International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I6P135

Volume 72 Issue 6, 397-408, June 2024

Original Article

Enhancing Sentiment Analysis in Noisy Stock Tweets Using Cognitive Particle Swarm Optimization-Based Sophisticated Deep Belief Network (CPSO-SDBN)

G. Priyadarshini[1], D. Karthika[2]

[1]*PKR Arts College for Women.*
[1]*KG College of Arts and Science.*
[2]*Department of Computer Science, VET Institute of Arts and Science (Co-education) College, Erode, TamilNadu, India.*

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

Received: 02 March 2024 Revised: 20 May 2024 Accepted: 05 June 2024 Published: 29 June 2024

Abstract *- In the fast-paced stock market, real-time information is the lifeblood of informed decision-making. Twitter, a prominent platform for disseminating news and opinions, holds a treasure trove of stock-related tweets that reflect market sentiment and trends. However, accurately classifying the sentiment expressed in these stock tweets poses a formidable challenge due to their unstructured nature and inherent noise. To tackle this issue, this research introduces a novel approach, CPSO-SDBN, which harnesses the power of Cognitive Particle Swarm Optimization (CPSO) to optimize a Sophisticated Deep Belief Network (SDBN) for sentiment analysis. CPSO-SDBN dynamically adapts to the ever-evolving and noisy stock tweet data by optimizing SDBN's architecture and hyperparameters. It achieves this by leveraging CPSO, which guides the model's configuration towards the most suitable setup for handling the complexities of stock tweet sentiment analysis. Leveraging the "Stock Tweets for Sentiment Analysis and Prediction" dataset, our research demonstrates significant improvements in sentiment analysis accuracy. These advancements empower traders, investors, and financial analysts with more precise sentiment insights, ultimately enhancing decision-making in the stock market.*

Keywords *- Stock market, Sentiment analysis, Twitter, Tweet, PSO, DBN.*

1. Introduction

Sentiment analysis, a branch of Natural Language Processing (NLP), delves into the intricate realm of human emotions encoded in text. This captivating field employs machine learning algorithms to decipher the sentiment behind words, phrases, or documents [1]. By categorizing text as positive, negative, or neutral, sentiment analysis empowers businesses and researchers to glean valuable insights from vast amounts of textual data. Whether gauging customer feedback to enhance products or monitoring social media for public opinion, sentiment analysis plays a pivotal role in decision-making. Its applications extend beyond marketing, reaching domains like healthcare for patient sentiment tracking, making it an indispensable tool in today's data-driven world [2]. Stock Tweets are concise yet impactful messages that inundate social media platforms, notably Twitter. They constitute a unique window into stock trading and investment, providing a rapid and accessible stream of information and sentiment analysis. In a mere 280 characters or less, users convey their insights, opinions, and reactions to the ever-evolving market landscape [3]. This immediacy makes Stock Tweets a valuable resource for traders and investors seeking real-time information and market sentiment. The brevity of Stock Tweets can be a double-edged sword. While they offer quick insights, they often oversimplify complex financial concepts and decisions, potentially leading to misguided actions. Investors should exercise caution and conduct thorough research when making trading decisions based on these tweets. Stock Tweets play a significant role in shaping market dynamics [4]. They can drive stock prices, influence trading strategies, and even trigger market-wide trends. For traders and investors, staying attuned to Stock Tweets is a valuable part of staying informed and adaptable in today's fast-paced financial landscape [5, 6]. Twitter and sentiment analysis share a symbiotic relationship, as the platform's real-time nature and vast user-generated content make it a goldmine for sentiment analysis applications. With its 280-character limit, Twitter encourages the concise and candid expression of thoughts, making it a rich source of unfiltered public sentiment [7]. Researchers, marketers, and organizations leverage sentiment analysis on Twitter to monitor brand reputation, track public opinion, and gain insights into trending topics. Sentiment analysis on Twitter involves utilizing Natural Language Processing (NLP) techniques and machine learning algorithms to classify tweets as positive, negative, or neutral [8]. This process can be challenging due to the brevity of tweets, slang, and sarcasm. Applications of Twitter sentiment analysis are diverse. Politicians gauge public sentiment during campaigns, businesses adjust marketing strategies based on customer feedback, and news agencies track public

reactions to breaking stories [9]. The sheer volume of tweets challenges data processing and scalability. Advanced techniques like topic modelling and sentiment lexicons help enhance accuracy. Twitter's real-time nature and user-generated content make it an ideal playground for sentiment analysis. It provides a treasure trove of data and demonstrates the practical utility of sentiment analysis in understanding public sentiment and making informed decisions in various domains [10].

Bio-inspired optimization is a computational approach that draws inspiration from the principles of natural processes such as evolution, swarm behaviour, and genetic inheritance to solve complex problems [11]. While bio-inspired optimization algorithms may not directly classify sentiments in Twitter, they offer several advantages when integrated into sentiment analysis systems.

- These algorithms can enhance feature selection and extraction processes. They can efficiently identify relevant keywords and features from vast Twitter data by mimicking biological mechanisms like genetic algorithms or particle swarm optimization. This helps sentiment analysis models focus on the most informative content [12].
- Bio-inspired optimization can aid in model parameter tuning. Just as organisms adapt to their environments, these algorithms can fine-tune the parameters of sentiment analysis models, optimizing their performance and accuracy.
- They can assist in optimizing the overall sentiment analysis workflow. For example, optimization algorithms can determine the best combination of preprocessing steps, feature engineering techniques, and model architectures to achieve optimal sentiment classification results.

Bio-inspired optimization techniques offer advantages in enhancing the efficiency and effectiveness of sentiment analysis on Twitter by improving feature selection, model parameter tuning, and workflow optimization. Integrating these principles can lead to more accurate and efficient sentiment classification in the dynamic and challenging Twitter environment.

1.1. Problem Statement

The core predicament in sentiment classification within the domain of stock tweets stems from the pervasive presence of noisy textual data. These stock-related tweets, typically characterized by their brevity, informality, and the inclusion of various forms of linguistic noise, such as misspellings, slang, and emojis, present a formidable challenge for sentiment analysis.

The brevity of tweets often condenses sentiment expression into a limited character space, leaving little room for nuanced cues. Additionally, informal language usage and unconventional expressions may introduce ambiguity, making it difficult to discern the intended sentiment polarity accurately. Misspellings and the incorporation of emojis further exacerbate the issue, adding complexity to the analysis and necessitating advanced natural language processing techniques to mitigate these challenges effectively.

1.2. Motivation

The motivation for researching sentiment classification within stock tweets lies in the critical significance of sentiment analysis in today's financial landscape. With social media, particularly Twitter, serving as a rapid and influential platform for market sentiment expression, accurate sentiment classification is pivotal for informed investment decision-making and understanding the dynamic relationship between public sentiment and stock price movements. However, the noisy and informal nature of stock tweets, characterized by misspellings, slang, and emojis, poses a significant challenge to reliable sentiment analysis. Addressing this challenge is essential for empowering investors with better insights and enhancing financial literacy, democratizing market information access, and contributing to financial markets' overall stability and transparency, making it a vital and far-reaching research endeavor.

1.3. Objectives

The overarching research objective is to enhance sentiment classification accuracy within stock tweets, addressing the distinctive challenges of noisy text data, including misspellings, slang, and emojis. The study aims to accomplish this by characterizing the extent of linguistic noise, implementing advanced data preprocessing techniques, innovating feature engineering approaches tailored to tweet brevity, developing state-of-the-art machine learning and NLP models adept at handling these linguistic nuances, rigorously evaluating model performance in the context of temporal stock tweet data, and applying the refined models to extract market sentiment insights and their correlation with stock price movements. Ultimately, this research seeks to empower investors, financial analysts, and market observers with more precise sentiment analysis tools, contributing to informed investment decisions and a deeper comprehension of the intricate dynamics between social media sentiment and financial markets.

2. Literature Review

The "Stock Market Decisions" [13] study delves into the intriguing relationship between Twitter activity and stock market decisions during two major global health crises. It explores how sentiments expressed on Twitter may influence investor behaviour, drawing comparisons between the H1N1 influenza pandemic and the COVID-19 pandemic. By examining the role of social media in shaping financial markets during pandemics, the research offers critical insights for investors and policymakers. The "Soccer Clubs Tweet Analysis" [14] comprehensive study explores the broader influence of Twitter activity on the stock prices of soccer clubs. By scrutinizing how social media engagement, sentiment, and trending topics on Twitter interact with the financial performance of soccer clubs, this research seeks to uncover the intricate relationship between

online discourse and the valuation of sports organizations. The study offers insights into the impact of digital communication platforms on the financial markets within the sports sector.

The "Stock Price Returns Twitter Effect" study [15] broadly investigates the impact of Twitter sentiment on stock price returns. It delves into how the sentiment expressed on Twitter, ranging from positive and negative opinions to trending topics and influencers' posts, can significantly influence the returns of stocks across various sectors. This research aims to provide a comprehensive understanding of the intricate relationship between social media discourse and financial markets, shedding light on how Twitter sentiment can sway investor decisions and ultimately impact stock prices. The "Kalman Filter-based Stock Market Prediction Method" [16] is a research approach that leverages the Kalman filter, a recursive mathematical tool, to enhance the efficiency of stock market prediction. This method aims to improve forecasting accuracy by dynamically updating predictions as new data becomes available, making it particularly well-suited for financial markets where data is continually evolving. By incorporating the Kalman filter into stock market prediction models, researchers seek to capture better the underlying trends and patterns in stock price movements, ultimately providing investors and financial analysts with more reliable tools for making informed investment decisions.

The "Optimized Intelligent Model" [17] is a research endeavor aimed at harnessing the power of social media sentiment analysis and artificial intelligence to predict stock market trends. This innovative approach leverages Twitter data, precisely customer opinions and sentiments, as valuable inputs for an intelligent model. The model aims to extract actionable insights from the vast sea of Twitter data through optimization techniques, identifying patterns and sentiments that can be correlated with stock price movements. By merging advanced analytical methods with real-time social media data, this research seeks to provide investors and financial analysts with a robust tool for making informed stock predictions. The "Optimal Deep Learning-Based LSTM" [18] is a research pursuit that integrates cutting-edge deep learning techniques, particularly Long Short-Term Memory (LSTM) neural networks, with Twitter sentiment analysis to enhance stock price prediction. This approach seeks to optimize the LSTM model to effectively capture intricate temporal patterns and relationships in stock price data while simultaneously considering the sentiment expressed on Twitter as valuable input. By doing so, researchers aim to improve the accuracy of stock price forecasts, enabling investors and financial analysts to make more informed decisions. The "Emotions in Twitter Communication" [19] delves into the fascinating relationship between emotions expressed in Twitter communication and companies' stock prices, particularly during the unprecedented COVID-19 pandemic. This research explores how the sentiment and emotional tone of tweets on the platform can influence investor behaviour and ultimately impact the stock market. By analyzing Twitter data within the context of the pandemic, researchers aim to uncover the dynamics through which emotions are reflected in social media discourse. The "Energy Stocks" [20] investigates the impact of Twitter sentiments on energy stocks and explores potential dependencies between companies within the energy sector. By analyzing sentiments expressed on Twitter, this research aims to determine whether social media discourse can influence energy companies' stock prices. Additionally, the study delves into the intercompany relationships within the energy sector to understand how sentiments about one company might affect others in the same industry.

The "Public Sentiment and Political Situation Analysis" [21] focuses on leveraging machine learning techniques to forecast stock market trends by considering two critical factors: public sentiment and the political situation. This study aims to harness the power of sentiment analysis on platforms like social media and news articles to gauge public sentiment towards the market and specific stocks. It evaluates the political landscape and policy decisions, which can significantly influence economic conditions. The "Social Mood - Twitter Perspective" [22] delves into the intriguing relationship between social mood and the Italian sovereign debt market, as reflected in Twitter discussions. By analyzing sentiment and trends on Twitter, this research aims to uncover how public sentiment can influence investor behaviour and impact the performance of Italian government bonds. The study offers a unique perspective on the role of social media in shaping financial markets, with a specific focus on sovereign debt, a critical component of a nation's fiscal stability.

The "Long Short-Term Memory (LSTM)" [23] framework harnesses social media sentiment analysis to improve stock market prediction. This deep learning model is known for capturing temporal dependencies in sequential data. This approach taps into sentiment dynamics that can significantly impact stock prices by systematically collecting and preprocessing social media data related to financial markets. One disadvantage of LSTM is their computational complexity and resource-intensive nature, which can pose challenges in real-time stock market prediction applications, requiring substantial computational power and potentially resulting in delayed predictions.

"Strawberry-based Bi-directional Recurrent Neural Model (SBRNM)" [17] framework harnesses customer opinions from Twitter to enhance stock prediction, employing the Strawberry-based Bi-directional Recurrent Neural Model (SBRNM) known for its capacity to capture temporal dependencies in sequential data. This approach capitalizes on real-time sentiment insights by systematically collecting and preprocessing Twitter data related to financial markets and stocks. The potential drawback is the inherent noise within Twitter data, where unrelated content, spam, or emotional expressions unrelated to financial markets may introduce inaccuracies, necessitating robust data filtering mechanisms to ensure reliable stock predictions.

3. Proposed Work

3.1. Sophisticated Deep Belief Network (DBN)

A Deep Belief Network (DBN) is a multi-layered neural network that employs unsupervised learning to learn hierarchical representations from data. It comprises multiple layers containing hidden units that progressively capture higher-level abstractions. Here is an in-depth breakdown of the training process.

3.1.1. Initialization

Initialize weight matrices $W^{(f)}$ and bias vectors $b^{(l)}$ for each layer $H^{(l)}$.These initializations can be performed randomly or using techniques like Gaussian initialization to facilitate convergence.

3.1.2. Layer-Wise Training of RBMs

Layer-wise training of Restricted Boltzmann Machines (RBMs) is crucial in building deep neural networks, particularly in unsupervised pre-training for deep learning models. This process involves training RBMs one layer at a time, starting from the visible layer (V) and moving towards the hidden layers (H).The details of training First RBM $\left(V \rightarrow H^{(1)}\right)$ is discussed below.

Positive Phase

The probabilities of hidden units being active given visible units' values are calculated using the sigmoid activation function and expressed in Equation (1).

$$p\left(h_j^{(1)} = 1|V\right) = \sigma\left(\sum_i W_{ij}^{(1)} V_i + b_j^{(1)}\right) \quad (1)$$

Sample Hidden Units

Sample the hidden units' binary activations based on the probabilities calculated in the positive phase, i.e. using Equation (1).

Negative Phase - Reconstruction in the Visible Layer

The probabilities of active, visible units in the sampled hidden units are estimated using Equation(2).

$$p\left(v_i = 1|H^{(1)}\right) = \sigma\left(\sum_j W_{ij}^{(1)} h_j^{(1)} + a_i\right) \quad (2)$$

Sample Visible Units

To sample the binary activations of visible units based on the probabilities estimated during the reconstruction phase. Given the sampled hidden unit activations, obtain the probabilities of the visible units being active (having a value of 1). Denote these probabilities as$P\left(v_i = 1\middle|h_i^{(1)}\right)$, where v_i represents the i-th visible unit and $h_i^{(1)}$represents the corresponding hidden unit.

To obtain binary activations for the visible units, perform stochastic binary sampling as follows for each visible unit v_i:

- Generate a random number between 0 and 1 using a uniform random number generator.
- Compare the generated random number to the probability $P\left(v_i = 1\middle|h_i^{(1)}\right)$.
- If the random number is less than $P\left(v_i = 1\middle|h_i^{(1)}\right)$, set $v_i = 1$; otherwise set $v_i = 0$.

By repeating this sampling process for each visible unit in the RBM, stochastic sampling introduces randomness into the binary activations of the visible units, a crucial aspect of the learning process in RBMs. It enables the RBM to explore various possible states of the visible units and gradually refine its representation of the data distribution. This iterative sampling process continues during the training of the RBM until convergence criteria are met or until a specified number of training epochs is reached.

Negative Phase - Hidden Layer

The theoretical computation of the probabilities of hidden units being active $P\left(v_i = 1\middle|h_i^{(1)}\right)$ given to the sampled visible units in RBM involves applying the principles of energy-based models and the sigmoid activation function. In RBM, the energy associated with a particular configuration of visible and hidden units is defined in Equation (3).

$$E(v,h) = -\sum_i \sum_j W_{ij} v_i h_j - \sum_j b_j h_j - \sum_i c_i v_i \quad (3)$$

Where v_i represents the i-th visible unit, h_j represents the j-th hidden unit in the first hidden layer, W_{ij} is the weight connecting the visible unit v_i to hidden unit h_j in the first hidden layer, b_j is the bias associated with hidden unit h_j layer, c_i is the bias associated with visible unit v_i.

Using Equation (4), the probabilities of active hidden units with sampled visible units are calculated.

$$p\left(h_j^{(1)} = 1|v\right) = \sigma\left(\sum_i W_{ij}^{(1)} + b_j^{(1)}\right) \quad (4)$$

Update Parameters

The weights and biases are updated based on the differences between the positive and negative phase probabilities using a learning rate (ϵ) and the same is expressed in Equation (5).

$$\begin{aligned} \Delta W_{ij}^{(1)} &= \epsilon\left(p\left(h_j^{(1)} = 1|V\right) V_i - p\left(h_j^{(1)} = 1|v\right) v_i\right) \\ \Delta b_j^{(1)} &= \epsilon\left(p\left(h_j^{(1)} = 1|V\right) - p\left(h_j^{(1)} = 1|v\right)\right) \\ \Delta a_i &= \epsilon\left(V_i - p\left(v_i = 1|H^{(1)}\right)\right) \end{aligned} \quad (5)$$

Training Subsequent RBMs

The exact process is repeated till it propagates the activations from the previous hidden layer $H^{(l-1)}$to the current hidden layer $H^{(l)}$, and it is expressed in Equation (6).

$$H^{(l-1)} \rightarrow H^{(l)} \quad (6)$$

3.1.3. Fine-Tuning

This research fine-tunes the so-far obtained solution using Algorithm 1.

Algorithm 1. Fine Tuning

Step 1: Stack the trained RBMs to form the DBN, using the activations of one RBM as input to the next.

Step 2: Calculate the gradients of a task-specific loss function (L)Concerning the DBN's parameters using labelled data:

$$\frac{\partial L}{\partial W^{(l)}}, \frac{\partial L}{\partial b^{(l)}}, \frac{\partial L}{\partial a}$$

Step 3: Update the parameters using a gradient descent algorithm, adjusting the weights and biases based on the calculated gradients and a learning rate (α):

$$W^{(l)} \leftarrow W^{(l)} - \alpha \frac{\partial L}{\partial W^{(l)}}$$

$$b^{(l)} \leftarrow b^{(l)} - \alpha \frac{\partial L}{\partial b^{(l)}}$$

$$a \leftarrow a - \alpha \frac{\partial L}{\partial a}$$

The overall SDBN process is provided in Algorithm 2.

Algorithm 2: SDBN

Step 1: Initialize each SDBN layer's weight matrices (W) and bias vectors (b).

Step 2: For each layer:

- **a.** Calculate probabilities of hidden units given visible units (Positive Phase).
- **b.** Sample hidden unit activations.
- **c.** Estimate visible unit probabilities given sampled hidden units (Negative Phase - Reconstruction).
- **d.** Sample visible unit activations.
- **e.** Calculate probabilities of active hidden units with sampled visible units.
- **f.** Update weights and biases using the differences between positive and negative phases.

Step 3: Train subsequent RBMs propagating activations from previous layers to current layers.

Step 4: Fine-tuning:

- **a.** Stack RBMs to form the DBN.
- **b.** Calculate gradients of a task-specific loss function using labelled data.
- **c.** Update parameters (W, b, a) using gradient descent.

The very few demerits of Sophisticated Deep Belief Networks (SDBNs) include:

- Sensitivity to Hyperparameters: High classification accuracy with SDBNs often depends on fine-tuning hyperparameters, which can be challenging and time-consuming.
- Choice of Architecture Impact: Inadequate or overly complex SDBN architectures can result in suboptimal classification accuracy.
- Long Training Times: Training deep networks like SDBNs can be computationally intensive and time-consuming, especially for large datasets, potentially impacting practicality.
- Risk of Local Minima: SDBNs can get stuck in local minima during training, leading to suboptimal solutions and reduced classification accuracy

To overcome the above demerits of SDBN there exists a need for utilizing bio-inspired optimization techniques. These optimization approaches draw inspiration from biological processes, such as genetic algorithms, particle swarm optimization, or simulated annealing, to fine-tune the SDBN's parameters and architecture. By harnessing the power of bio-inspired optimization, researchers and practitioners can mitigate the limitations of DBNs, enhance their classification accuracy, and make them more accessible and adaptable to a wide range of applications.

3.2. Cognitive Particle Swarm Optimization

In 1995, the pioneering work of Dr. Kennedy and Professor Eberhart introduced a transformative optimization technique known as Particle Swarm Optimization (PSO) [24]. This ingenious method presents a versatile and readily implementable approach, functioning as a random heuristic that elegantly taps into the power of built-in parallelism. Cognitive Particle Swarm Optimization (CPSO) inspired by PSO. CPSO stands out for its remarkable prowess in optimizing complex, non-differentiable, non-continuous, and rapidly converging nonlinear functions. It has since become an indispensable tool in the domain of computational intelligence. At the heart of the CPSO algorithm lies a swarm of 'particles,' each representing a potential solution to the optimization problem. This dynamic assembly of particles collectively constitutes what is referred to as the 'population,' encompassing every conceivable solution within its ranks. These particles traverse the problem's solution space, making incremental adjustments guided by their historical experiences and shared knowledge. Their ultimate goal is to converge toward the most optimal solution achievable collaboratively.

The issues present in optimization when applied in the stock marketing domain are represented in Equation (7).

$$min\ g(P)e.f.P\ \omega\ E\ \subseteq\ B^Y \quad (7)$$

Where $P = (P_1, P_2, \ldots, P_T)$ represents a matrix of decision variables consisting of T vectors, each with m dimensions, within the feasible space E. Equation (7) represents the focal point of CPSO's mission. It involves a matrix of decision variables, denoted as $P = (P_1, P_2, \ldots, P_T)$, where each P_i is a vector comprising m dimensions. This matrix operates within the confines of a feasible space denoted as E, ensuring that solutions remain within acceptable boundaries. The problem statement encapsulates the essence of the optimization challenge: finding the set of variables P that minimizes the objective

function $g(P)$ while adhering to the constraints defined by E. This encapsulates the CPSO methodology's fundamental essence, which seeks to optimize and navigate the intricate landscape of nonlinear functions with remarkable efficiency.

3.2.1. Random Initialization

In the traditional PSO algorithm, the process of initializing the positions of particles in the initial population is a crucial starting point. It involves generating random values for each dimension of an individual particle, ensuring that each particle begins its optimization journey from a unique and diverse starting point.

This step is instrumental in promoting exploration across the solution space. This random initialization process ensures diversity in the starting positions of particles, fostering exploration and enabling the algorithm to potentially discover a wide range of solutions within the problem space.

For each particle denoted as P_s, belonging to the initial population ($s = 1,2, \dots, T$), and for each dimension ($w = 1,2, \dots, Y$), the position P_{sw} is computed using the following Equation(8):

$$P_{sw} = Low_w + rand(0,1) * (Om_w - Low_w), \quad s = 1,2, \dots, T, \quad w = 1,2, \dots, Y \tag{8}$$

Where T represents the total population size in a specific region, while Y signifies the average dimensionality of these particles. The bounds for each dimension, indicated by Om_w (i.e., upper bound) and Low_w (i.e., lower bound), are provided to restrict the possible values each dimension can take.

Using$rand(0,1)$ generates a uniformly distributed random number within the range [0, 1], adding a stochastic element to the particle positions' initialization.

3.2.2. Gaussian-based Particle Position and Velocity Update

In CPSO, the movement of particles through the solution space can be conceptually enriched by incorporating Gaussian principles, which introduce probabilistic elements into the particle dynamics. This addition provides a more nuanced understanding of how particles explore and exploit the solution landscape.

Gaussian-based Velocity Update

A Gaussian-based Velocity Update refers to modifying or adapting the standard velocity update mechanism in optimization algorithms, such as CPSO, that incorporates Gaussian principles and probabilistic concepts. Equation (10) mathematically expresses the Gaussian-based Position Update.

$$R_s^{(f+1)} = \pi R_s^{(f)} + u_1 . b_1 . \left(\hat{M}_s^{(f)} - P_s^{(f)}\right) + u_2 . b_2 . \left(\hat{J}_s^{(f)} - P_s^{(f)}\right) \tag{9}$$

This research can consider the Gaussian influence on the velocity update through Equation (9).

- $\pi R_s^{(f)}$ represents the particle's current velocity at iteration f, adjusted by an inertial weight π. In a Gaussian context, we can interpret π as controlling the standard deviation of a Gaussian distribution that influences the particle's velocity. Higher π values lead to a broader distribution, encouraging more exploratory movements, while lower π values result in a narrower distribution, promoting the exploitation of promising regions.
- u_1 and u_2 acting as individual learning factors can be seen as parameters influencing the mean shift of Gaussian distributions. These factors determine how strongly a particle's personal best position $\hat{M}_s^{(f)}$ and the global best position $\hat{J}_s^{(f)}$ affect the mean velocity update. Higher values of u_1 and u_2 lead to more substantial shifts in the mean, emphasizing the influence of these positions.
- b_1 and b_2, introduced as random integers, contribute to the stochasticity of the particle's velocity update. We can regard these random variables as perturbations with Gaussian noise in a Gaussian context, influencing the velocity with minor, random variations following a Gaussian distribution. This stochasticity ensures diversity in particle movement and adaptability to unforeseen changes in the solution landscape.

Gaussian-based Position Update

A Gaussian-based Position Update refers to a modification or adaptation of the standard position update mechanism in optimization algorithms, such as Cognitive Particle Swarm Optimization (CPSO), that incorporates Gaussian principles and probabilistic concepts into the computation of new particle positions. Equation (11) mathematically expresses the Gaussian-based Position Update.

$$P_s^{(f+1)} = P_s^{(f)} + R_s^{(f+1)} \tag{11}$$

Regarding the position update, the Gaussian concept can be extended to accumulating velocity updates. i.e., The position update $P_s^{(f+1)}$ represents the accumulation of velocity changes akin to integrating the stochastic increments. In a Gaussian context, this accumulation can be thought of as tracking the particle's position as it meanders through the solution space, subject to the Gaussian-influenced velocity adjustments. By incorporating Gaussian-inspired concepts, CPSO takes on a probabilistic character, acknowledging the uncertainty and exploration-exploitation balance inherent in optimization problems. This enriched perspective enhances our understanding of how CPSO particles navigate the complex solutions landscape.

3.2.3. Gaussian Constraints in Particle Position and Velocity

Incorporating Gaussian constraints within CPSO provides a probabilistic element to the position and velocity of particles. This Gaussian-based approach introduces randomness following Gaussian distributions to enrich the

exploration and exploitation processes.

Gaussian-Enhanced Particle Position Constraints

Each P_{sy} is now subject to Gaussian constraints within the range $[-P_{max}, P_{max}]$. This means that instead of strictly bounding the positions to a fixed interval, the positions follow a Gaussian distribution within this range. The Gaussian distribution allows for more diverse positioning around the mean, enhancing the exploration capacity of each particle along each dimension. Equation (12) describes how to update particle positions with Gaussian influences:

$$P_{sy} = P_{sy} + \varepsilon \times Gaussian(0, \sigma_{position}) \quad (12)$$

Where P_{sy} is the updated position along dimension y.ε is a random value sampled from a standard Gaussian distribution (mean=0, standard deviation = 1).$Gaussian(0, \sigma_{position})$ generates random values from a Gaussian distribution with a mean of 0 and a standard deviation of $\sigma_{position}$, which controls the spread of the Gaussian influence. Adjusting $\sigma_{position}$ allows you to control how much particles' positions deviate from their current values within the constraints

Gaussian-Enhanced Particle Velocity Constraints

R_{sy} the velocity of the sth particle is influenced by Gaussian constraints. The velocity components $(R_{s1}, R_{s2}, \dots R_{sY})$ now follow Gaussian distributions within the specified range $[-R_{max}, R_{max}]$.

This Gaussian influence allows for random but controlled variations in velocity, providing adaptability to the particles while preventing them from exceeding the maximum allowable speed. The following equation describes how to update particle velocities with Gaussian influences:

$$R_{sy} = \prod (R_{sy} + \varepsilon \times Gaussian(0, \sigma_{position})) \quad (13)$$

Where R_{sy} is the updated velocity component along dimension y.

This Gaussian-based constraint into particle positions and velocities in the CPSO algorithm embraces a probabilistic approach. This adds an element of uncertainty to the movement and positioning of particles, aiding exploration by allowing for a more diverse search across the solution space. The Gaussian influence ensures a balanced exploration-exploitation trade-off, resulting in a more robust optimization process.

3.2.4. Local Best Solution

Equation (14) represents the optimal position achieved by the sth particle in the CPSO swarm up to the current iteration. This vector contains values for each dimension (1 to Y), indicating where the particle has previously found its most promising solutions within the search space. It is the particle's personal best position encountered during its optimization journey.

$$\widehat{M}_s = \left(\widehat{M}_{s1}, \widehat{M}_{s2}, \dots, \widehat{M}_{sY}\right)^F \quad (14)$$

Equation(15) represents the velocity update for the sth particle at iteration $(f+1)$ by incorporating Gaussian-based principles from earlier responses.

$$R_s^{(f+1)} = R_s^{(f)} + u_1 \times b_1 \times \left(\widehat{M}_s^{(f)} - P_s^{(f)}\right) + u_2 \times b_2 \times \left(\hat{J}_s^{(f)} - P_s^{(f)}\right) \quad (15)$$

Where $\widehat{M}_s^{(f)}$ indicates the personal best (best-known position) of the sth particle up to the current iteration, $\hat{J}_s^{(f)}$ indicates the global best (best-known position across the entire swarm) up to the current iteration and $P_s^{(f)}$ represents the current position of the sth particle.

3.2.5. Global Best Solution

Equation (16) represents the most significant or best-known position across the entire CPSO swarm up to the current iteration. It is essentially the global best position any particle within the swarm finds. This vector contains values for each dimension (1 to Y), signifying the swarm's consensus on the most promising solution in the search space at any given point in the optimization process.

$$\hat{J}_s = \left(\hat{J}_{s1}, \hat{J}_{s2}, \dots\dots, \hat{J}_{sY}\right)^F \quad (16)$$

The position update for the sth particle at iteration (f+1) is carried out using Equation(17).

$$P_s^{(f+1)} = P_s^{(f)} + R_s^{(f+1)} \quad (17)$$

Where $P_s^{(f)}$represents the current position of the sth particle at iteration (f).$R_s^{(f+1)}$is the updated velocity of the sth particle at iteration $(f+1)$, considering Gaussian-based influences.

In the context of CPSO, these two vectors play a critical role in guiding the movement and convergence of particles. The personal best $\widehat{M}_s$ serves as an individual reference point for each particle, helping it navigate toward previously successful regions in the search space. The global best $\hat{J}_s$ guides the entire swarm toward the most promising solution found collectively by all particles. The interplay between personal and global bests enables CPSOs to efficiently explore and exploit the search space, ultimately converging toward high-quality solutions.

3.2.6. Selection and Update of Best Particles

In the CPSO algorithm, a crucial aspect of refining the current population and enhancing the search for optimal solutions involves the selection and updating of individual particle bests and the global best. These updates ensure that particles continually strive toward better solutions. The process is guided by the minimization of the objective function $g(P)$.

$$\begin{cases} M_s^* = g\left(\widehat{M}_s^{(f)}\right) \text{ and } \widehat{M}_s^{(f+1)} = \widehat{M}_s^{(f)}, \ \text{if } g\left(P_s^{(f+1)}\right) \geq g\left(\widehat{M}_s^{(f)}\right) \\ M_s^* = g\left(P_s^{(f+1)}\right) \text{ and } \widehat{M}_s^{(f+1)} = P_s^{(f+1)}, \ \text{if } g\left(P_s^{(f+1)}\right) < g\left(\widehat{M}_s^{(f)}\right) \end{cases} \quad (18)$$

If the fitness value of the trial individual, $g\left(P_s^{(f+1)}\right)$, is equal to or greater than the fitness value of the retained individual's personal best, $g\left(\widehat{M}_s^{(f)}\right)$, then the retained personal best remains unchanged, and $\widehat{M}_s^{(f+1)}$is set equal

to$\widehat{M}_s^{(f+1)}$. This signifies that the particle has not found a better solution and retains its current best position.

However, if the fitness value of the trial individual is lower than the retained personal best, $\left(P_s^{(f+1)}\right) < g\left(\widehat{M}_s^{(f)}\right)$, then the retained personal best, M_s^*, is updated to the fitness value of the trial individual, $g\left(P_s^{(f+1)}\right) \geq g\left(\widehat{M}_s^{(f)}\right)$ is set equal to the trial individual's position, $P_s^{(f+1)}$. This indicates that the particle has discovered a superior solution and updates its best position accordingly.

$$\hat{J}^{(f+1)} = P_a\omega\left\{\widehat{M}_1^{(f+1)}, \widehat{M}_2^{(f+1)}, \ldots, \widehat{M}_T^{(f+1)}\right\} \quad (19)$$

In Equation(19), the global best position, $\hat{J}^{(f+1)}$, is updated. This is achieved by considering the positions of all particles in the population at iteration $(f+1)$, denoted as $\widehat{M}_1^{(f+1)}, \widehat{M}_2^{(f+1)}, \ldots, \widehat{M}_T^{(f+1)}$. The selected global best position represents a combination of the best positions from all particles.

$$J^{*(f+1)} = g\left(\hat{J}^{(f+1)}\right) \quad (20)$$

Algorithm 3: CPSO

Step 1: Initialization
- Initialize a population of particles within the feasible space E.
- Set each particle's best position. $M_i$to its initial position P_i.
- Initialize the global best position (J) based on the best positions from all particles.
- Set the global best fitness value (J^*) to the fitness value of the global best position.

Step 2: Random Initialization
- For each particle, Initialize its position in each dimension with Gaussian-based influences.

Step 3: Gaussian-based Velocity and Position Update
- For each particle, update the particle's velocity and position using Gaussian-based principles.

Step 4: Gaussian Constraints
- Apply Gaussian constraints to particle positions and velocities for adaptability.

Step 5: Selection and Update of Best Particles
- For each particle:
  - ✓ If the trial position has a better fitness value than the personal best, update the personal best.
  - ✓ Update the global best position based on the best positions from all particles.
  - ✓ Calculate the global best fitness value.

Step 6: Termination Condition
- Repeat the above steps for a specified number of iterations or until convergence.

In Equation(20), the global best fitness value, $J^{*(f+1)}$, is computed by evaluating the objective function g(.) on the global best position $J^{*(f+1)}$.

This fitness value represents the overall best solution found by the entire population at iteration $(f+1)$. Equation(18) to Equation(20) collectively govern the updating of particle bests and the global best, driving the CPSO algorithm to iteratively explore and converge towards optimal solutions in the minimization problem.

The global best represents the best solution found by the entire swarm of particles, while the particle best stores the best solutions each particle has encountered individually.

4. About Dataset

The "Stock Tweets for Sentiment Analysis and Prediction" dataset is valuable for researchers and analysts interested in understanding the relationship between social media sentiment and stock market trends. This dataset comprises over 80,000 tweets collected from the top 25 most closely monitored stock tickers on Yahoo Finance from September 30, 2021, to September 30, 2022.

What sets this dataset apart is that it includes stock market price and volume data for the corresponding dates and stocks, providing a comprehensive view of market dynamics. Each entry in the dataset contains essential information, including the date and time of the tweet, the full text of the tweet, the stock ticker name, and the corresponding company name. Researchers can use this data for various purposes:

- Sentiment Analysis: Analyzing the sentiment of tweets related to specific stocks can help gauge public sentiment and investor sentiment over time. This information is invaluable for understanding market sentiment fluctuations.

- Stock Price Prediction: Analysts can develop predictive models to anticipate stock price movements by combining sentiment and historical stock market data. This can be a powerful tool for investors and traders.

- Exploring Sentiment-Price Connection: Researchers can delve deeper into the intricate relationship between public sentiment expressed on social media and the subsequent impact on stock prices. This exploration can lead to insights into market behavior.

The dataset's inspiration from existing sentiment analysis lexicons and stock market sentiment datasets ensures its relevance and potential for furthering our

understanding of the interplay between social media, sentiment, and financial markets.

Researchers and data scientists can leverage this dataset to uncover new insights and develop innovative stock market analysis and prediction strategies.

Algorithm 4: CPSO-SDBN

Step 1: **Initialization:**
- Initialize a population of CPSO particles, each representing an SDBN configuration.
- Parameters for SDBN architecture, such as the number of layers, hidden units, learning rates, and others, are encoded in the particles.

Step 2: **Optimization with CPSO:**

For each CPSO iteration:

a. Evaluate the fitness of each particle (SDBN configuration) using a fitness function that measures SDBN performance on a validation dataset.

b. Update each particle's personal best position (SDBN configuration) if it achieves better fitness.

c. Update the global best position (SDBN configuration) if any particle achieves better fitness.

d. Update particle positions (SDBN configurations) using CPSO velocity and position update equations.

Step 3: **Training DBN Models:**
- After CPSO optimization converges (or after a set number of iterations), select the best-performing DBN configuration based on the global best position.

Step 4: **Train a DBN model using the selected DBN configuration on the target task**

a. Initialize DBN architecture and hyperparameters based on the selected configuration.

b. Pretrain the DBN layers using unsupervised learning methods like Restricted Boltzmann Machines (RBMs) or Contrastive Divergence (CD).

c. Fine-tune the DBN using supervised learning with labeled data.

d. Optionally, repeat steps b and c for multiple DBN configurations if an ensemble of DBNs is desired.

Step 5: **Final Model Selection**:
- Select the final trained DBN model based on its performance on a separate test dataset.

Step 6: **Termination Condition:**
- Repeat the above steps for a specified number of CPSO iterations or until convergence criteria are met.

4.1. Fusion of CPSO and SDBN

Cognitive Particle Swarm Optimization-Deep Belief Network (CPSO-DBN) is a powerful hybrid approach that combines two distinct artificial intelligence techniques: Cognitive Particle Swarm Optimization (CPSO) and Deep Belief Networks (DBN). CPSO-DBN is designed to address complex optimization problems and leverage deep learning capabilities for feature learning, classification, and data representation tasks. CPSO-DBN represents the fusion of CPSO's optimization capabilities with the deep learning power of DBNs. In CPSO-DBN, CPSO is used to optimize the hyperparameters and architecture of DBN models. This optimization process aims to find the best DBN configuration that maximizes performance on a given task. CPSO-DBN automates the often complex and time-consuming task of hyperparameter tuning for DBNs, making it more accessible and efficient. Algorithm 4 provides the overall process involved in CPSO-SDBN.

5. Results and Discussion

5.1. Precision Analysis

Precision is a crucial evaluation measure that assesses the accuracy of positive predictions made by these models. It represents the ratio of true positive predictions to the total positive predictions made. In Figure 1, we observe varying levels of precision across the three models, which can be attributed to differences in their underlying working mechanisms.LSTM, a widely used recurrent neural network architecture, exhibits a precision of 47.77%. This value suggests that LSTM is moderately effective at correctly classifying positive instances while avoiding false positives. Its performance is primarily based on its ability to capture sequential dependencies in the data. SBRNM showcases a higher precision of 60.13%. This suggests that SBRNM is more adept at distinguishing relevant information from noise in the dataset. Bi-directional processing, which allows the model to consider past and future context, likely improves precision. CPSO-SDBN stands out with the highest precision of 82.07%. This indicates that CPSO-SDBN excels at minimizing false positive errors and accurately identifying positive instances. Its use of cognitive particle swarm optimization likely enhances its ability to fine-tune the model's parameters for precision-oriented tasks.

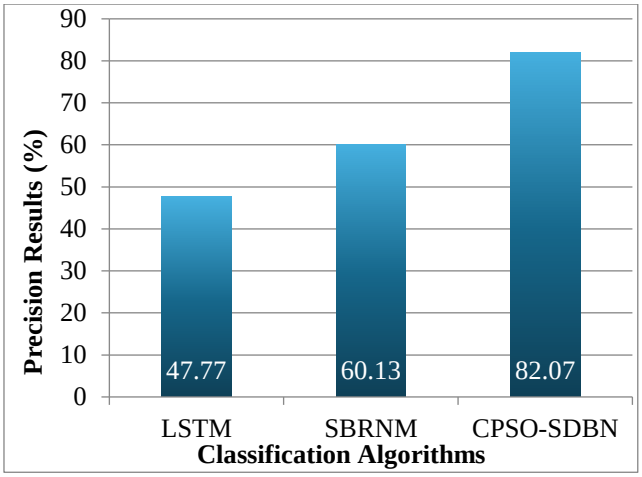

Fig. 1 Precision

Figure 1 illustrates the precision performance of different models, highlighting their varying abilities to provide accurate positive predictions. While LSTM, SBRNM, and CPSO-SDBN contribute to predictive precision, their diverse working mechanisms result in distinct performance levels in this critical metric. Precision is a pivotal consideration when selecting a model for tasks where minimizing false positives is crucial, such as medical diagnosis or anomaly detection.

5.2. Recall Analysis

Figure 2 presents the recall performance metric for diverse models, including LSTM, SBRNM, and CPSO-SDBN. Figure 2 depicts Recall as a critical evaluation measure to assess a model's ability to correctly identify and capture all relevant positive instances within a dataset. Recall is significant in scenarios where the cost of missing positive instances is high, such as medical diagnosis or fault detection, as it quantifies the model's effectiveness in minimizing false negatives. A high recall value indicates the model proficiently captures most positive cases in the data. LSTM demonstrates a recall value of 43.72%. The underlying mechanism of LSTM, which allows it to maintain long-term dependencies in sequences, contributes to its ability to capture certain positive instances.

However, its Recall might be limited due to potential challenges in capturing subtle or rare positive patterns. SBRNM exhibits a higher recall of 60.49%. This indicates that SBRNM is more effective at capturing relevant positive instances, especially those within complex and contextually rich data. The bidirectional processing employed by SBRNM likely aids in this capability. CPSO-SDBN presents a recall value of 81.72%, the highest among the models. This suggests that CPSO-SDBN excels in capturing the most actual positive instances while minimizing false negatives. The cognitive particle swarm optimization mechanism likely enhances the model's ability to fine-tune its parameters for recall-oriented tasks. It is particularly suitable for applications where missing positive cases is a critical concern.

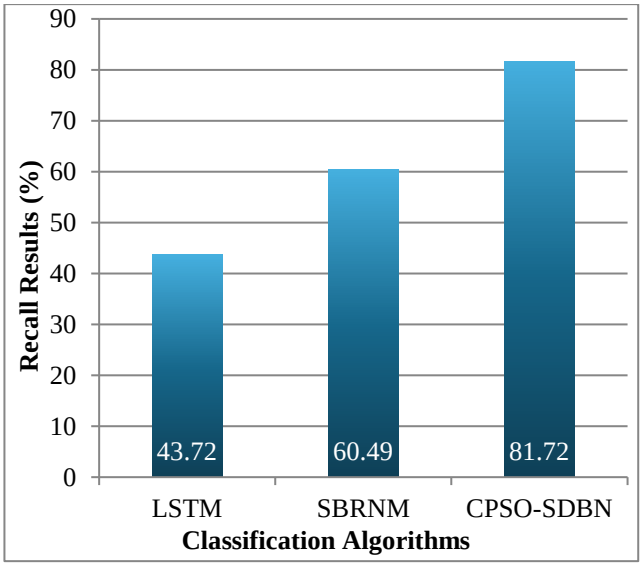

Fig. 2 Recall

Figure 2 illustrates the recall performance of different models, emphasizing their varying abilities to correctly identify and capture positive instances within the dataset. While LSTM, SBRNM, and CPSO-SDBN contribute to Recall, their distinct working mechanisms result in differing performance levels in this important metric. The Recall is a fundamental consideration when choosing a model for tasks that prioritize the detection of positive instances to minimize potential adverse consequences of false negatives.

5.3. Classification Accuracy Analysis

Figure 3 presents the classification accuracy performance metric for a selection of models, including LSTM, SBRNM, and CPSO-SDBN. Classification accuracy is widely used in machine learning and data analysis to assess the model's ability to correctly classify positive and negative instances. It represents the ratio of correctly predicted instances to the total number of instances in the dataset, providing an overall measure of a model's performance.

LSTM is an example of a model that utilizes recurrent neural network (RNN) architecture. It maintains a memory state to capture sequential dependencies in data, making it particularly effective in tasks where order and context are crucial, such as natural language processing.

The moderate accuracy of LSTM (44.83%) in Figure 3 suggests that while it excels at handling sequential information, it may face challenges in achieving high accuracy across various classes when dealing with complex or multi-modal data. SBRNM stands out with an accuracy of 61.33%.

This model incorporates bidirectional processing, enabling it to consider past and future contexts when making predictions. This feature is especially advantageous for tasks involving complex and context-rich data, which may explain its higher accuracy than LSTM. The bidirectional mechanism allows SBRNM to capture more nuanced patterns in the data.

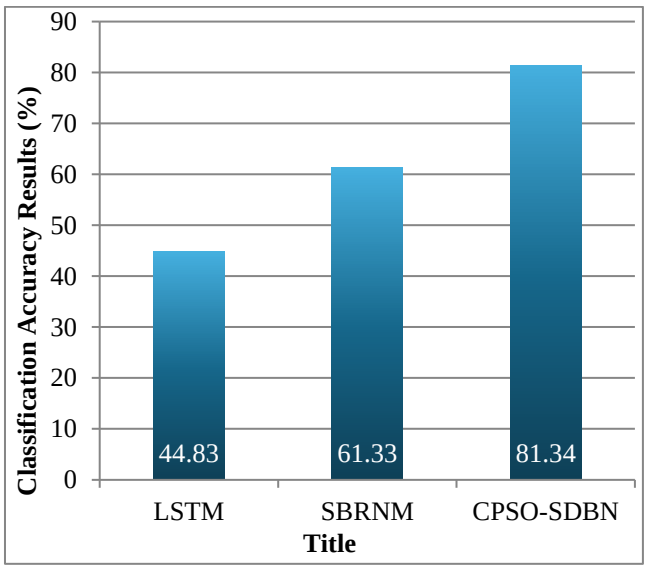

Fig. 3 Classification accuracy

CPSO-SDBN attains an impressive accuracy of 81.34%. Its underlying mechanism involves cognitive particle swarm optimization, which aids in optimizing the model's parameters for superior accuracy across various classes. This optimization approach likely contributes to CPSO-SDBN's ability to provide highly accurate predictions across multiple classes, making it particularly suitable for applications where overall accuracy is paramount.

Figure 3 showcases the classification accuracy of different models, and their varying abilities are attributed to their unique working mechanisms. LSTM excels in handling sequential data, SBRNM leverages bidirectional processing for nuanced context, and CPSO-SDBN employs cognitive particle swarm optimization for optimizing parameters and achieving high overall accuracy. The choice of model should be made considering the specific requirements of the task and the nature of the data being analyzed.

5.4. F-Measure Analysis

Figure 4 illustrates the F-Measure performance metric for models, including LSTM, SBRNM, and CPSO-SDBN. As shown in this figure, the F-Measure is a composite metric that combines precision and Recall, providing a balanced assessment of a model's performance in tasks requiring a trade-off between these two aspects.

The F-Measure is particularly useful when there is a need to balance minimizing false positives (precision) and false negatives (Recall). It is calculated as the harmonic mean of precision and Recall, allowing it to capture the model's ability to provide accurate positive predictions while not missing relevant positive instances.

LSTM achieves an F-measure of 45.66% in Figure 4. The LSTM's mechanism enables it to capture sequential dependencies in data, contributing to its ability to strike a balance between precision and Recall. This makes it a suitable choice for tasks where maintaining the balance between false positives and false negatives is essential. SBRN exhibits an F-Measure of 60.31%, indicating its effectiveness in balancing precision and Recall.

The bidirectional processing mechanism incorporated into SBRNM allows it to consider past and future contexts, enhancing its ability to capture relevant data patterns without compromising precision. CPSO-SDBN demonstrates an F-Measure of 81.90%, the highest among the models. This underscores CPSO-SDBN's ability to balance precision and Recall strongly. The cognitive particle swarm optimization mechanism is crucial in optimizing the model's parameters to achieve this balance. It is a valuable choice for tasks where a harmonious trade-off is vital.

Figure 4 presents the F-Measure performance of different models, emphasizing their varying abilities to strike a balance between precision and Recall.

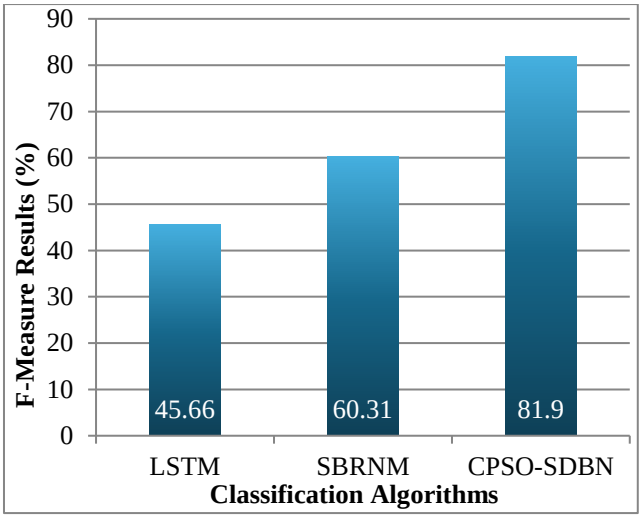

Fig. 4 F-Measure

While LSTM, SBRNM, and CPSO-SDBN contribute to the F -Measure, their distinct working mechanisms result in different performance levels in achieving this crucial balance. The F-Measure is especially valuable in scenarios where it is important to find a compromise between minimizing false positives and false negatives, such as in medical diagnostics or information retrieval systems.

6. Conclusion

Real-time information is indispensable in the realm of stock markets, where timely and informed decisions can significantly impact financial outcomes. Social media platforms, particularly Twitter, have emerged as vital sources for disseminating financial news, opinions, and market sentiments. Stock tweets, a prevalent form of user-generated content on Twitter, offer a rich and real-time glimpse into investor sentiment. However, extracting meaningful insights from these stock tweets is complex due to their unstructured nature and the noise inherent in online discussions. To tackle this challenge effectively, this research has proposed a novel approach called CPSO-SDBN, a fusion of Cognitive Particle Swarm Optimization (CPSO) and a specialized Deep Belief Network (SDBN) to optimize sentiment analysis. CPSO-SDBN dynamically adapts to the rapidly evolving and noisy stock tweet data by optimizing SDBN's architecture and hyperparameters. Through the power of CPSO, this approach tailors the sentiment analysis model, enabling it to effectively handle the intricacies and nuances of stock tweet sentiment classification. Our experiments using the "Stock Tweets for Sentiment Analysis and Prediction" dataset validate the efficacy of CPSO-SDBN, showcasing significant enhancements in sentiment analysis accuracy. This advancement equips traders, investors, and financial analysts with sharper insights into market sentiments, empowering them to make more informed decisions in the ever-dynamic landscape of the stock market. Future enhancements may include real-time data integration and advanced natural language processing for improved noisy stock tweet sentiment analysis accuracy. Exploring ensemble techniques could enhance model robustness in handling dynamic social media data.

References

[1] Jiahang Zhang, and Chi Zhang, "Do Cryptocurrency Markets React to Issuer Sentiments? Evidence from Twitter," *Research in International Business and Finance*, vol. 61, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Cheng Qian et al., "Understanding Public Opinions on Social Media for Financial Sentiment Analysis Using AI-based Techniques," *Information Processing & Management*, vol. 59, no. 6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Huihui Ni, Shuting Wang, and Peng Cheng, "A Hybrid Approach for Stock Trend Prediction Based on Tweets Embedding and Historical Prices," *World Wide Web*, vol. 24, no. 3, pp. 849-868, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Efstathios Polyzos, "Escalating Tension and the War in Ukraine: Evidence Using Impulse Response Functions on Economic Indicators and Twitter Sentiment," *Research in International Business and Finance*, vol. 66, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Daniele Ballinari, Francesco Audrino, and Fabio Sigrist, "When does Attention Matter? The Effect of Investor Attention on Stock Market volatility Around News Releases," *International Review of Financial Analysis*, vol. 82, pp. 1-28, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Loukas Ilias, and Ioanna Roussaki, "Detecting Malicious Activity in Twitter using Deep Learning Techniques," *Applied Soft Computing*, vol. 107, pp. 1-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Duong Phuong Thao Pham, Ngoc Quang Anh Huynh, and Duy Duong, "The Impact of US Presidents on Market Returns: Evidence from Trump's Tweets," *Research in International Business and Finance*, vol. 62, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Rui Fan, Oleksandr Talavera, and Vu Tran, "Information Flows and the Law of One Price," *International Review of Financial Analysis*, vol. 85, pp. 1-11, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Yihong Zhang et al., "Twitter-Aided Decision Making: A Review of Recent Developments," *Applied Intelligence*, vol. 52, pp. 13839-13854, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Darren Leitch, and Mohamed Sherif, "Twitter Mood, CEO Succession Announcements and Stock Returns," *Journal of Computational Science*, vol. 21, pp. 1-10, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Hera Shaheen, Shikha Agarwal, and Prabhat Ranjan, "MinMaxScaler Binary PSO for Feature Selection," *First International Conference on Sustainable Technologies for Computational Intelligence, Advances in Intelligent Systems and Computing*, pp. 705-716, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] Daniel Câmara, "4 - Swarm Intelligence (SI)," *Bio-inspired Networking*, pp. 81-102, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[13] David Valle-Cruz et al., "Does Twitter Affect Stock Market Decisions? Financial Sentiment Analysis During Pandemics: A Comparative Study of the H1N1 and the COVID-19 Periods," *Cognitive Computation*, vol. 14, pp. 372-387, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Attila Sóti et al., "Influence of Twitter Activity on the Stock Price of Soccer Clubs," *Social Network Analysis and Mining*, vol. 10, pp. 1-12, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Imen Hamraoui, and Adel Boubaker, "Impact of Twitter Sentiment on Stock Price Returns," *Social Network Analysis and Mining*, vol. 12, no. 1, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] N. Deepika, and M. Nirupama Bhat, "An Efficient Stock Market Prediction Method Based on Kalman Filter," *Journal of the Institution of Engineers (India): Series B*, vol. 102, no. 4, pp. 629-644, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Tareek Pattewar, and Dinesh Jain, "Stock Prediction Analysis by Customers Opinion in Twitter Data Using an Optimized Intelligent Model," *Social Network Analysis and Mining*, vol. 12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] T. Swathi, N. Kasiviswanath, and A. Ananda Rao, "An Optimal Deep Learning-based LSTM for Stock Price Prediction Using Twitter Sentiment Analysis," *Applied Intelligence*, vol. 52, pp. 13675-13688, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Suparna Dhar, and Indranil Bose, "Emotions in Twitter Communication and Stock Prices of Firms: The Impact of Covid-19 Pandemic," *Decision*, vol. 47, pp. 385-399, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Emrah Sitki Yilmaz, Asli Ozpolat, and Mehmet Akif Destek, "Do Twitter Sentiments Really Effective on Energy Stocks? Evidence from the Intercompany Dependency," *Environmental Science and Pollution Research*, vol. 29, pp. 78757-78767, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Wasiat Khan et al., "Predicting Stock Market Trends Using Machine Learning Algorithms via Public Sentiment and Political Situation Analysis," *Soft Computing*, vol. 24, pp. 11019-11043, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Giovanni Carnazza, "The Impact of the Social Mood on the Italian Sovereign Debt Market: A Twitter Perspective," *Italian Economic Journal*, vol. 10, pp. 125-154, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Pooja Mehta, Sharnil Pandya, and Ketan Kotecha, "Harvesting Social Media Sentiment Analysis to Enhance Stock Market Prediction using Deep Learning," *PeerJ Computer Science*, vol. 7, pp. 1-21, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, pp. 1942- 1948, 1995. [CrossRef] [Google Scholar] [Publisher Link]