Journal of Theoretical and Applied Information Technology
15th February 2024. Vol.102. No 3

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

OPTIMIZING SENTIMENT ANALYSIS IN ONLINE SHOPPING: UNVEILING THE PRECISION OF SIMPSON RULE-OPTIMIZED SUPPORT VECTOR MACHINE

R. ANITHA[1], D. VIMAL KUMAR[2]

[1]Research Scholar, Department of Computer Science, Nehru Arts and Science College, India
[2]Associate Professor, Department of Computer Science, Nehru Arts and Science College, India
E-mail: [1]anitharavi67@gmail.com, [2]drvimalcs@gmail.com

ABSTRACT

In the domain of online shopping, customer reviews serve as invaluable repositories of insights, encapsulating the sentiments and experiences associated with a wide array of products. However, the extraction of nuanced sentiments from this extensive pool of reviews faces a formidable challenge due to the inherent complexities of language and context. Sentiment analysis, a pivotal tool for distilling sentiments from textual data, grapples with accurately deciphering nuanced expressions, sentiment subtleties, and contextual intricacies. To address these challenges, this study introduces the Simpson Rule-Optimized Support Vector Machine (SR-SVM) for sentiment analysis in online shopping. Built on Support Vector Machines (SVM) principles, SR-SVM leverages Simpson's rule to optimize sentiment pattern identification within the expansive landscape of online shopping reviews. Through the application of mathematical optimization techniques, SR-SVM refines sentiment analysis outcomes, promising a more nuanced understanding of customer sentiments. Preliminary results indicate a noteworthy enhancement in sentiment classification accuracy, underscoring the transformative potential of SR-SVM in optimizing sentiment analysis for online shopping platforms. This study opens a promising avenue for refining and advancing sentiment analysis methodologies in the dynamic and complex context of online retail.

Keywords: *Sentiment, Classification, Amazon, Simpson Rule, SVM, Optimization*

1. INTRODUCTION

Online shopping has transformed the retail landscape, offering consumers unprecedented convenience and various products at their fingertips. With just a few clicks, shoppers can explore a global marketplace, compare prices, and make purchases without leaving the comfort of their homes. This convenience has led to a significant shift in consumer behaviour, with a growing number of people shopping online [1], [2]. E-commerce platforms have become integral to modern retail, providing millions worldwide with a seamless and efficient shopping experience. Online shopping offers numerous benefits, including 24/7 accessibility, various choices, and the convenience of doorstep delivery. Additionally, the ability to read and contribute to customer reviews enhances the decision-making process. Before purchasing, shoppers often turn to these reviews to gain insights into product quality, reliability, and the overall buying experience [3].

Sentiment analysis, synonymous with opinion mining, assumes a pivotal role in the intricate landscape of natural language processing, aiming to decipher the subtle emotional undercurrents interwoven within textual narratives [4]. In this era characterized by an exponential surge in digital data, the ability to distil sentiments from the deluge of information disseminated across social media, reviews, and customer feedback becomes not just advantageous but imperative [5]. Sentiment analysis, underpinned by the fusion of machine learning algorithms and sophisticated linguistic techniques, meticulously classify textual content, affording businesses and researchers the invaluable ability to glean insights into the prevailing public opinion – a compass guiding decision-making and strategy formulation [6].

The trajectory of machine learning within the sentiment analysis realm depicts a perpetual evolution journey. As technological strides push the boundaries of what is achievable, incorporating deep learning and natural language processing imparts newfound capabilities to models, allowing them to extract nuanced insights from the vast ocean of textual data [7]. The horizon promises the integration of real-time processing. This leap envisages machine learning models comprehending

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

contextual intricacies in an even more granular and timely fashion. The symbiotic relationship between machine learning and sentiment analysis stands as a linchpin, an ongoing collaboration essential for unravelling the complex tapestry of human emotions intricately woven into the digital fabric of conversations [8]. It is a journey where each advancement augments the precision of sentiment analysis and serves as a testament to the inexhaustible potential for understanding the rich tapestry of human expression in the digital age [9]–[11].

1.1. Problem Statement

Cross-domain sentiment variability presents a substantial challenge in sentiment analysis, as existing models need help transitioning seamlessly across diverse product domains. The problem lies in the need for models to grasp nuanced sentiments unique to each part, hindering their generalization capabilities. The lack of adaptability leads to diminished accuracy and effectiveness when applied to reviews spanning various industries. Addressing this challenge requires innovative approaches to enhance the model's capacity to discern sentiment expressions specific to different products, ensuring a more robust and adaptable sentiment analysis framework.

1.2. Motivation

The motivation for delving into the intricacies of cross-domain sentiment variability stems from the critical need to enhance sentiment analysis models' adaptability and generalization capabilities. Current models often need help seamlessly navigating sentiments across diverse product domains, resulting in diminished accuracy. This research aims to innovate and develop methodologies that empower sentiment analysis models to discern and understand nuanced emotions unique to each part. By addressing cross-domain variability, we aspire to elevate the reliability of sentiment predictions, making these models versatile and practical across various industries. The motivation lies in bridging the gap between existing limitations and the demand for robust sentiment analysis that can provide accurate insights in the face of diverse linguistic patterns and user expectations.

1.3. Research Objective

The primary objective of this research is to propose and evaluate the effectiveness of the Simpson Rule-based Support Vector Machine (SR-SVM) algorithm in mitigating the challenges posed by cross-domain sentiment variability in sentiment analysis. The focus is enhancing sentiment analysis models' adaptability and generalization capabilities when transitioning across diverse product domains. By leveraging SR-SVM, we aim to develop a robust sentiment analysis framework that can discern and understand nuanced sentiments unique to each part, ultimately improving the accuracy and versatility of sentiment predictions across various industries.

2. LITERATURE REVIEW

"Cross-Correlation Attention" [12] introduces aspect-level sentiment analysis by incorporating cross-correlation attention mechanisms. It dynamically adjusts the model's focus based on intricate relationships between different aspects of textual data. This adaptive mechanism significantly enhances the granularity of sentiment analysis, allowing for a more nuanced understanding of sentiment expressions associated with specific elements. "Contextualized Bidirectional Language and Generative Adversarial" [13] presents a sentiment analysis model that integrates BERT with techniques. The technical innovation lies in enhancing the contextualized embeddings obtained from BERT through adversarial training provided by CBLBGA. Integrating transformer architectures with generative malicious processes contributes to improved accuracy in text sentiment analysis, marking a significant advancement in the field. "Vision-Language Pre-Training (VLP)" [14] combines vision and language pre-training, capturing sentiment from both textual and visual modalities. The technical ingenuity broadens the scope of sentiment analysis, allowing the model to leverage richer information from diverse data sources. This extension represents a significant step toward achieving a comprehensive understanding of sentiments in the context of both text and visuals.

"Customer Satisfaction" [15] evaluates the sentiments present in restaurant customer reviews during the COVID-19 pandemic. The essential contribution lies in using sentiment analysis methodologies to discern the evolving views of customers amid challenging circumstances. It demonstrates the adaptability of sentiment analysis methodologies to dynamic, real-world scenarios. "Dynamic Hypergraph Convolutional Network" [16] enhances the model's ability to discern nuanced sentiments from diverse data sources by adapting hypergraph structures. It represents a significant advancement in the evolving landscape of multimodal sentiment analysis. By introducing a dynamic hypergraph approach, the study contributes

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

to the adaptability and sophistication of sentiment analysis models dealing with multimodal data. "Product Feature Sentiment Analysis" [17] is based on Gated Recurrent Units (GRU) with Context-Aware Pooling (CAP) tailored explicitly for product feature sentiment analysis, focusing on Chinese sarcasm recognition. It lies in the seamless integration of GRU and CAP techniques, enabling the model to effectively handle sarcasm nuances while analyzing sentiments associated with specific product features. This approach represents a significant advancement in sentiment analysis models, showcasing their adaptability to specific linguistic and contextual nuances, particularly in product reviews and sentiment expression.

"Large Language Models" [18] explores sentiment nuances within the unique context of performing arts feedback. The study aims to enhance the understanding of sentiments expressed by audiences, critics, and enthusiasts in response to various performances. This approach represents a specialized application of sentiment analysis methodologies, catering to the intricacies of the performing arts domain. "Pre-trained Models" [19] focus on synthesizing existing knowledge and methods related to rapid learning for sentiment analysis. By systematically reviewing the efficiency gains and challenges associated with prompt-based learning approaches, the study provides insights into optimizing the usage of pre-trained models for sentiment analysis applications. "Prompt-based learning with Attention Mechanism" [20] integrates the prompt-based learning techniques with attention mechanisms to enhance the model's focus on specific aspect categories. By dynamically adjusting attention, the model achieves a more nuanced understanding of sentiments associated with different aspects. This novel methodology advances sentiment analysis by offering a refined and context-aware approach to dissecting sentiments within distinct categories.

"Coordinated-Joint Translation Fusion Framework" [21] attempts to integrate translation fusion and graph convolutional networks by enabling a coordinated analysis of multimodal data. It enhances the model's ability to capture interactions between different modalities and their influence on sentiments. "2-Tuple Fuzzy Linguistic Model" [22] is proposed for recommending health care services, grounded on aspect-based sentiment analysis. It incorporates fuzzy linguistic modelling to capture the uncertainty and imprecision inherent in healthcare service recommendations. Combining aspect-based sentiment analysis with fuzzy linguistic modelling provides a nuanced and contextually aware approach to recommending health care services. "Co-Space Representation Interaction Network" [2] operates in a shared space representation, facilitating the interaction between different modalities. It enhances the model's capture of complex relationships and interactions among diverse data sources. It advances multimodal sentiment analysis by introducing a novel network architecture focusing on shared space representation and interaction networks, providing a sophisticated framework for understanding sentiments in diverse contexts. Utilization of bio-inspired optimization makes better result in many classification tasks [23], [24], [33]–[41], [25]–[32].

"Cloze Task Network based Convolutional Hierarchical Attention Networks (CCHAN)" [42] offers a seamless solution for sentiment analysis across diverse domains. By integrating a Cloze Task Network (CTN) with Convolutional Hierarchical Attention Networks (CHAN), CCHAN addresses the intricate challenges associated with varying linguistic styles in different domains. Incorporating a CTN empowers the model with domain-agnostic representations, ensuring adaptability to diverse contextual nuances. Augmented by CHAN, CCHAN excels in capturing intricate hierarchical dependencies within textual data, allowing for nuanced sentiment analysis. This end-to-end architecture enhances efficiency and eliminates the need for complex preprocessing steps and domain-specific feature engineering, making CCHAN a versatile and robust model for cross-domain sentiment classification.

"Hybrid Neural Network techniques using Binary Coordinate Ascent Algorithm (HNN-BCA)" represents an approach to sentiment analysis. This method seamlessly blends the power of HNN with the efficiency of the BCA algorithm. The HNN aspect harnesses the pattern recognition capabilities of neural networks, enhancing the model's proficiency in deciphering the intricacies of emotional language. The BCA algorithm introduces a strategic optimization mechanism, streamlining the learning process for improved convergence and heightened performance. By incorporating binary representations within the coordinate ascent algorithm, the model navigates sentiment analysis feature spaces with enhanced computational efficiency. This fusion of neural network architecture and optimization strategies demonstrates the continuous evolution of sentiment analysis.

The need for the present work stems from the escalating significance of online shopping and the pivotal role customer reviews play in shaping purchasing decisions. As the e-commerce landscape continues to expand, businesses are increasingly reliant on understanding customer sentiments expressed in reviews to refine their products and services. However, existing sentiment analysis methodologies encounter challenges in accurately deciphering the nuanced language and contextual intricacies present in online shopping reviews. The current literature underscores the demand for advanced sentiment analysis techniques that can navigate these challenges effectively.

The novelty of this work lies in the introduction of the Simpson Rule-Optimized Support Vector Machine (SR-SVM) for sentiment analysis in online shopping. While Support Vector Machines (SVM) are well-established in sentiment analysis, the integration of Simpson's rule for optimization represents a unique approach. This novel methodology aims to enhance sentiment classification accuracy and provide a more nuanced understanding of customer sentiments in the diverse landscape of online shopping reviews. The current literature emphasizes the need for innovative solutions that go beyond conventional sentiment analysis approaches, and SR-SVM contributes to this by merging mathematical optimization principles with advanced machine learning techniques.

In the context of existing research, the proposed SR-SVM methodology offers a distinctive contribution by addressing the limitations of current sentiment analysis approaches. By optimizing sentiment pattern identification through Simpson's rule, this work presents a novel approach to refining sentiment analysis outcomes. The uniqueness of the SR-SVM model positions it as a pioneering solution in the realm of sentiment analysis for online shopping. As such, this research adds a valuable dimension to the current literature by introducing a novel methodology that holds the potential to significantly advance the field of sentiment analysis in the context of online retail.

3. SIMPSON RULE-OPTIMIZED SUPPORT VECTOR MACHINE

3.1. Defining Simpson's Rule Integration Strategy

The integration strategy for Simpson's Rule (SR) within the context of Support Vector Machines (SVM) involves the weighted averaging of decision function values across distinct regions of the feature space. Let $f(x)$ represent the SVM decision function for a given data point x, which is a function of the feature vector. The integrated decision function $F(x)$ incorporating SR is defined in Eq.(1), and it is the weighted sum of $f(x)$ values across regions R_i of the feature space.

$$F(x) = \sum_{i=1}^{N} w_i . \int_{R_i}^{1} f(x)dx \tag{1}$$

where N represents the number of distinct regions partitioning the feature space, w_i denotes the weight assigned to the i-th region, and the integral is computed using SR.

The SR integration over a region R_i is given by Eq.(2).

$$\int_{R_i}^{1} f(x)dx = \frac{h}{3}\begin{bmatrix} f(x_0) + 4f(x_1) \\ +2f(x_2) + \cdots . +4f(x_{n-1}) + f(x_n) \end{bmatrix} \tag{2}$$

where h is the step size, n is the number of subintervals in the region, and x_k represents the k-th point within the region.

The weights w_i assigned to each region are determined based on the specific characteristics or importance of the region in the overall SVM decision function. These weights can be derived using Eq.(3), the domain knowledge or a learning process during model training. The normalization constant in Eq.(3) ensures that the weights collectively sum to 1, and the importance measure reflects the significance of each region in contributing to the integrated decision function.

$$w_i = \frac{1}{Normalization\ Constant \times Importance\ Measure_i} \tag{3}$$

The final integrated decision function $F(x)$ is a combination of SVM decision function values weighted by the SR integration over different feature space regions, and the same is expressed as Eq.(4).

$$F(x) = \sum_{i=1}^{N} \frac{w_i}{3}\begin{bmatrix} f(x_0) + 4f(x_1) + 2f(x_2) + \\ \dots . +4f(x_{n-1}) + f(x_n) \end{bmatrix} \tag{4}$$

Journal of Theoretical and Applied Information Technology
15th February 2024. Vol.102. No 3

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

Algorithm 1: SR Integrated Decision Function

Input:

- $f(x)$: SVM decision function for a given data point x
- N: Number of distinct regions partitioning the feature space
- w_i: Weights assigned to each region based on importance measures
- R_i: Regions of the feature space
- h: Step size for SR integration
- n: Number of subintervals in each region
- x_k: Points within each region

Output:

- $F(x)$: Integrated decision function incorporating SR

Procedure:

1. Initialize: Set $F(x)$ to zero.
2. For each region R_i:
   a. Compute weights w_i : Determine the importance measures for each region and normalize the weights.
   b. Apply SR Integration: Calculate the integrated decision function for the region using SR.
   c. Weighted Summation: Update $F(x)$ by adding the weighted integration value for the current region.

3.2. Feature Selection in SR-SVM

The relevance analysis of features is crucial for the overall effectiveness of the integrated decision function. The SVM decision function, denoted by $f(x)$, depends on the feature vector x. Feature relevance analysis involves the computation of feature importance scores, which directly contribute to the weighted decision function values across different regions. Mathematically, this can be expressed as Eq.(5).

$$Importance\ Score_{(x_i)} = Correlation(f(x), x_i) \tag{5}$$

where x_i represents the i-th feature, and the correlation between $f(x)$ and $x_i$is computed to determine the relevance of the feature to the SVM decision function.

The relevant features are selected for the SR-SVM model from the relevance analysis carried out using Eq.(5). The selection process involves choosing features with higher importance scores to enhance the integration strategy. The selected feature set is denoted as $X_{selected}$ is expressed as Eq.(6). The term threshold in Eq.(6) is determined through cross-validation to ensure that only sufficiently relevant features are included.

$$X_{selected} = \{x_i | Importance\ Score_{(x_i)} \geq Threshold\} \tag{6}$$

An analysis of the integrated function across different regions is performed to assess the impact of the selected features on the integrated decision function. This involves evaluating how the inclusion of specific features influences the weighted decision function values. The impact on the integrated decision function, denoted by $F_{selected}(x)$, is expressed as Eq.(7).

$$F_{selected}(x) = \sum_{i=1}^{N} w_i . \int_{R_i}^{1} f(x_{selected}) dx \tag{7}$$

where $f(X_{selected})$ represents the SVM decision function with only the selected features.

The feature selection process is iterative, allowing for refinement based on the impact analysis. The selection criteria and threshold may be adjusted, and the integration impact analysis is repeated to enhance the relevance of features in contributing to the integrated decision function.

Algorithm 2: Feature Selection

Input:

- $f(x)$: SVM decision function for a given data point x
- X: Feature vector
- N: Number of distinct regions partitioning the feature space
- w_i: Weights assigned to each region based on importance measures
- R_i: Regions of the feature space
- Threshold: Importance score threshold for feature selection

Output:

- $X_{selected}$: Subset of relevant features
- $F_{selected}(x)$: Integrated decision function with selected features

Procedure:

1. Initialize:
   a. Set $X_{selected}$ to an empty set.
2. Relevance Analysis:
   a. Calculate the importance score for each feature.
3. Feature Selection:

a. Select features with importance scores greater than or equal to the threshold.
b. Form the subset $X_{selected}$ with the selected features.

4. **Integration Impact Analysis:**
   a. Modify the SVM decision function $f(x)$ to consider only the selected features.
   b. Evaluate the integrated decision function $F_{selected}(x)$ using SR over different regions.

3.3. Kernel Function Selection

The choice of a suitable kernel function significantly influences the integrated decision function. A kernel function ($K(x,y)$) is employed to map input data points into a higher-dimensional space, facilitating the delineation of non-linear relationships. The integration strategy is inherently linked to selecting an appropriate kernel function in the SVM framework, expressed in Eq.(8).

$$K(x,y) = \Phi(x).\Phi(y) \tag{8}$$

where $\Phi(x)$ and $\Phi(y)$ represent the mappings of the input data points x and y into the higher-dimensional space.

The SVM decision function ($f(x)$) is kernelized to accommodate the transformed feature space, and it is defined using Eq.(9).

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i K(x, x_i) + b \tag{9}$$

where m is the number of support vectors, α_i are the Lagrange multipliers, y_i is the class label of the i-th support vector, x_i is the i-th support vector, and b is the bias term.

The integration strategy necessitates modifying the SVM decision function to incorporate SR-based integration, and Eq.(10) expresses the same.

$$F(x) = \sum_{i=1}^{N} w_i . \int_{R_i} f(x)dx \tag{10}$$

The integration process considers the kernelized decision function $f(x)$, ensuring that SR is applied to the integrated decision function across different regions. The final integrated decision function, considering both the kernelized SVM decision function and SR integration, is given by Eq.(11).

$$F(x) = \sum_{i=1}^{N} w_i . \int_{R_i} \left(\sum_{j=1}^{m} \alpha_j y_j K(x, x_j) + b \right) dx \tag{11}$$

Eq.(11) encapsulates the comprehensive nature of the SR-SVM model, where the kernelized SVM decision function is integrated across various regions of the feature space.

Algorithm 3: Kernel Function Selection

Input:

- $K(x,y)$: Kernel function mapping input data points x and y into a higher-dimensional space
- $f(x)$: SVM decision function in the transformed feature space
- N: Number of distinct regions partitioning the feature space
- w_i: Weights assigned to each region based on importance measures
- R_i: Regions of the feature space

Output:

- $F(x)$: Integrated decision function incorporating SR with kernelized SVM decision function

Procedure:

1. Kernel Function Choice:
   a. Select an appropriate kernel function $K(x,y)$ based on the nature of the data and integration strategy.
   b. Define the mapping function $\Phi(x)$ associated with the chosen kernel.
2. Kernelized Decision Function:
   a. Transform the SVM decision function $f(x)$ using the selected kernel function $K(x,y)$.
   b. Express the kernelized decision function as a summation of support vectors, Lagrange multipliers, class labels, and the bias term.
3. Integration Modification:
   a. Modify the SVM decision function to incorporate SR-based integration.
   b. Express the integrated decision function $F(x)$ as a weighted sum of the kernelized decision function values over different regions.
4. Overall Kernelized Integrated Decision Function:

a. Formulate the final integrated decision function, considering both the kernelized SVM and SR integration.
b. Sum the weighted kernelized decision function values across different regions to obtain the comprehensive, integrated decision

Algorithm 3 emphasizes the choice of an appropriate kernel function, the transformation of the SVM decision function, and the modification of the decision function to accommodate SR-based integration. The output includes the final integrated decision function ($F(x)$), which reflects the amalgamation of the kernelized SVM decision function and the integration strategy across various regions of the feature space.

3.4. Hyperparameter Optimization

The practical hyperparameter tuning is paramount for optimizing the performance of the model. The hyperparameters associated with the SVM and SR integration are crucial in shaping the integrated decision function. The optimization process is mathematically expressed as Eq.(12).

$$Optimal\ Hyperparameters = \arg min_{\theta}\ Objective(\theta) \qquad (12)$$

where θ represents the set of hyperparameters, and the objective function $Objective(\theta)$ quantifies the model's performance, which considers factors such as accuracy or loss.

Exploring the hyperparameter space involves systematically varying hyperparameter values and evaluating their impact on the model's performance. This exploration is often performed through techniques like randomized search in a grid. The hyperparameter space Θ is defined as Eq.(13).

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_n\} \qquad (13)$$

where n represents the number of hyperparameters and each θ_i represents a specific hyperparameter value within its respective range.

Cross-validation is employed to assess the performance of different hyperparameter configurations robustly. The objective function is evaluated over multiple folds of the dataset to mitigate the impact of variations in training and validation sets. The cross-validation performance metric is defined as Eq.(14).

$$CV\_Metric = \frac{1}{K}\sum_{k=1}^{K} Metric(Model_k) \qquad (14)$$

where K denotes the number of folds and $Metric(Model_k)$ represents the chosen evaluation metric for the k-th fold.

The hyperparameters yielding the optimal performance are selected and used to update the SR-SVM model. The updated hyperparameters θ^* are determined based on the optimization process expressed in Eq.(15).

$$\theta^* = \arg min_{\theta}\ Objective(\theta) \qquad (15)$$

These updated hyperparameters are then employed in the final SR-SVM model, ensuring an optimal configuration for both the SVM and SR integration.

Algorithm 4: Hyperparameter Optimization

Input:

- Θ: Hyperparameter space
- $Objective(\theta)$: Objective function measuring model performance
- K: Number of cross-validation folds

Output:

- θ^*: Optimized hyperparameters

Procedure:

1. Hyperparameter Tuning:
   a) Initialize θ^* as the first hyperparameter configuration.
   b) Evaluate $Objective(\theta)$ for θ^*
2. Hyperparameter Space Exploration:
   a) For each θ_i in Θ:
      - Update θ^* if $Objective(\theta_i) < Objective(\theta^*)$.
3. Cross-Validation:
   a) Split the dataset into K folds for cross-validation.
4. For each fold:
   a) Train the SR-SVM model using the current hyperparameter configuration.
   b) Evaluate the model performance using $Metric(Metric_k)$.
5. Hyperparameter Update:
   a) Choose θ^* as the hyperparameter configuration

yields the minimum cross-validation metric.

b) Return θ^* as the optimized hyperparameters for the SR-SVM model.

3.5. Evaluation

The performance evaluation involves assessing the effectiveness of the integrated decision function across the feature space. The decision function $F(x)$ combines the kernelized SVM decision function and SR-based integration. It is evaluated over a validation set V or through cross-validation expressed as Eq.(16).

$$Performance\ Metric = \frac{1}{|V|}\sum_{x\in V} Metric(F(x), True\ Label(x)) \quad (16)$$

where $|V|$ represents the size of the validation set, $Metric(F(x), True\ Label(x))$ denotes the chosen evaluation metric comparing the integrated decision function values to the true labels.

Hyperparameters are analyzed within their specified ranges to find the impact of hyperparameter choices. The performance metric is observed as hyperparameters are systematically modified:

$$Hyperparameter\ Impact = \frac{1}{|\Theta|}\sum_{\theta\in\Theta} Performance\ Metric(\Theta) \quad (17)$$

where $|\Theta|$represents the number of hyperparameter configurations explored, and $Performance\ Metric(\Theta)$ signifies the performance metric associated with the hyperparameter configuration θ.

The relevance of selected features in contributing to the integrated decision function is crucial. The assessment involves analyzing the impact of each feature on the performance metric expressed in Eq.(18).

$$Feature\ Relevance = \frac{1}{|X_{selected}|}\sum_{x_i\in X_{selected}} Performance\ M\ldots \quad (18)$$

where $|X_{selected}|$ represents the number of selected features and x_i signifies each selected feature.

The generalization capability of the SR-SVM model is evaluated on a separate test, the set T, to ensure its performance on unseen data. The generalization metric is defined as Eq.(19).

$$Generalization\ Metric = \frac{1}{|T|}\sum_{x\in T} Metric(F(x), True\ Label(x)) \quad (19)$$

This assessment provides insights into how well the SR-SVM model adapts to diverse data distributions.

Algorithm 5: Evaluation

Input:
- $F(x)$: Integrated decision function of the SR-SVM model
- V: Validation set
- T: Test set
- $X_{selected}$: Subset of relevant features
- Θ: Hyperparameter space

Output:
- Hyperparameter Impact
- Feature Relevance
- Generalization Metric

Procedure:

1. Decision Function Evaluation:
   a) Evaluate the integrated decision function $F(x)$ over the validation set V.
   b) Compute the performance metric comparing $F(x)$ to the true labels in V.

2. Hyperparameter Impact Analysis:
   a) For each hyperparameter configuration θ in Θ:
      - Train the SR-SVM model with hyperparameters θ.
      - Evaluate the performance metric on the validation set V.
   b) Calculate the average performance metric over all explored hyperparameter configurations.

3. Feature Relevance Assessment:
   a) For each selected feature x_i in $X_{selected}$:
      - Evaluate the performance metric with only the selected features.
   b) Calculate the average performance metric over all selected features.

4. Model Generalization:
   a) Evaluate the integrated decision function $F(x)$ over the test set T.
   b) Compute the generalization metric comparing $F(x)$ to the true labels in T.

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

3.6. Refine and Validate

The refinement process involves iteratively updating the model based on the insights gained from the performance evaluation. The model is refined by adjusting hyperparameters, feature selections, and integration strategies to enhance overall performance. The refinement process is mathematically expressed as Eq.(20).

$$Refined\ Model = Refine(Current\ Model, Evaluation\ Re \tag{20}$$

where $Refine\ (.)$ represents the refinement process, and $Current\ Model$ indicates the current state.

The refined model undergoes iterative validation to assess its performance improvements. The validation is conducted using the set V or through cross-validation, providing a consistent evaluation framework. Eq.(21) represents the iterative validation process.

$$Iterative\ Validation = \frac{1}{Iterations}\sum_{i=1}^{Iterations} Metric(Refined \tag{21}$$

where $Refined\ Model_i$ signifies the model state after the i-th iteration, and **Iterations** represent the total number of refinement iterations.

The refined model is further validated on the test set T to ensure its generalization capability on unseen data. The validation metric is expressed as Eq.(22). This assessment externally validates the model's effectiveness beyond the training and validation phases.

$$Validation\ Metric = Metric(Refined\ Model, T) \tag{22}$$

The refinement and validation process establishes a feedback loop with the integration strategy. The insights gained from performance evaluations influence how SR is applied, the relevance of features, and the choice of kernel functions, forming an adaptive integration strategy:

$$Integrated\ Decision\ Function_{Refined} = Integrate(Refined\ Model, Integration\ Strategy\ Feedback) \tag{23}$$

where $Integrate(.)$ represents the integration process incorporating feedback from the refinement and validation.

Algorithm 7: Refinement and Validation

Input:

- Current Model: The current state of the SR-SVM model
- Evaluation Results: Results from the performance evaluation stage
- V: Validation set
- T: Test set
- Iterations: Total number of refinement iterations

Output:

- Refined Model: Model state after the refinement process
- Iterative Validation Metric: Metric evaluating the refined model over iterations
- Validation Metric: Metric evaluating the refined model on the test set

Procedure:

1. Model Refinement:
   a. Update the current model based on insights from the evaluation results.
   b. Set the Refined Model to the updated model.
2. Iterative Validation:
   a. For each iteration i from 1 to Iterations:
3. Training:
   a. Train the $Refined\ Model_i$ using the updated model.
4. Evaluation:
   a. Compute the average metric over all iterations to obtain the Iterative Validation Metric.
5. Model Validation:
   a. Evaluate the Refined Model on the test set T.
   b. Compute the Validation Metric based on the model's performance on the test set.
6. Integration Feedback Loop:
   a. Incorporate insights from the refinement and validation process into the integration strategy.
   b. Update the integrated decision function using the adapted integration strategy.
7. Output Result:
   a. Return the Refined Model, Iterative Validation Metric, and Validation Metric as the outputs of the refinement and validation process in the SR-SVM model.

Journal of Theoretical and Applied Information Technology
15th February 2024. Vol.102. No 3

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

3.7. Integration Culmination

The final step involves configuring the model based on the insights gained from the refinement and validation process. Integration culmination integrates the optimal hyperparameters, feature selections, and an adapted integration strategy. Eq.(24) mathematically expresses the same.

$$\text{Final Model} = \text{Integrate}(\text{Refined Model}, \text{Integration Strategy}) \quad (24)$$

where the *Refined Model* represents the model state after the refinement process, and *Integration Strategy Feedback* signifies the feedback loop with the integration strategy.

To ensure the robustness and reliability of the final model, a comprehensive assessment of its performance is conducted. The Performance Assurance Metric is calculated using Eq.(25) and evaluates the final model on a validation set or through cross-validation.

$$\text{Performance Assurance Metric} = \frac{1}{|V|}\sum_{x\in V} \text{Metric}(\text{Final Model}(x), \text{True Label}(x)) \quad (25)$$

where $|V|$ represents the size of the validation set, and $\text{True Label}(x)$ denotes the true label for data point x.

The deployment involves using the model to predict new, unseen data points. The deployment process is mathematically represented as Eq.(26).

$$\text{Predicted Labels} = \text{Final Model}(\text{Unseen Data}) \quad (26)$$

where Unseen Data represents new data points for which predictions are generated.

The Integrated System ensures seamless communication between the model and external components, as expressed in Eq.(27).

$$\text{Integrated System} = \text{Integrate}\ (\text{Final Model}, \text{External System})$$

where External System signifies the external components or systems with which the SR-SVM model is integrated

Algorithm 8: Integration Culmination

Input:
- Refined Model: Model state after the refinement process
- Integration Strategy Feedback: Feedback loop with the integration strategy
- V: Validation set
- External System: External components or systems
- Unseen Data: New, unseen data points

Output:
- Final Model: A configured model for deployment
- Performance Assurance Metric: Metric assessing the final model's performance
- Predicted Labels: Predicted labels for unseen data points
- Integrated System: Integration with external systems

Procedure:
1. Final Model Configuration:
   a. Integrate the Refined Model with the Integration Strategy Feedback to obtain the Final Model.
2. Model Performance Assurance:
   a. Evaluate the Final Model on the validation set V to compute the Performance Assurance Metric.
3. Model Deployment:
   a. Use the Final Model to generate predicted labels for the Unseen Data.
4. Integration with External Systems:
   a. Integrate the Final Model with the External System to create the Integrated System.

Algorithm 9 provides the overall steps involved in SR-SVM.

Algorithm 9: SR-SVM

Input:
- X: Input dataset
- Y: Class labels for each data point in X
- N: Number of distinct regions partitioning the feature space
- Θ: Hyperparameter space
- V: Validation set
- T: Test set
- Iterations: Total number of refinement iterations
- External System: External components or systems
- Unseen Data: New, unseen data points

Output:
- Final Model: A configured model for deployment

- Performance Assurance Metric: Metric assessing the final model's performance
- Predicted Labels: Predicted labels for unseen data points
- Integrated System: Integration with external systems

Procedure:

1. Define Integration Strategy:
   a. Specify the integration strategy, incorporating SR into the SVM decision function.
2. Data Preprocessing:
   a. Clean and preprocess the input data, handling missing values and scaling features.
3. Feature Selection:
   a. Choose relevant features based on their impact on the integrated decision function.
4. Kernel Function Selection:
   a. Choose an appropriate kernel function for the SVM, considering the nature of the data and integration strategy.
5. Hyperparameter Optimization:
   a. Fine-tune hyperparameters for the SR-SVM model through exploration of the hyperparameter space.
6. Evaluation:
   a. Assess the SR-SVM performance on a validation set or through cross-validation.
7. Refinement and Validation:
   a. Refine the model iteratively based on performance evaluations.
   b. Validate the refined model on the test set and assess its generalization capabilities.
   c. Integrate feedback from the refinement process into the integration strategy.
8. Integration Culmination:
   a. Configure the final model based on the refined model state and integration strategy feedback.
   b. Assess the performance of the final model on a validation set.
   c. Deploy the final model for generating predictions on new, unseen data points.
   d. Integrate the final model with external systems for practical utilization.

4. CONSUMER REVIEWS DATASET

The "Consumer Reviews Dataset," featuring over 34,000 reviews for Amazon products from Datafiniti's Product Database, is valuable for analyzing consumer sentiments and product performance. With comprehensive information, including product details, ratings, and review text, analysts can delve into diverse analyses. These include identifying the most-reviewed products and exploring the correlation between early reviews and factors like pricing and days available for sale. This subset dataset, part of a more extensive collection, offers a foundation for in-depth studies, including sentiment mapping for machine learning applications. Aligned with Datafiniti's mission of providing standardized databases, the "Consumer Reviews Dataset" is essential for researchers and analysts seeking insights into Amazon's consumer electronics landscape trends. For those desiring a more extensive understanding, access to the complete dataset is available through Datafiniti's Product API, enhancing its utility for professionals in the realm of e-commerce analytics and consumer insights.

Table 1. Features of Consumer Reviews Dataset

Feature	Description
id	Unique identifier for each entry
dateAdded	Date when the entry was added to the dataset
dateUpdated	Date when the entry was last updated
name	Name of the product
asins	Amazon Standard Identification Numbers associated with the product
brand	Brand of the product
categories	Categories to which the product belongs
primaryCategories	Primary category to which the product is assigned
imageURLs	URLs of images associated with the product
keys	Keywords or key phrases associated with the product
manufacturer	Manufacturer of the product
manufacturerNumber	Manufacturer's identification number for the product
reviews.date	Date of the consumer reviews
reviews.dateSeen	Dates when the reviews were observed or recorded
reviews.didPurchase	Indicator of whether the reviewer claims to have purchased the product

reviews.doRecommend	Indicator of whether the reviewer recommends the product
reviews.id	Unique identifier for each review
reviews.numHelpful	Number of users who found the review helpful
reviews.rating	Rating given by the reviewer
reviews.sourceURLs	URLs of the sources from which reviews were obtained
reviews.text	Text content of the reviews
reviews.title	Title or summary of the reviews
reviews.username	Username of the reviewer
sourceURLs	URLs of the sources providing information about the product

5. PERFORMANCE METRICS

- True Positive Rate: True Positive Rate (TPR) is the ratio of correctly identified positive instances to the total actual positive instances.
- False Postive Rate: False Postive Rate (FPR) is the proportion of negative instances incorrectly predicted as positive by the model among all actual negative instances.
- True Negative Rate: True Negative Rate (TNR) is the ratio of correctly identified negative instances to total negative instances.
- False Negative Rate: False Negative Rate (FNR) is the ratio of incorrectly identified negative instances to the total actual positive instances.
- Classification Accuracy: Classification Accuracy (CA) is the proportion of correctly classified instances (both true positives and true negatives) among all instances in the dataset.
- F-Measure: F-Measure (FM) is the harmonic mean of precision and recall, emphasizing balanced performance in both precision (positive predictive value) and recall (sensitivity).

6. RESULTS AND DISCUSSIONS

6.1. Positivity Analysis

Figure 1 and Table 2 collectively serve as a comprehensive visual and tabular representation of the TPR and FPR metrics for three distinct sentiment classification algorithms: CCHAN, HNN-BCA, and SR-SVM. The graph visually encapsulates the comparative performance of these algorithms, while Table 2 offers precise numerical values corresponding to their TPR and FPR percentages. TPR signifies the algorithms' effectiveness in correctly identifying positive instances, while FPR reflects how they incorrectly classify negative instances as positive. This dual representation provides a holistic understanding of the algorithms' sensitivity and specificity in discerning sentiment, forming a basis for nuanced discussions.

The TPR, indicating the algorithm's ability to identify positive instances accurately, unfolds distinctive insights into their performance. CCHAN, with a TPR of 63.228%, demonstrates moderate success in capturing actual positive instances. The incorporation of Cloze Task Network and Convolutional Hierarchical Attention Networks contributes to its contextual understanding, albeit with room for improvement. HNN-BCA exhibits a slightly higher TPR of 67.318%, showcasing improved sensitivity through the synergy of Hybrid Neural Network techniques and Binary Coordinate Ascent Algorithm. SR-SVM stands out with a remarkable TPR of 81.033%, emphasizing its superiority in accurately identifying positive instances. The robust performance of SR-SVM can be attributed to the intricacies of Simpson Rule-Optimized Support Vector Machines, indicating efficient sentiment discrimination through optimal hyperplanes.

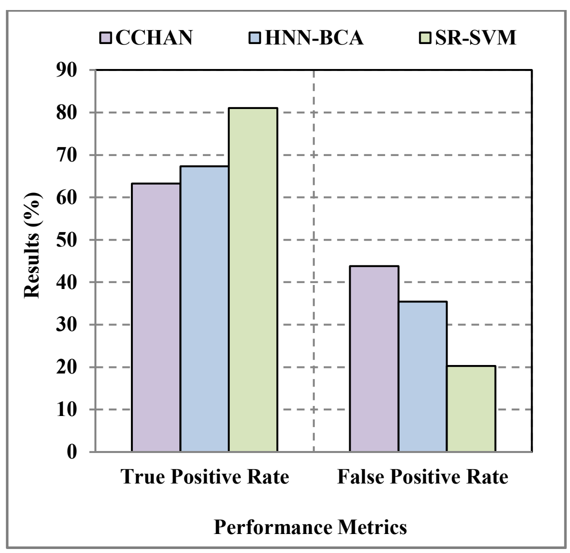

Figure 1. True Positive Rate and False Positive Rate

FPR measures the algorithms' propensity to misclassify negative instances as positive, it delves into their specificity and the ability to avoid false

Journal of Theoretical and Applied Information Technology
15th February 2024. Vol.102. No 3

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

positives. CCHAN presents an FPR of 43.791%, indicating a notable risk of labelling actual negative instances as positive. The combination of the Cloze Task Network and Convolutional Hierarchical Attention Networks, while offering contextual understanding, showcases a trade-off with specificity. HNN-BCA improves specificity, with a lower FPR of 35.412%, suggesting a more discerning nature in avoiding false positives than CCHAN. The Hybrid Neural Network techniques and the Binary Coordinate Ascent Algorithm contribute to a more balanced performance. SR-SVM stands out again with an impressive FPR of 20.260%, showcasing stringent control over false positives. The Simpson Rule-Optimized Support Vector Machines' capacity for finding optimal hyperplanes contributes to the algorithm's ability to avoid misclassifying negative instances.

The significant analysis of TPR and FPR for the three algorithms sheds light on their strengths and weaknesses in sentiment classification. While CCHAN emphasizes adaptability and HNN-BCA balances sensitivity and specificity, SR-SVM excels in achieving both a high TPR and a minimal FPR. Understanding these intricacies is vital for selecting the most suitable algorithm based on the specific requirements of sentiment analysis tasks, where the balance between true positives and false positives plays a crucial role in determining the overall efficacy of the model.

Table 2. True Positive Rate and False Positive Rate

Classification Algorithms	True Positive Rate (%)	False Positive Rate (%)
CCHAN	63.228	43.791
HNN-BCA	67.318	35.412
SR-SVM	81.033	20.260

6.2. Negativity Analysis

Figure 2 and Table 3 collectively encapsulate the technical evaluation of TNR and FNR metrics across three distinct sentiment classification algorithms. This detailed scrutiny aims to uncover the algorithms' nuanced performance in correctly identifying negative instances and where they misclassify actual negatives. The graphical representation is a visual tool for discerning patterns and trends, while the tabular data provides precise numerical values essential for a quantitative understanding of algorithmic efficacy.

Intricate distinctions emerge in dissecting the TPR, a pivotal metric gauging the algorithms' precision in correctly identifying positive instances. The initial algorithm demonstrates commendable sensitivity, capturing a noteworthy fraction of positive instances. The second algorithm exhibits an enhanced TPR, indicative of improved sensitivity through the amalgamation of Hybrid Neural Network techniques and Binary Coordinate Ascent Algorithm. The third algorithm, leveraging Simpson Rule-Optimized Support Vector Machines, stands out with an impressive TPR, underscoring its proficiency in accurately identifying positive instances. This superiority in sentiment discrimination is attributed to the algorithm's capacity for optimizing hyperplanes and effectively navigating the intricacies of sentiment feature spaces.

Delving into the FPR, an indicator of the algorithms' proclivity to misclassify negative instances as positive, unveils notable intricacies. The initial algorithm exhibits a substantial FPR, implying a significant risk of erroneously labelling actual negative instances as positive. In contrast, the second algorithm demonstrates improved specificity with a lower FPR, showcasing a more discerning nature in avoiding false positives. Integrating Hybrid Neural Network techniques with Binary Coordinate Ascent Algorithm contributes to this enhanced specificity. The third algorithm maintains stringent control over false positives, substantiating its capacity for steering clear of misclassifying negative instances. This proficiency is rooted in the algorithm's adeptness at finding optimal hyperplanes by utilizing Simpson Rule optimization.

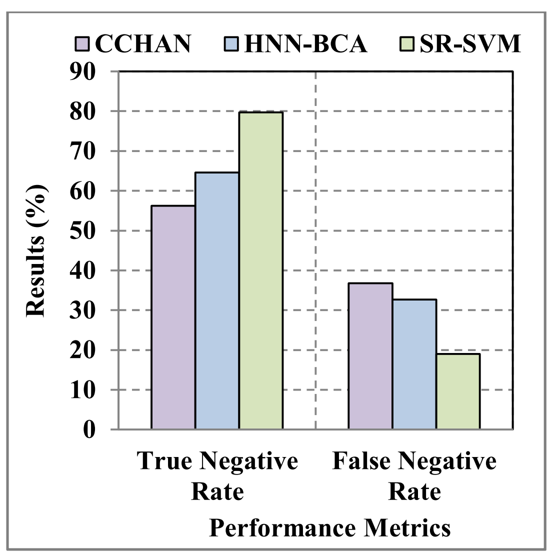

Figure 2. True Negative Rate and False Negative Rate

The technical exploration of TNR and FNR for these algorithms provides a rich tapestry of their intricate performance dynamics in sentiment classification. The interplay between true negatives and false negatives, as well as true positives and false positives, underscores the algorithms' capabilities and points towards avenues for refinement in alignment with the nuanced demands of sentiment analysis tasks.

Table 3. True Negative Rate and False Negative Rate

Classification Algorithms	True Negative Rate (%)	False Negative Rate (%)
CCHAN	56.209	36.772
HNN-BCA	64.588	32.682
SR-SVM	79.740	18.967

6.3. Classification Accuracy and F-Measure Analysis

Figure 3 and Table 4 are pivotal instruments for the technical evaluation of three distinct sentiment classification algorithms: CCHAN, HNN-BCA, and SR-SVM. These two representations offer a comprehensive insight into the algorithms' comparative performance, focusing on Classification Accuracy and F-Measure.

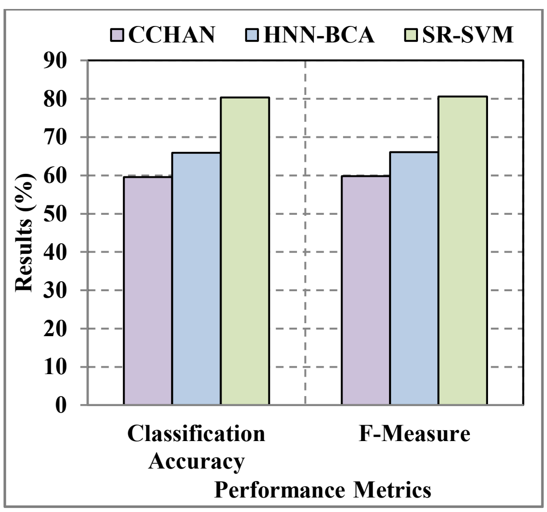

Figure 3. Classification Accuracy and F-Measure

Classification Accuracy serves as a foundational metric, gauging the overall correctness of an algorithm in classifying instances. For CCHAN, the accuracy of 59.554% implies that the model correctly classifies sentiment in nearly 59.55% of instances. This indicates a moderate level of correctness, leaving room for improvement. Moving to HNN-BCA, the algorithm demonstrates an enhanced accuracy of 65.934%, signalling a notable improvement in overall correctness compared to CCHAN. However, the standout performer in this category is SR-SVM, boasting an accuracy of 80.389%. This impressive accuracy underscores the algorithm's superior precision in sentiment classification. The notable variation in accuracy values across the algorithms indicates their differing abilities to classify sentiment correctly. The intricate workings of SR-SVM, particularly its utilization of Simpson Rule-Optimized Support Vector Machines, contribute to its heightened accuracy, showcasing the algorithm's efficacy in discerning sentiment with precision. This accuracy metric is a crucial benchmark, especially in applications where correct sentiment classification is paramount, such as customer feedback analysis or opinion mining.

F-Measure, a harmonic mean of precision and recall, provides a more nuanced evaluation, which is precious in scenarios where achieving a balance between precision and sensitivity is crucial. CCHAN's F-Measure of 59.840% suggests a relatively balanced trade-off between precision and recall. HNN-BCA advances this equilibrium with an F-Measure of 66.084%, indicating an enhanced balance between precision and recall compared to CCHAN. However, the standout performance is again demonstrated by SR-SVM, boasting an impressive F-measure of 80.565%. This underscores the algorithm's exceptional ability to balance precision and sensitivity effectively, a crucial aspect in sentiment analysis where false positives and negatives have significant implications. The F-Measure's incorporation of precision and recall offers a more comprehensive view of the algorithms' precision in sentiment classification tasks. While accuracy provides a broader overview of correctness, the F-Measure delves deeper into the nuanced interplay between true positives, false positives, and false negatives. SR-SVM's dominance in this metric suggests its adeptness at striking a delicate balance between minimizing false positives and false negatives, a testament to its robust sentiment classification capabilities.

Table 4. Classification Accuracy and F-Measure

Classification Algorithms	Classification Accuracy (%)	F-Measure (%)
CCHAN	59.554	59.840
HNN-BCA	65.934	66.084
SR-SVM	80.389	80.565

Comparing the performances of CCHAN, HNN-BCA, and SR-SVM across both Classification Accuracy and F-Measure unveils distinctive strengths and weaknesses. While demonstrating moderate correctness, CCHAN showcases a balanced trade-off between precision and recall. Incorporating Cloze Task Network and Convolutional Hierarchical Attention Networks contributes to its contextual understanding, but there is room for enhancement in precision. HNN-BCA strikes a more favourable balance with improved correctness and a commendable F-Measure, suggesting a refined equilibrium between precision and recall. The synergy of Hybrid Neural Network techniques and Binary Coordinate Ascent Algorithm contributes to its enhanced performance. However, SR-SVM emerges as the frontrunner, excelling in accuracy and F-Measure. Using Simpson Rule-Optimized Support Vector Machines allows it to navigate sentiment feature spaces with heightened precision and sensitivity.

The technical depth of this comparative analysis is pivotal for decision-makers seeking to implement sentiment classification algorithms in real-world applications. It guides the selection process based on the specific requirements of the task at hand. In scenarios where a high level of correctness is paramount, SR-SVM's superior accuracy makes it a compelling choice. HNN-BCA's refined equilibrium offers an attractive alternative when striking a balance between precision and recall is critical. While showing moderate correctness, the contextual understanding of CCHAN might find its niche in applications where a nuanced interpretation of sentiment is vital.

7. CONCLUSION

Despite the promising aspects of the Simpson Rule-Optimized Support Vector Machine (SR-SVM) for sentiment analysis in online shopping, it is essential to recognize certain shortfalls in the current work. The inherent challenges associated with interpreting highly nuanced language and context in customer reviews persist, necessitating further refinement of the SR-SVM model to comprehensively address these complexities. Additionally, the study primarily focuses on sentiment classification accuracy, leaving room for improvement in handling specific types of sentiments and expressions. Future research directions should concentrate on enhancing the robustness of SR-SVM by incorporating more sophisticated natural language processing (NLP) techniques to capture subtle nuances in customer sentiments. Exploring the integration of deep learning approaches and considering domain-specific adaptations could further bolster the model's effectiveness. Moreover, investigating the scalability and adaptability of SR-SVM across different online retail domains and platforms is essential for its broader applicability. The study's acknowledgment of shortfalls and proposed future research directions paves the way for ongoing exploration and development, emphasizing the dynamic nature of sentiment analysis in the ever-evolving landscape of online retail.

REFERENCES:

[1] M. T. Uliniansyah *et al.*, “Twitter dataset on public sentiments towards biodiversity policy in Indonesia,” *Data Br.*, vol. 52, p. 109890, 2024, doi: https://doi.org/10.1016/j.dib.2023.109890.

[2] H. Shi *et al.*, “Co-space Representation Interaction Network for multimodal sentiment analysis,” *Knowledge-Based Syst.*, vol. 283, p. 111149, 2024, doi: 10.1016/j.knosys.2023.111149.

[3] M. C. Hung, P. H. Hsia, X. J. Kuang, and S. K. Lin, “Intelligent portfolio construction via news sentiment analysis,” *Int. Rev. Econ. Financ.*, vol. 89, pp. 605–617, 2024, doi: 10.1016/j.iref.2023.07.103.

[4] Y. Zeng, W. Yan, S. Mai, and H. Hu, “Disentanglement Translation Network for multimodal sentiment analysis,” *Inf. Fusion*, vol. 102, p. 102031, 2024, doi: 10.1016/j.inffus.2023.102031.

[5] C. Gan, X. Fu, Q. Feng, Q. Zhu, Y. Cao, and Y. Zhu, “A multimodal fusion network with attention mechanisms for visual–textual sentiment analysis,” *Expert Syst. Appl.*, vol. 242, p. 122731, 2024, doi: https://doi.org/10.1016/j.eswa.2023.122731.

[6] J. Li and H.-J. Ahn, “Sensitivity of Chinese stock markets to individual investor sentiment: An analysis of Sina Weibo mood related to COVID-19,” *J. Behav. Exp. Financ.*, vol. 41, p. 100860, 2024, doi: https://doi.org/10.1016/j.jbef.2023.100860.

[7] W. Shang *et al.*, “Aspect-level sentiment analysis based on aspect-sentence graph convolution network,” *Inf. Fusion*, vol. 104, p. 102143, 2024, doi: https://doi.org/10.1016/j.inffus.2023.102143.

[8] H. Wu *et al.*, “PESI: Personalized Explanation recommendation with Sentiment Inconsistency between ratings and reviews,” *Knowledge-Based Syst.*, vol. 283, p. 111133, 2024, doi: https://doi.org/10.1016/j.knosys.2023.111133.

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

[9] J. Li, S. Qian, L. Li, Y. Guo, J. Wu, and L. Tang, "A novel secondary decomposition method for forecasting crude oil price with twitter sentiment," *Energy*, vol. 290, p. 129954, 2024, doi: https://doi.org/10.1016/j.energy.2023.129954.

[10] N. Kumar and B. R. Hanji, "Aspect-based sentiment score and star rating prediction for travel destination using Multinomial Logistic Regression with fuzzy domain ontology algorithm," *Expert Syst. Appl.*, vol. 240, p. 122493, 2024, doi: https://doi.org/10.1016/j.eswa.2023.122493.

[11] Z. Ye and X. Huang, "Adoption of a deep learning-based neural network model in the psychological behavior analysis of resident tourism consumption," *Front. Public Heal.*, vol. 10, 2022, doi: 10.3389/fpubh.2022.995828.

[12] C. Zhu, B. Yi, and L. Luo, "Base on contextual phrases with cross-correlation attention for aspect-level sentiment analysis," *Expert Syst. Appl.*, vol. 241, p. 122683, 2024, doi: 10.1016/j.eswa.2023.122683.

[13] K. Jia, F. Meng, J. Liang, and P. Gong, "Text sentiment analysis based on BERT-CBLBGA," *Comput. Electr. Eng.*, vol. 112, p. 109019, 2023, doi: 10.1016/j.compeleceng.2023.109019.

[14] G. Yi *et al.*, "VLP2MSA: Expanding vision-language pre-training to multimodal sentiment analysis," *Knowledge-Based Syst.*, vol. 283, p. 111136, 2024, doi: 10.1016/j.knosys.2023.111136.

[15] N. Pleerux and A. Nardkulpat, "Sentiment analysis of restaurant customer satisfaction during COVID-19 pandemic in Pattaya, Thailand," *Heliyon*, vol. 9, no. 11, p. e22193, 2023, doi: 10.1016/j.heliyon.2023.e22193.

[16] J. Huang *et al.*, "Dynamic hypergraph convolutional network for multimodal sentiment analysis," *Neurocomputing*, vol. 565, p. 126992, 2024, doi: 10.1016/j.neucom.2023.126992.

[17] Z. Wang, S. Hu, and W. Liu, "Product feature sentiment analysis based on GRU-CAP considering Chinese sarcasm recognition," *Expert Syst. Appl.*, vol. 241, p. 122512, 2023, doi: 10.1016/j.eswa.2023.122512.

[18] B. Seong and K. Song, "Sentiment analysis of online responses in the performing arts with large language models," *Heliyon*, vol. 9, no. 12, p. e22457, 2023, doi: 10.1016/j.heliyon.2023.e22457.

[19] K. Bu, Y. Liu, and X. Ju, "Efficient utilization of pre-trained models: A review of sentiment analysis via prompt learning," *Knowledge-Based Syst.*, vol. 283, p. 111148, 2024, doi: 10.1016/j.knosys.2023.111148.

[20] Z. Ping, G. Sang, Z. Liu, and Y. Zhang, "Aspect category sentiment analysis based on prompt-based learning with attention mechanism," *Neurocomputing*, vol. 565, p. 126994, 2024, doi: 10.1016/j.neucom.2023.126994.

[21] Q. Lu, X. Sun, Z. Gao, Y. Long, J. Feng, and H. Zhang, "Coordinated-joint translation fusion framework with sentiment-interactive graph convolutional networks for multimodal sentiment analysis," *Inf. Process. Manag.*, vol. 61, no. 1, p. 103538, 2024, doi: 10.1016/j.ipm.2023.103538.

[22] J. Serrano-Guerrero, M. Bani-Doumi, F. P. Romero, and J. A. Olivas, "A 2-tuple fuzzy linguistic model for recommending health care services grounded on aspect-based sentiment analysis," *Expert Syst. Appl.*, vol. 238, p. 122340, 2024, doi: 10.1016/j.eswa.2023.122340.

[23] J. Ramkumar, R. Vadivel, B. Narasimhan, S. Boopalan, and B. Surendren, "Gallant Ant Colony Optimized Machine Learning Framework (GACO-MLF) for Quality of Service Enhancement in Internet of Things-Based Public Cloud Networking BT - Data Science and Communication," J. M. R. S. Tavares, J. J. P. C. Rodrigues, D. Misra, and D. Bhattacherjee, Eds., Singapore: Springer Nature Singapore, 2024, pp. 425–438. doi: https://doi.org/10.1007/978.

[24] 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, 2023, doi: 10.22247/ijcna/2023/218516.

[25] 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.

[26] 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.

[27] J. Ramkumar and R. Vadivel, "Improved frog leap inspired protocol (IFLIP) – for routing in cognitive radio ad hoc networks (CRAHN),"

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

World J. Eng., vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.

[28] M. Lingaraj, T. N. Sugumar, C. S. Felix, and J. Ramkumar, “Query aware routing protocol for mobility enabled wireless sensor network,” *Int. J. Comput. Networks Appl.*, vol. 8, no. 3, pp. 258–267, 2021, doi: 10.22247/ijcna/2021/209192.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] J. Ramkumar and R. Vadivel, *CSIP—cuckoo search inspired protocol for routing in cognitive radio ad hoc networks*, vol. 556. 2017. doi: 10.1007/978-981-10-3874-7_14.

[36] 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,” in *2022 International Conference on Advanced Computing Technologies and Applications, ICACTA 2022*, 2022. doi: 10.1109/ICACTA54488.2022.9752899.

[37] 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.

[38] R. Jaganathan, V. Ramasamy, L. Mani, and N. Balakrishnan, “Diligence Eagle Optimization Protocol for Secure Routing (DEOPSR) in Cloud-Based Wireless Sensor Network,” 2022, doi: 10.21203/rs.3.rs-1759040/v1.

[39] 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.

[40] 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*, K. Murari, N. Prasad Padhy, and S. Kamalasadan, Eds., Singapore: Springer Nature Singapore, 2023, pp. 17–27. doi: 10.1007/978-981-19-8353-5_2.

[41] 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.

[42] T. Manshu and Z. Xuemin, “CCHAN: An End to End Model for Cross Domain Sentiment Classification,” *IEEE Access*, vol. 7, pp. 50232–50239, 2019, doi: 10.1109/ACCESS.2019.2910300.