for low-resource languages," *J. Inf. Sci.*, vol. 48, no. 4, pp. 449–462, 2022, doi: 10.1177/0165551520962781.

[7] A. Shaddeli, F. Soleimanian Gharehchopogh, M. Masdari, and V. Solouk, "An Improved African Vulture Optimization Algorithm for Feature Selection Problems and Its Application of Sentiment Analysis on Movie Reviews," *Big Data Cogn. Comput.*, vol. 6, no. 4, 2022, doi: 10.3390/bdcc6040104.

[8] N. Lakshmidevi, M. Vamsikrishna, and S. S. Nayak, "Classification of sentiments on online products using deep learning model – RNN," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 1, pp. 7165–7172, 2019, doi: 10.35940/ijeat.A1910.109119.

[9] J. Wang and C. Li, "Research on User Satisfaction of Video Education Application Based on Reviews," in *ICEIEC 2020 - Proceedings of 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication*, 2020, pp. 340–343. doi: 10.1109/ICEIEC49280.2020.9152252.

[10] M. Moradi, M. Dass, and P. Kumar, "Differential effects of analytical versus emotional rhetorical style on review helpfulness," *J. Bus. Res.*, vol. 154, 2023, doi: 10.1016/j.jbusres.2022.113361.

[11] K. Patel *et al.*, "Facial Sentiment Analysis Using AI Techniques: State-of-the-Art, Taxonomies, and Challenges," *IEEE Access*, vol. 8, pp. 90495–90519, 2020, doi: 10.1109/ACCESS.2020.2993803.

[12] A. Senthilkumar, J. Ramkumar, M. Lingaraj, D. Jayaraj, and B. Sureshkumar, "Minimizing Energy Consumption in Vehicular Sensor Networks Using Relentless Particle Swarm Optimization Routing," *Int. J. Comput. Networks Appl.*, vol. 10, no. 2, pp. 217–230, 2023, doi: 10.22247/ijcna/2023/220737.

[13] J. Ramkumar, S. S. Dinakaran, M. Lingaraj, S. Boopalan, and B. Narasimhan, "IoT-Based Kalman Filtering and Particle Swarm Optimization for Detecting Skin Lesion," in *Lecture Notes in Electrical Engineering*, 2023, vol. 975, pp. 17–27. doi: 10.1007/978-981-19-8353-5_2.

[14] R. Vadivel and J. Ramkumar, "QoS-enabled improved cuckoo search-inspired protocol (ICSIP) for IoT-based healthcare applications," *Inc. Internet Things Healthc. Appl. Wearable Devices*, pp. 109–121, 2019, doi: 10.4018/978-1-7998-1090-2.ch006.

[15] R. Jaganathan and V. Ramasamy, "Performance modeling of bio-inspired routing protocols in Cognitive Radio Ad Hoc Network to reduce end-to-end delay," *Int. J. Intell. Eng. Syst.*, vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/IJIES2019.0228.22.

[16] J. Ramkumar and R. Vadivel, "Improved frog leap inspired protocol (IFLIP) – for routing in cognitive radio ad hoc networks (CRAHN)," *World J. Eng.*, vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.

[17] J. Ramkumar and R. Vadivel, "CSIP—cuckoo search inspired protocol for routing in cognitive radio ad hoc networks," in

Advances in Intelligent Systems and Computing, 2017, vol. 556, pp. 145–153. doi: 10.1007/978-981-10-3874-7_14.

[18] L. Mani, S. Arumugam, and R. Jaganathan, “Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol,” *ACM Int. Conf. Proceeding Ser.*, pp. 1–5, Dec. 2022, doi: 10.1145/3590837.3590907.

[19] D. Jayaraj, J. Ramkumar, M. Lingaraj, and B. Sureshkumar, “AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network,” *Int. J. Comput. Networks Appl.*, vol. 10, no. 1, pp. 119–129, Jan. 2023, doi: 10.22247/ijcna/2023/218516.

[20] J. Ramkumar, C. Kumuthini, B. Narasimhan, and S. Boopalan, “Energy Consumption Minimization in Cognitive Radio Mobile Ad-Hoc Networks using Enriched Ad-hoc On-demand Distance Vector Protocol,” *2022 Int. Conf. Adv. Comput. Technol. Appl. ICACTA 2022*, pp. 1–6, Mar. 2022, doi: 10.1109/ICACTA54488.2022.9752899.

[21] P. Menakadevi and J. Ramkumar, “Robust Optimization Based Extreme Learning Machine for Sentiment Analysis in Big Data,” *2022 Int. Conf. Adv. Comput. Technol. Appl. ICACTA 2022*, pp. 1–5, Mar. 2022, doi: 10.1109/ICACTA54488.2022.9753203.

[22] J. Ramkumar, R. Vadivel, and B. Narasimhan, “Constrained Cuckoo Search Optimization Based Protocol for Routing in Cloud Network,” *Int. J. Comput. Networks Appl.*, vol. 8, no. 6, pp. 795–803, 2021, doi: 10.22247/ijcna/2021/210727.

[23] J. Ramkumar and R. Vadivel, “Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks,” *Wirel. Pers. Commun.*, vol. 120, no. 2, pp. 887–909, Apr. 2021, doi: 10.1007/s11277-021-08495-z.

[24] J. Ramkumar and R. Vadivel, “Whale optimization routing protocol for minimizing energy consumption in cognitive radio wireless sensor network,” *Int. J. Comput. Networks Appl.*, vol. 8, no. 4, pp. 455–464, 2021, doi: 10.22247/ijcna/2021/209711.

[25] R. Jaganathan and R. Vadivel, “Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks,” *Int. J. Comput. Digit. Syst.*, vol. 10, no. 1, pp. 1063–1074, 2021, doi: 10.12785/ijcds/100196.

[26] J. Ramkumar and R. Vadivel, “Improved Wolf prey inspired protocol for routing in cognitive radio Ad Hoc networks,” *Int. J. Comput. Networks Appl.*, vol. 7, no. 5, pp. 126–136, 2020, doi: 10.22247/ijcna/2020/202977.

[27] J. Ramkumar, K. S. Jeen Marseline, and D. R. Medhunhashini, “Relentless Firefly Optimization-Based Routing Protocol (RFORP) for Securing Fintech Data in IoT-Based Ad-Hoc Networks,” *Int. J. Comput. Networks Appl.*, vol. 10, no. 4, pp. 668–687, Aug. 2023, doi: 10.22247/ijcna/2023/223319.

[28] S. J and K. U, “Sentiment analysis of amazon user reviews using a hybrid approach,” *Meas. Sensors*, vol. 27, p. 100790, 2023, doi: 10.1016/j.measen.2023.100790.

[29] M. J. Sánchez-Franco, F. J. Arenas-Márquez, and M. Alonso-Dos-Santos, “Using structural topic modelling to predict users’ sentiment towards intelligent personal agents. An application for Amazon’s echo and Google Home,” *J. Retail. Consum. Serv.*, vol. 63, p. 102658, 2021, doi: 10.1016/j.jretconser.2021.102658.

[30] P. Hajek, L. Hikkerova, and J. M. Sahut, “Fake review detection in e-Commerce platforms using aspect-based sentiment analysis,” *J. Bus. Res.*, vol. 167, p. 114143, 2023, doi: 10.1016/j.jbusres.2023.114143.

[31] J. Chen, N. Song, Y. Su, S. Zhao, and Y. Zhang, “Learning user sentiment orientation in social networks for sentiment analysis,” *Inf. Sci. (Ny).*, vol. 616, pp. 526–538, 2022, doi: 10.1016/j.ins.2022.10.135.

[32] J. Sangeetha and U. Kumaran, “A hybrid optimization algorithm using BiLSTM structure for sentiment analysis,” *Meas. Sensors*, vol. 25, 2023, doi: 10.1016/j.measen.2022.100619.

[33] P. Atandoh, F. Zhang, D. Adu-Gyamfi, P. H. Atandoh, and R. E. Nuhoho, “Integrated deep learning paradigm for document-based sentiment analysis,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 7, p. 101578, 2023, doi: 10.1016/j.jksuci.2023.101578.

[34] Y. Du, X. Jin, R. Yan, and J. Yan, “Sentiment enhanced answer generation and information fusing for product-related question answering,” *Inf. Sci. (Ny).*, vol. 627, pp. 205–219, 2023, doi: 10.1016/j.ins.2023.01.098.

[35] A. Mehbodniya, M. V. Rao, L. G. David, K. G. Joe Nige, and P. Vennam, “Online product sentiment analysis using random

evolutionary whale optimization algorithm and deep belief network," *Pattern Recognit. Lett.*, vol. 159, pp. 1–8, 2022, doi: 10.1016/j.patrec.2022.04.024.

[36] Y. Zhu, Y. Qiu, Q. Wu, F. L. Wang, and Y. Rao, "Topic Driven Adaptive Network for cross-domain sentiment classification," *Inf. Process. Manag.*, vol. 60, no. 2, 2023, doi: 10.1016/j.ipm.2022.103230.

[37] C. Choudhary, I. Singh, and M. Kumar, "SARWAS: Deep ensemble learning techniques for sentiment based recommendation system," *Expert Syst. Appl.*, vol. 216, 2023, doi: 10.1016/j.eswa.2022.119420.