Journal of Theoretical and Applied Information Technology
31st March 2024. Vol.102. No 6
© Little Lion Scientific

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

OPTIMIZING SHOPPING EXPERIENCES: BAYESIAN OPTIMIZATION-INSPIRED RANDOM FOREST FOR ENHANCED SENTIMENT ANALYSIS

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

This article investigates the optimization of sentiment analysis within online shopping. It begins by outlining the inherent challenges in sentiment analysis, particularly in handling large volumes of unstructured textual data and achieving accurate results amidst noise and context complexities. In response to these challenges, the proposed work introduces the utilization of Bayesian Optimization-inspired Random Forest (BO-RF) as a solution. This innovative approach aims to enhance sentiment analysis accuracy and efficiency by leveraging the combined strengths of Bayesian Optimization and Random Forest algorithms. The mechanism of the proposed work involves leveraging Bayesian Optimization to tune the hyperparameters of the Random Forest model efficiently. This process optimizes the model's performance, improving the accuracy of sentiment analysis tasks. BO-RF excels in feature selection, enabling the extraction of relevant information from text data while filtering out noise. Through comprehensive experiments and evaluations, the results demonstrate the superior performance of BO-RF compared to traditional sentiment analysis methods. The proposed approach achieves higher accuracy rates and greater efficiency in sentiment classification tasks, showcasing its potential to revolutionize sentiment analysis within online shopping.

Keywords: *Sentiment Analysis, Bayesian Optimization, Random Forest, Online Shopping, Accuracy*

1. INTRODUCTION

Customer reviews play a pivotal role in shaping the online shopping experience. They provide valuable information beyond product descriptions, offering real-world perspectives from individuals who have already purchased. Positive reviews can instil confidence in potential buyers, while negative reviews can serve as warnings, guiding consumers away from potential pitfalls [1]. The transparency facilitated by customer reviews fosters a sense of trust between buyers and sellers, creating a virtual marketplace where informed decisions can be made. In addition to influencing individual purchasing decisions, customer reviews contribute to the overall reputation of products and sellers on online platforms [2], [3]. A high volume of positive reviews can enhance a product's visibility and credibility, ultimately driving sales. Managing and analyzing this influx of customer feedback presents its own set of challenges [4].

In the intricate landscape of business strategy, sentiment analysis serves as a sophisticated instrument, allowing companies to gain nuanced insights into customer perceptions. By scrutinizing diverse sources such as online reviews, social media interactions, and survey responses, organizations can distil valuable information regarding customer satisfaction, pinpoint areas for enhancement, and guide their decision-making with a data-centric approach [5]. However, the utility of sentiment analysis extends beyond its diagnostic capabilities, playing a pivotal role in brand management. In brand management, sentiment analysis becomes a proactive force [6]. Companies leverage this analytical tool to swiftly identify and address negative sentiments and strategically harness positive feedback to fortify their reputation in a fiercely competitive market. The ability to navigate the complex interplay of positive and negative sentiments allows businesses to mould their public image intentionally, fostering a more favourable perception among their target audience [7].

At the heart of sentiment analysis lies the indispensable role of machine learning, mainly through the avenue of supervised learning. This facet enables models to grasp the nuanced contextual intricacies inherent in language, navigating through layers of sarcasm, cultural subtleties, and diverse expressions with remarkable finesse [8]. Incorporating unsupervised learning techniques like clustering and topic modelling further refine sentiment analysis capabilities. By discerning intricate patterns within unlabeled data, these techniques contribute to a more comprehensive understanding of the sentiment landscape [9], [10]. The synergy between machine learning and sentiment analysis is a testament to technological sophistication. This collaboration amplifies the precision of calculations and fortifies adaptability, ensuring that these analytical tools remain finely attuned to the ever-evolving dynamics of the digital communication landscape. In essence, the intersection of technology and human perception within sentiment analysis gives businesses a resilient and proactive framework for navigating the multifaceted currents of public opinion [11].

1.1. Problem Statement

The challenge of contextual ambiguity and sarcasm complicates sentiment analysis, introducing uncertainties in accurately deciphering the sentiments embedded in user-generated content. Existing models often need to improve understanding of the subtle nuances and figurative language, resulting in misinterpretations. This problem significantly impacts the reliability of sentiment predictions, particularly in reviews where sarcasm and contextual intricacies are prevalent. Developing advanced natural language processing techniques to unravel the layers of linguistic complexity is imperative for overcoming this challenge and ensuring precise sentiment analysis in the face of contextual ambiguity and sarcasm.

1.2. Motivation

The motivation for investigating contextual ambiguity and sarcasm in sentiment analysis arises from the imperative to enhance the precision of sentiment predictions in user-generated content. Current models often need help deciphering subtle nuances and figurative language, leading to misinterpretations and compromised reliability. This research seeks to develop advanced natural language processing techniques that can unravel the layers of linguistic complexity, enabling sentiment analysis models to comprehend and accurately interpret contextual intricacies and sarcasm. The motivation stems from the aspiration to improve the practical applicability of sentiment analysis in real-world scenarios, where user expressions often entail diverse contextual nuances and figurative elements.

1.3. Research Objective

This research addresses contextual ambiguity and sarcasm in sentiment analysis by proposing and evaluating the Bayesian Optimization-based Random Forest (BO-RF) algorithm. The research objective is to enhance the precision of sentiment predictions in user-generated content, where existing models often need to improve in deciphering subtle nuances and figurative language. By leveraging the capabilities of BO-RF, we seek to develop advanced natural language processing techniques that can unravel the layers of linguistic complexity, enabling sentiment analysis models to interpret contextual intricacies and sarcasm accurately, thus improving the practical applicability of sentiment analysis in real-world scenarios.

2. LITERATURE REVIEW

"Masked Language Modelling" [12] leverages the techniques from masked language modelling. The model predicts affective tokens within the context of ABSA, allowing for a more nuanced understanding of sentiment expressions associated with specific aspects. It can capture the intricacies of sentiment nuances in diverse contexts. "Lexicon-based Unsupervised Sentiment Analysis" [13] focuses on adapting negation instances. By incorporating negation adjustments, this model refines its accuracy in discerning the sentiment associated with equipment health status, particularly in cases where negation impacts the overall sentiment context. It is a tailored sentiment analysis technique that caters to the specifics of industrial equipment monitoring, contributing to enhanced accuracy in health status assessments. "Spanish Financial Stability Report" [14] is proposed for applying sentiment analysis techniques to the financial domain, offering insights into sentiments prevalent in financial stability reports written in Spanish. Addressing the unique language of economic discourse provides a more accurate understanding of sentiment expressions, contributing to a nuanced comprehension of sentiments within financial stability reports in Spanish.

"Aspect-Based Sentiment Analysis (ABSA)" [15] is the creation of a comprehensive and domain-specific dataset that facilitates the training and evaluation of ABSA models. It includes diverse user reviews annotated with aspect-level sentiments, offering a valuable resource for developing, refining, and benchmarking ABSA algorithms. "Pseudo Dense Counterfactual Augmentation" [16] designed to enhance ABSA. This approach addresses the challenge of data sparsity and introduces a mechanism for capturing diverse sentiments. It contributes to the improved generalization and robustness of ABSA models, ensuring more effective sentiment analysis in scenarios with limited labelled data. The "Bayesian Sentiment Analysis" [17] approach addresses the challenges of cold start and sparsity in ranking-based recommender systems.

It focuses on the recommendation process, enhancing adaptability to scenarios with limited data. It mitigates sparsity issues in ranking-based recommender systems. It represents an advancement in recommender system research, offering a tailored solution to limited user interactions.

"Big Data Sentiment Analysis" [18] is proposed for Large-scale Text Processing (LTP) that aims for effective text classification. It enables a deep analysis of sentiments within the business environment. It focuses on public perception of business dynamics in Heilongjiang Province, showcasing the applicability of advanced text classification techniques in discerning sentiments at a large scale. "Prompt Semantic Augmented Network (PSAN)" [19] incorporates prompt semantic augmentation by enhancing its ability to understand and analyze sentiments associated with specific aspects. PSAN leverages to augment semantic understanding, contributing to a more nuanced and context-aware sentiment analysis. It provides the importance of semantic augmentation for improved accuracy in dissecting sentiments across various aspects. "Online Medical Reviews" [20] focuses on constructing an ABSA model aiming to meticulously design a model capable of discerning sentiments associated with specific elements within the complex domain of medical reviews. The intricacies of medical language and diverse aspects within studies offer a better understanding of sentiments, providing valuable insights for healthcare professionals and stakeholders.

"Multi-Information Fusion and Interaction Neural Network" [21] is proposed to allow the model to fuse diverse sources of information effectively for a comprehensive sentiment analysis. This neural network model addresses the challenge of capturing sentiments across various aspects by dynamically interacting with multiple information channels, contributing to enhanced accuracy. The importance of synthesizing diverse information is deeply analyzed for better accuracy. "Text-Guided Multi-Task Learning Network" [22] focus on efficient analysis of sentiments across multiple modalities, providing a cohesive understanding of sentiments in diverse data sources. It narrows to multimodal sentiment analysis by introducing a network architecture that optimally utilizes textual guidance to improve sentiment understanding across multiple tasks. "Disentanglement Translation Network" [23] facilitates effective translation and sentiment analysis across different modalities. This model enhances its ability to capture the unique characteristics of each modality, contributing to a more refined understanding of sentiments in multimodal data. It represents a significant advancement in the quest for sophisticated multimodal sentiment analysis methodologies.

"Cloze Task Network based Convolutional Hierarchical Attention Networks (CCHAN)" [24] elevates the model's capacity for adaptive sentiment understanding. Unlike static embeddings that maintain a fixed representation for each word, dynamic embeddings in CCHAN adjust in real-time based on the surrounding context within sentences. It is pivotal in scenarios where sentiment subtly evolves within the confines of the same document or review. By dynamically refining contextual embeddings, CCHAN excels at capturing the nuanced changes in sentiment expression, resulting in a model that is accurate and highly sensitive to the intricacies of sentiment shifts. This feature makes CCHAN an invaluable tool for applications demanding fine-grained sentiment analysis in dynamic linguistic environments, ensuring its efficacy in capturing the subtle nuances inherent in various forms of textual communication. Bio-inspired optimization is also applied in multiple domain to attain better results [25], [26], [35]–[44], [27]–[34].

"Hybrid Neural Network techniques using Binary Coordinate Ascent Algorithm (HNN-BCA)" [45] extends its prowess beyond static evaluations by incorporating a robust temporal analysis approach. Unlike traditional sentiment analysis models, HNN-BCA considers the dynamic nature of sentiment expression over time. The HNN architecture is augmented with temporal features, allowing the model to capture evolving sentiments in a sequence of text. The BCA Algorithm complements this by optimizing the model's secular learning, ensuring it adapts effectively to changing sentiment patterns. This temporal dimension adds a new layer of sophistication to sentiment analysis, enabling applications to track sentiment trends, event-specific reactions, and dynamic social media discourse.

3. BAYESIAN OPTIMIZATION RANDOM FOREST (BA-RF)

Bayesian Optimization-inspired Random Forest (BO-RF) merges the strengths of two powerful techniques: Bayesian optimization and random forests. This hybrid approach combines the robustness of random forests in handling complex datasets with the efficiency and adaptability of Bayesian optimization in hyperparameter tuning. BO-RF optimizes the hyperparameters of the random forest algorithm using a Bayesian optimization framework, allowing for intelligent exploration of the hyperparameter space to maximize model performance. BO-RF converges to an optimal solution more efficiently than traditional random search methods by iteratively selecting hyperparameters based on their expected performance. The Bayesian optimization component allows BO-RF to learn from previous iterations, adapt its search strategy, and allocate computational resources more effectively, leading to faster convergence and improved model accuracy. This approach is particularly beneficial for tasks where model performance is critical, such as predictive modeling, classification, and regression tasks in various domains.

3.1. Initialization of Random Forest Hyperparameters

The initialization of hyperparameters in the BO-RF algorithm is crucial to set the foundation for subsequent optimization. The Random Forest algorithm is characterized by hyperparameters denoted as $\Theta = \{\theta_1, \theta_2, \dots, \theta_k\}$, where k represents the number of hyperparameters. These hyperparameters include the number of trees (N_{trees}), maximum depth of each tree (D_{max}), and other relevant settings. The initial configuration is represented as Θ_0, where $\Theta_0 = \{\theta_{1_0}, \theta_{2_0}, \dots, \theta_{k_0}\}$.

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_k\} \quad (1)$$

$$\Theta_0 = \{\theta_{1_0}, \theta_{2_0}, \dots, \theta_{k_0}\} \quad (2)$$

Define the objective function $f(\Theta)$ that encapsulates the performance metric to be optimized, such as classification accuracy or mean squared error in a regression task.

$$f(\Theta) = Performance\ Metric \quad (3)$$

Select a Gaussian process (GP) as the surrogate model to approximate the unknown objective function $f(\Theta)$. The goal function and its uncertainty may be approximated probabilistically using the surrogate model.

$$GP\big(\mu(\Theta), \Sigma(\Theta)\big) \quad (4)$$

where $\mu(\Theta)$ is the mean function and $\Sigma(\Theta)$ is the covariance function of the Gaussian process.

Algorithm 1: Initialization of Random Forest Hyperparameters

```
Input:
  • $\Theta$: Set of Random Forest hyperparameters.
  • $k$: Number of hyperparameters.

Output:
  • $\Theta_0$ : Initial configuration of Random Forest hyperparameters.

Procedures:
 Step 1: Initialize Hyperparameters($\boldsymbol{\Theta}, \boldsymbol{k}$):
     a. $N_{trees}, D_{max}$, and other hyperparameters are elements of $\Theta$.
     b. $N_{trees_0}, D_{max_0}$, and other initial hyperparameter values are elements of $\Theta_0$.
     c. $\Theta_0 = \Theta$ (Copy initial hyperparameter values).
 Step 2: Define Objective Function():
     a. $f(\Theta)$ represents the objective function.
     b. Performance metric is calculated based on $\Theta$.
 Step 3: Select Surrogate Model():
     a. Gaussian process $GP$ is chosen as the surrogate model.
     b. $\mu(\Theta)$ and $\Sigma(\Theta)$ denote the mean and covariance functions of the Gaussian process, respectively.
```

3.2. Surrogate Modeling with Gaussian Process

The second step in the BO-RF algorithm involves selecting a surrogate model to approximate the unknown objective function using a Gaussian process (GP). This process aims to capture the underlying performance landscape of the Random Forest hyperparameters. To begin, this research initializes the surrogate model by defining the Gaussian process GP with its mean function $\mu(\Theta)$ and covariance function $\Sigma(\Theta)$.

$$GP\big(\mu(\Theta), \Sigma(\Theta)\big) \quad (5)$$

where Θ represents the Random Forest hyperparameters. This Gaussian process serves as a probabilistic model, providing insights into the behaviour of the objective function and accounting for uncertainties in the performance landscape.

The surrogate model aims to map the Random Forest hyperparameters Θ to the objective function $f(\Theta)$. This mapping is established through the mean function $\mu(\Theta)$, representing the expected value of the objective function.

$$f(\Theta) = \mu(\Theta) \quad (6)$$

Eq.(6) is a predictive tool estimating the objective function's behaviour across the hyperparameter space.

In parallel, the covariance function $\Sigma(\Theta)$ characterizes the relationships between different hyperparameter configurations. It captures the uncertainty associated with the objective function's predictions and shapes the Gaussian process.

$$\Sigma(\Theta) \quad (7)$$

Eq.(7) encodes how changes in one set of hyperparameters correlate with changes in another, providing a measure of similarity or dissimilarity.

The initialization of the surrogate model at this stage is crucial. The initial set of Random Forest hyperparameters Θ_0 from the Eq.(2) influences the starting configuration of the Gaussian process. It sets the tone for subsequent Bayesian Optimization iterations, impacting the efficiency of the search for optimal hyperparameters.

Algorithm 2: Surrogate Modeling with Gaussian Process

```
Input:
    • $\Theta$: Random Forest hyperparameters.
    • $f(\Theta)$: Objective function.
    • $k$: Number of hyperparameters.

Output:
    • $GP(\mu(\Theta), \Sigma(\Theta))$: Gaussian process surrogate model.

Procedure:
  Step 1: InitializeSurrogateModel ($\boldsymbol{\Theta}, \boldsymbol{f(\Theta)}, \boldsymbol{k}$):
      a. Create an empty Gaussian process $GP$.
      b. Define the mean function $\mu(\Theta)$ and covariance function $\Sigma(\Theta)$ for the Gaussian process.
  Step 2: MapObjectiveFunction():
      a. Utilize the surrogate model to map the Random Forest hyperparameters $\Theta$ to the objective function $f(\Theta)$.
      b. Establish $f(\Theta) = \mu(\Theta)$ through the mean function.
  Step 3: CaptureCovarianceStructure():
      a. Capture the covariance structure $\Sigma(\Theta)$ within the Gaussian process.
      b. The covariance structure characterizes relationships between different hyperparameter configurations.
  Step 4: ImpactofInitialization():
      a. Acknowledge the impact of the initial set of hyperparameters $\Theta_0$ on the surrogate model.
      b. The initial configuration influences subsequent iterations in the Bayesian Optimization process.
```

Algorithm 2 outlines the steps involved in surrogate modelling with a Gaussian process within the context of the BA-RF algorithm. It emphasizes the initialization of the surrogate model, mapping the objective function, and capturing covariance structures to provide a probabilistic estimate of the objective function's behaviour.

3.3. Capturing Covariance Structure in Gaussian Process

In the BA-RF algorithm, the third step involves capturing the covariance structure within the Gaussian process surrogate model. This crucial step enhances our understanding of how changes in one set of Random Forest hyperparameters correlate with changes in another, shaping the probabilistic model's predictive capability.

The covariance structure, denoted as $\Sigma(\Theta)$, is a fundamental component of the Gaussian process (GP) surrogate model. It characterizes the relationships between different configurations of Random Forest hyperparameters. Specifically, the covariance matrix $\Sigma(\Theta)$ quantifies the degree of similarity or dissimilarity between pairs of hyperparameter sets.

$$\Sigma(\Theta) \quad (8)$$

This matrix provides insights into how variations in one hyperparameter impact the objective function in conjunction with variations in other hyperparameters. The surrogate model utilizes this information to predict the objective function's behaviour across the hyperparameter space.

The elements of the covariance matrix $\Sigma(\Theta)$ are determined by the choice of the covariance function. Each element $\sum_{ij}(\Theta)$ represents the covariance between the i-th and j-th hyperparameters. The diagonal elements $\sum_{ii}(\Theta)$ indicate the variance of the i-th hyperparameter.

$$\sum_{ii}(\Theta) \tag{9}$$

This matrix encapsulates the interdependence among hyperparameters, contributing to the model's understanding of the objective function landscape.

The covariance structure also encodes information about the relationships between hyperparameters. Positive covariance $(\Sigma_{ij} > 0)$ signifies a positive correlation, suggesting that increases in one hyperparameter will likely coincide with increases in another. Conversely, negative covariance $(\Sigma_{ij} < 0)$ indicates a negative correlation.

$$\begin{aligned}&Covariance\ Relationship\colon \Sigma_{ij}\ is\\&> 0\ (positive\ correlation)\\&Covariance\ Relationship\colon \Sigma_{ij}\ is\\&< 0\ (negative\ correlation)\end{aligned} \tag{10}$$

Understanding these relationships is crucial in guiding the search for optimal hyperparameters during the subsequent Bayesian Optimization iterations.

Algorithm 3: Capturing Covariance Structure in Gaussian Process

```
Input:
    • $\Theta$: Random Forest hyperparameters.
    • $k$: Number of hyperparameters.

Output:
    • $\Sigma(\Theta)$: Covariance matrix capturing relationships between hyperparameters.

Procedure:
  Step 1: CaptureCovarianceStructure($\boldsymbol{\Theta}, \boldsymbol{k}$):
      a. Initialize an empty covariance matrix $\Sigma(\Theta)$ of size $k \times k$.
      b. Define the covariance function to determine relationships between hyperparameters.
  Step 2: DetermineCovarianceElements():
      a. Iterate through each pair of hyperparameters, calculating $\sum_{ij}(\Theta)$ for the covariance matrix.
      b. Diagonal elements $(\sum_{ii}(\Theta))$ represent the variance of each hyperparameter.
  Step 3: PositiveCovarianceRelationship():
      a. Identify positive covariance relationships $(\Sigma_{ij} > 0)$between pairs of hyperparameters.
      b. Positive covariance signifies a positive correlation, suggesting concurrent increases in hyperparameter values.
  Step 4: NegativeCovarianceRelationship():
      a. Identify negative covariance relationships $(\Sigma_{ij} < 0)$ between pairs of hyperparameters.
      b. Negative covariance signifies a negative correlation, indicating concurrent decreases or increases in hyperparameter values.
  Step 5: OutputCovarianceMatrix():
      a. Return the covariance matrix $\Sigma(\Theta)$ capturing relationships between Random Forest hyperparameters.
```

Algorithm 3 captures the covariance structure within the Gaussian process surrogate model. The procedure calculates covariance elements, identifies relationships, and produces a covariance matrix that informs the probabilistic modelling of the objective function landscape.

3.4. Bayesian Optimization Iterations

This step builds on the surrogate model established in earlier steps, leveraging the probabilistic insights gained from the Gaussian process to make informed decisions about where to evaluate the objective function next.

The acquisition function, denoted as $A(\Theta)$, plays a pivotal role in the iterative optimization process. It acts as a heuristic to balance exploration and exploitation. Several standard acquisition functions exist, such as Expected Improvement (EI), Probability of Improvement (PI), and Upper Confidence Bound (UCB). The selection of the acquisition function is a critical aspect of Bayesian Optimization, influencing the algorithm's exploration strategy.

$$A(\Theta) \tag{12}$$

The acquisition function guides the selection of the next set of Random Forest hyperparameters to evaluate the objective function. By maximizing the acquisition function, the algorithm identifies promising regions in the hyperparameter space. The chosen hyperparameter configuration is denoted as Θ_{next}.

$$\Theta_{next} = \arg max\, A(\Theta) \tag{13}$$

The algorithm evaluates the objective function at the selected hyperparameter configuration Θ_{next} . This involves computing the performance metric for the Random Forest model with the chosen hyperparameters.

$$f(\Theta_{next}) = Objective\ Function\ Evaluation \tag{14}$$

The surrogate model is updated with the new observation following the objective function evaluation. The updated Gaussian process now incorporates information about the objective function's behaviour at the chosen hyperparameter configuration.

$$GP\big(\mu(\Theta), \Sigma(\Theta)\big) \tag{15}$$

The surrogate model is updated iteratively as the objective function is evaluated and hyperparameters are selected until a stopping criteria, such as convergence or a preset number of iterations, is reached.

```
Algorithm 4. Bayesian Optimization Iterations
Input:
    • $A(\Theta)$: Acquisition function.
    • $\Theta$: Random Forest hyperparameters.
    • $f(\Theta)$: Objective function.
    • $k$: Number of hyperparameters.
    • Stopping criterion.

Output:
    • $\Theta_{next}$ : Hyperparameter configuration.
    • $f(\Theta_{next})$: Objective function evaluation.
    • Updated surrogate model $GP\big(\mu(\Theta), \Sigma(\Theta)\big)$.

Procedure:
Step 1: InitializeBayesianOptimization
        $(A(\Theta), \Theta, f(\Theta), k, Stopping\ Criterio$
        :
        a. Set initial hyperparameters $\Theta_0$ based on the surrogate model initialization.
        b. Initialize surrogate model $GP$ with $\Theta_0$ and $f(\Theta_0)$.
        c. Set iteration counter $t = 0$.
Step 2: IterativeOptimization():
        a. While the stopping criterion is not met:
            • Calculate acquisition function $A(\Theta_t)$ using the surrogate model.
            • Identify the next hyperparameter configuration $\Theta_{next} = \arg\max A(\Theta_t)$ .
            • Evaluate the objective function at $\Theta_{next}$: $f(\Theta_{next})$.
            • Update the surrogate model $GP$ with the new observation $\big(\Theta_{next}, f(\Theta_{next})\big)$.
            • Increment iteration counter $t = t + 1$.
```

Algorithm 4 outlines the iterative optimization process in Bayesian Optimization, emphasizing the role of the acquisition function in guiding the exploration-exploitation trade-off. The procedure iteratively refines the hyperparameter configuration based on the surrogate model's probabilistic insights until the stopping criterion is met.

3.5. Updating the Surrogate Model

In BO-RF algorithm, the fifth step involves updating the surrogate model after each iteration. This crucial step ensures that the probabilistic model accurately represents the observed behaviour of the objective function at the chosen hyperparameter configurations.

Upon the evaluation of the objective function at the selected hyperparameter configuration Θ_{next}, the surrogate model is updated to incorporate this new observation. The Gaussian process (GP) is refined to reflect better the true performance landscape of the Random Forest model.

$$GP\big(\mu(\Theta), \Sigma(\Theta)\big) \tag{16}$$

The updated surrogate model is based on an augmented dataset that includes the previous observations $(\Theta_i, f(\Theta_i))$ and the most recent evaluation $(\Theta_{next}, f(\Theta_{next}))$.

$$\{(\Theta_i, f(\Theta_i)) | i = 1,2, \dots t - 1, (\Theta_{next}, f(\Theta_{next}))\} \quad (17)$$

The mean function $\mu(\Theta)$ of the Gaussian process is recalculated based on the augmented dataset. It represents the expected value of the objective function at any given hyperparameter configuration.

$$\mu(\Theta) \quad (18)$$

Simultaneously, the covariance function $\Sigma(\Theta)$ is adjusted for the new observation. The covariance structure characterizes the relationships between different hyperparameter configurations and plays a crucial role in guiding the exploration of the hyperparameter space.

$$\Sigma(\Theta) \quad (19)$$

Algorithm 5 captures the continuous refinement of the Gaussian process, ensuring it accurately represents the observed behaviour of the objective function at different hyperparameter configurations.

```
Algorithm 5. Updating the Surrogate Model

Input:
    • $GP(\mu(\Theta), \Sigma(\Theta))$: Surrogate model with mean and covariance functions.
    • $\Theta_{next}$: Newly evaluated hyperparameter configuration.
    • $f(\Theta_{next})$: Objective function value at $\Theta_{next}$.
    • Dataset: Existing dataset with hyperparameter configurations and corresponding objective function values.

Output:
    • Updated surrogate model $GP(\mu(\Theta), \Sigma(\Theta))$ reflecting the new observation.

Procedure:
  Step 1: UpdateSurrogateModel $(\boldsymbol{GP(\mu(\Theta), \Sigma(\Theta)), \Theta_{next}, f(\Theta_{next})})$
        a. Augment the dataset with the new observation: $Dataset \leftarrow Dataset \cup \{(\Theta_{next}, f(\Theta_{next}))\}$.
        b. Recalculate the mean function $\mu(\Theta)$ based on the augmented dataset.
        c. Adjust the covariance structure $\Sigma(\Theta)$ to account for the new observation.
        d. Return the updated surrogate model $GP(\mu(\Theta), \Sigma(\Theta))$ reflecting the refined understanding of the objective function landscape.
```

3.6. Out-Of-Bag (OOB) Error Estimation

OOB leverages the inherent nature of the Random Forest algorithm, which constructs multiple decision trees based on bootstrapped subsets, to estimate the model's performance without requiring a separate validation set.

Before delving into OOB error estimation, it is essential to understand the training process of the Random Forest. The ensemble of decision trees, denoted as $T_1, T_2, \dots, T_N$, is constructed based on bootstrapped subsets of the original dataset. Each tree is trained with a subset of the data, introducing diversity into the ensemble.

$$T_1, T_2, \dots, T_N \text{ are decision trees} \quad (20)$$

During the construction of each decision tree T_i, a subset of samples is left out, forming the OOB samples. These OOB samples are denoted as D_{OOB} and consist of data points not included in the bootstrap sample for the i-th tree.

$$D_{OOB_i} = Out - of - bag\ samples\ for\ T_i \quad (21)$$

The OOB samples are then used to obtain predictions from each decision tree T_i. Let y_{pred_i} represent the predicted values on the OOB samples by the i-th tree.

$$y_{pred_i} = T_i(\text{OOB samples}) \quad (22)$$

The predictions from individual trees are aggregated to obtain the final ensemble prediction on the OOB samples. For classification tasks, a majority vote is often employed, while for regression tasks, the predictions are averaged.

$$Final\ OOB\ Prediction = \frac{1}{N}\sum_{i=1}^{N} y_{pred_i} \quad (23)$$

The utilization of OOB samples for error estimation aligns with the overarching goal of understanding the model's performance. The updated surrogate model from Section 3.5 enhances the accuracy of predictions, contributing to a more reliable OOB error estimate. This step ensures a robust evaluation of the Random Forest model's performance, facilitating informed decisions in the subsequent steps of the BO-RF algorithm. This mathematical description elucidates the integral role of OOB error estimation in the BA-RF algorithm, bridging the gap between the training process and the overall evaluation of the model's effectiveness.

```
Algorithm 6. Out-of-bag (OOB) Error Estimation
Input:
    • Random Forest ensemble $T_1, T_2, \dots, T_N$.
    • Original dataset.
    • Training process details.
    • Decision tree construction details.

Output:
    • OOB predictions.
    • Aggregated OOB prediction.

Procedure:
  Step 1: RandomForestTraining():
      a. For each decision tree $T_i$ in the ensemble:
          • Construct a bootstrap sample from the original dataset.
          • Train $T_i$ with the bootstrap sample.
          • Identify the OOB samples $D_{OOB_i}$ left out during the construction of $T_i$.
          • Predict values $y_{pred_i}$ on the OOB samples using $T_i$.
  Step 2: AggregateOOBPredictions():
      a. Aggregate individual OOB predictions $y_{pred_i}$ from all decision trees.
      b. Use a majority vote for classification tasks to obtain the final OOB prediction.
      c. For regression tasks, average the OOB predictions to obtain the final ensemble prediction.
```

Algorithm 6 outlines the steps involved in the OOB error estimation process within the context of the BA-RF algorithm. It details the training of the Random Forest ensemble and the subsequent aggregation of OOB predictions for a robust evaluation of model performance.

3.7. Bayesian Optimization Convergence Criterion

In the BO-RF algorithm, the seventh step involves establishing a convergence criterion to determine when to conclude the optimization process. This criterion is essential for preventing unnecessary iterations, ensuring computational efficiency, and providing a reliable solution for the identified hyperparameters.

The convergence criterion often relies on monitoring the behaviour of the objective function value over successive iterations. Let f_t denote the objective function value at iteration t, and f_{t-1} represent the objective function value at the previous iteration. The optimization process is considered to have converged when the change in the objective function value is below a predefined threshold ϵ.

$$|f_t - f_{t-1}| \leq \epsilon \quad (24)$$

Convergence can be defined based on the stability of the identified hyperparameter configuration. Let Θ_t represent the hyperparameter configuration at iteration t, and Θ_{t-1} denote the hyperparameter configuration at the previous iteration. Convergence is achieved when the hyperparameter configurations remain stable over iterations.

$$\Theta_t = \Theta_{t-1} \quad (25)$$

To ensure computational efficiency, a predefined maximum number of iterations (T_{max}) can be established. The optimization process concludes when the number of iterations reaches this limit.

$$t \geq T_{max} \quad (26)$$

Algorithm 7 involves assessing convergence in the BA-RF algorithm. It provides mechanisms to determine convergence based on changes in the objective function values, stability of hyperparameter configurations, and adherence to a predefined iteration limit. The convergence decision is a critical aspect of the overall optimization process.

Algorithm 7: Bayesian Optimization Convergence Criterion

```
Input:
    • Objective function values $f_t$.
    • Hyperparameter configurations $\Theta_t$.
    • Maximum number of iterations $T_{max}$
    • Convergence threshold $\epsilon$.

Output:
    • Convergence decision.

Procedure:
  Step 1: ObjectiveFunctionConvergence ($\boldsymbol{f_t}, \boldsymbol{\epsilon}$):
      a. Initialize $t = 1$ as the first iteration.
      b. While $t > 1$:
          • Check if $|f_t - f_{t-1}| \leq \epsilon$.
          • If true, consider convergence achieved.
          • If false, continue to the next iteration.
  Step 2: HyperparameterConfigurationConvergence($\boldsymbol{\Theta_t}$):
      a. Initialize $t = 1$ as the first iteration.
      b. While $t > 1$:
          • Check if $\Theta_t = \Theta_{t-1}$.
          • If true, consider convergence achieved.
          • If false, continue to the next iteration.
  Step 3: IterationLimit( $\boldsymbol{t}, \boldsymbol{T_{max}}$):
      a. Check if $t \geq T_{max}$.
      b. If true, consider convergence achieved.
      c. If false, continue to the next iteration.
  Step 4: ConvergenceDecision():
      a. Return the decision based on the convergence criteria.
      b. If any of the convergence criteria are met, consider convergence achieved.
      c. If none of the criteria are met, continue the optimization process.
```

3.8. Optimal Hyperparameter Selection

This step marks the culmination of the algorithm, providing a refined set of hyperparameters that yield optimal performance for the given task. The primary goal is to optimize the objective function $f(\Theta)$ for the Random Forest hyperparameters Θ. The identified optimal hyperparameters, denoted as Θ^*, maximize or minimize the objective function, depending on the nature of the optimization task.

The surrogate model, represented by the Gaussian process (GP), has been continuously refined throughout the Bayesian Optimization iterations. The optimal hyperparameters are often selected based on the mean function $\mu(\Theta)$ of the Gaussian process, representing the expected value of the objective function.

$$\Theta^* = arg\ \max \mu(\Theta) \qquad (27)$$

While the mean function guides the selection of the optimal hyperparameters, it is essential to consider the uncertainty associated with the predictions. The covariance function $\Sigma(\Theta)$ measures uncertainty, allowing for a more informed decision.

$$Incorporate\ uncertainty : \Theta^* = arg\ \max\big(\mu(\Theta) + k.\sigma(\Theta)\big) \qquad (28)$$

where, $\sigma(\Theta)$ represents the standard deviation associated with the Gaussian process predictions, and κ is a user-defined parameter that balances exploration and exploitation.

Optimal Hyperparameter Selection elucidates the mathematical framework for selecting optimal hyperparameters in the BO-RF algorithm. Incorporating uncertainty provides a nuanced approach to decision-making, ensuring a balance between exploiting known promising regions and exploring potentially superior areas in the hyperparameter space.

Algorithm 8: Optimal Hyperparameter Selection

```
Input:
    • Surrogate model $GP(\mu(\Theta), \Sigma(\Theta))$ with mean and covariance functions.
    • Exploration-exploitation parameter $\kappa$.

Output:
    • Optimal hyperparameters $\Theta^*$maximizing the objective function.
```

```
Procedure:
  Step 1: SelectOptimalHyperparameters $(GP(\mu(\Theta), \Sigma(\Theta)), \kappa)$:
      a. Calculate the mean function $\mu(\Theta)$ based on the surrogate model.
      b. Calculate the standard deviation function $\sigma(\Theta)$ based on the surrogate model.
      c. Incorporate uncertainty into the selection process: $\Theta^* = \arg\max\big(\mu(\Theta) + \kappa \cdot \sigma(\Theta)\big)$.
      d. d. Return the optimal hyperparameters $\Theta^*$.
```

Algorithm 8 outlines the process for selecting optimal hyperparameters in the BA-RF algorithm. It utilizes the refined surrogate model to balance exploration and exploitation, considering both the mean function and the uncertainty captured by the standard deviation. The result is a set of hyperparameters that maximize the objective function within the defined constraints.

3.9. Model Retraining with Optimal Hyperparameters

The BO-RF algorithm involves retraining the Random Forest model using the optimal hyperparameters identified in Section 3.8. This step ensures the model achieves its maximum potential performance by utilizing the refined set of hyperparameters gleaned through the Bayesian Optimization process.

The optimal hyperparameters, denoted as Θ^* , are the result of the Bayesian Optimization convergence and selection process in Step 8. These hyperparameters represent the configuration that maximizes or minimizes the objective function, embodying the refined understanding of the Random Forest model's behaviour.

$$\Theta^* = f(\Theta^*) = max\ or\ min\ f(\Theta) \quad (29)$$

With the optimal hyperparameters in hand, the Random Forest model is retrained using the entire original dataset. This step contrasts with the iterative optimization process, where the model was trained on bootstrapped subsets. Retraining the full dataset ensures that the model benefits from all the available information and generalizes well to new, unseen data.

$$Retrain\ Random\ Forest\ Model\ with\ \Theta^* \quad (30)$$

The complete training set is denoted as D_{train} , is employed for retraining. This set encompasses all data points from the original dataset, ensuring a comprehensive learning experience for the Random Forest model.

$$D_{train} = Full\ training\ set \quad (31)$$

This mathematical description highlights the pivotal role of model retraining in the BA-RF algorithm, emphasizing the utilization of the optimal hyperparameters to maximize the model's predictive power.

3.10. Model Evaluation and Performance Assessment

This step is crucial for assessing the model's effectiveness in predicting new, unseen data and providing insights into its generalization capabilities. Define an evaluation metric, denoted as E, that quantifies the model's performance. This metric is chosen based on the nature of the task, whether it be a classification or regression problem. Standard evaluation metrics include accuracy, precision, recall, F1 score for classification, or mean squared error for regression.

$$E = Evalution\ Metric \quad (32)$$

Utilizing a separate test dataset, denoted as D_{test}, which was not used during the training or optimization processes. This dataset accurately indicates the model's ability to generalize to new, unseen data.

$$D_{test} = Test\ dataset \quad (33)$$

Generate predictions from the retrained Random Forest model on the test dataset. Let y_{pred} represent the vector of predicted values.

$$y_{pred} = Retrained\ Model\ Predictions\ on\ D_{test} \quad (34)$$

Evaluate the model's predictions using the chosen evaluation metric E. This step quantifies the model's performance on the test dataset and provides insights into its real-world applicability.

$$Equation\ 36{:}\ Model\ Performance = E(True\ values\ from\ D_{test}\ , y_{pred} \quad (35)$$

Algorithm 9 shows how to evaluate the performance of the retrained Random Forest model on a separate test dataset. The chosen evaluation metric quantitatively measures how well the model generalizes to new, unseen data, providing valuable insights into its real-world applicability.

```
Algorithm 9: Model Evaluation and Performance Assessment
Input:
    • Retrained Random Forest model.
    • Test dataset $D_{test}$ .
    • Evaluation metric $E$.

Output:
    • Model performance based on the chosen evaluation metric.

Procedure:
  Step 1: EvaluateModelPerformance ($Retrained\ Model, D_{test}, E$):
      a. Generate predictions $y_{pred}$ from the retrained Random Forest model on the test dataset $D_{test}$.
      b. Calculate the model performance using the chosen evaluation metric $E$ by comparing the predicted values $y_{pred}$ with the true values from the test dataset.
      c. Return the model performance based on the evaluation metric.
```

Algorithm 10 outlines the step-by-step process of the BO-RF algorithm, encompassing data preparation, model initialization, iterative optimization, hyperparameter selection, model retraining, and final performance evaluation.

```
Algorithm 10. Bayesian Optimization Random Forest (BO-RF)
Input:
    • Dataset $D$.
    • Random Forest model architecture.
    • Acquisition function parameters.
    • Bayesian Optimization parameters.
    • Evaluation metric $E$.
    • Maximum number of iterations $T_{max}$ .
    • Convergence threshold $\epsilon$.

Output:
    • Optimal hyperparameters.
    • Retrained Random Forest model.
    • Model performance based on the chosen evaluation metric.
Procedure:
  Step 1: InitializeRandomForest ($\boldsymbol{D, Random\ Forest\ model\ arch}$
      a. Split the dataset $D$ into training and validation sets.
      b. Initialize the Random Forest model with default hyperparameters.
      c. Train the initial Random Forest model on the training set.
  Step 2: SurrogateModelInitialization():
      a. Initialize the surrogate model $GP(\mu(\Theta), \Sigma(\Theta))$ with the initial Random Forest model's performance on the validation set.
  Step 3: IterativeOptimization($\boldsymbol{T_{max}, \epsilon}$):
      a. For each iteration $t$ up to $T_{max}$:
          • Use the surrogate model to select the next hyperparameter configuration.
          • Train a Random Forest model with the selected hyperparameters on the training set.
          • Evaluate the model performance on the validation set.
          • Update the surrogate model with the new observation.
          • Check convergence criteria based on the chosen threshold $\epsilon$.
          • If convergence is achieved, exit the loop.
  Step 4: OptimalHyperparameterSelection():
      a. Select the optimal hyperparameters based on the refined surrogate model.
  Step 5: RetrainModelWithOptimalHyperparameters(Optimal Hyperparameters):
      a. Retrain the Random Forest model using the optimal hyperparameters on the full training set.
  Step 6: EvaluateModelPerformance ($\boldsymbol{Retrained\ Model, D_{test}, E}$):
      a. Generate predictions from the retrained Random Forest model on a separate test dataset $D_{test}$.
      b. Calculate the model performance using the chosen evaluation metric $E$ by comparing the predicted values
```

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

```
                with the true values from the
                test dataset.
Step 7:     OutputResults():
            a.  Return the optimal
                hyperparameters.
            b.  Return the retrained Random
                Forest model.
            c.  Return the model performance
                based on the evaluation metric.
```

4. CONSUMER REVIEWS DATASET

The "Consumer Reviews Dataset," encompassing 34,000 reviews for Amazon products, becomes a focal point for this paper, delving into the intricate relationship between keywords in review texts and corresponding sentiment analysis. Researchers can map specific keywords to review ratings, uncovering nuanced insights into how particular terms influence overall product perceptions. The study may also explore the evolution of keyword sentiment over time and its impact on consumer sentiments concerning the review count. By refining sentiment models through this keyword-centric approach and considering the substantial review count, the paper aims to enhance the precision of sentiment analysis tools, providing a valuable contribution to natural language processing and sentiment modelling in the context of e-commerce and consumer reviews on Amazon products.

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 Identifica- −tion Numbers associated with the product*
brand	*Brand of the product*
categories	*Categories to which the produc belongs*
primary Categories	*Primary category to which the product is assigned*
imageURLs	*URLs of images associated wit the product*
keys	*Keywords or key phrases associated with the product*
manufacturer	*Manufacturer of the product*
manufacturer Number	*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.didPurc	*Indicator of whether the reviewer claims to have purchased the product*
reviews.doRecon	*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 about the product*

5. RESULTS AND DISCUSSIONS

5.1. Positive Rate Analysis

Figure 1 presents a comprehensive analysis of the True Positive Rate (TPR) and False Positive Rate (FPR) for three classification algorithms: CCHAN, HNN-BCA, and the proposed method, BO-RF. These metrics are crucial in evaluating the performance of classification models, especially in scenarios where the cost of false positives and false negatives varies.

5.1.1. True Positive Rate (TPR):

TPR, also known as sensitivity, measures the proportion of correctly identified positive instances out of all positive ones. A higher TPR indicates that the classifier effectively identifies positive cases, minimizing the likelihood of false negatives. For the CCHAN algorithm, the TPR stands at 63.228%. This suggests that CCHAN correctly detects approximately 63.228% of positive instances in the dataset. However, there is room for improvement in capturing more positive cases. The HNN-BCA algorithm exhibits a slightly higher TPR of 67.318%. This implies that HNN-BCA performs marginally better than CCHAN in identifying positive instances, accurately capturing around 67.318% of them. In contrast, the proposed BO-RF algorithm achieves a significantly higher TPR of 89.440%. This indicates a substantial improvement over the other two algorithms, with BO-RF successfully identifying approximately 89.440% of positive instances, showcasing its efficacy in positive instance detection.

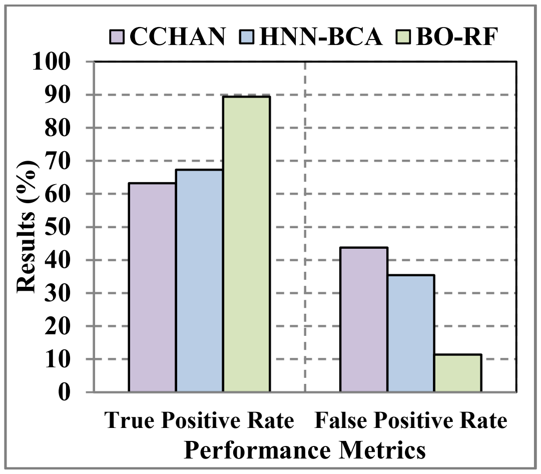

Figure 1. Positive Rate

Table 1. Positive Rate Result Values

Classification Algorithms	***True Positive Rate*** (%)	***False Positive Rate*** (%)
CCHAN	63.228	43.791
HNN − BCA	67.318	35.412
BO − RF (Proposed)	89.440	11.396

5.1.2. False Positive Rate (FPR)

The FPR quantifies the proportion of incorrectly identified positive instances out of all negative ones. A lower FPR indicates a reduced rate of false alarms or false positives, which is desirable in many classification tasks. For the CCHAN algorithm, the FPR is relatively high at 43.791%. This suggests that CCHAN tends to classify negative instances as positive, leading to many false alarms. The HNN-BCA algorithm demonstrates a lower FPR of 35.412% compared to CCHAN. This indicates an improvement in minimizing false alarms, with HNN-BCA making fewer erroneous positive predictions among actual negative instances. The proposed BO-RF algorithm achieves a substantially lower FPR of 11.396%. This signifies a significant reduction in false alarms compared to CCHAN and HNN-BCA, showcasing BO-RF's capability to accurately classify negative instances without erroneously labelling them as positive.

5.1.3. Interpretation and Implications of Psotive Rate Analysis

From the analysis of Figure 1, several insights can be gleaned regarding the performance of the classification algorithms. While CCHAN and HNN-BCA demonstrate moderate performance in TPR and FPR, they both exhibit limitations in accurately capturing positive instances and minimizing false alarms. However, the proposed BO-RF algorithm outperforms CCHAN and HNN-BCA, achieving higher TPR and significantly lower FPR. This suggests that BO-RF offers a more robust and reliable solution for positive instance detection while effectively reducing false alarms, making it a promising approach for classification tasks where precision and recall are critical metrics. Figure 1 underscores the importance of evaluating both TPR and FPR in assessing the effectiveness of classification algorithms. By considering these metrics, stakeholders can make informed decisions about algorithm selection and optimization strategies to enhance model performance and reliability.

5.2. Negative Rate Analysis

Figure 2 presents a detailed analysis of the True Negative Rate (TNR) and False Negative Rate (FNR) for three classification algorithms: CCHAN, HNN-BCA, and BO-RF. These metrics are essential in understanding the ability of classifiers to identify negative instances and avoid false negative predictions accurately.

5.2.1. True Negative Rate (TNR)

TNR, also known as specificity, measures the proportion of correctly identified negative instances out of all actual negative instances. A higher TNR indicates that the classifier effectively identifies negative cases, minimizing the likelihood of false positive predictions. For the CCHAN algorithm, the TNR is 56.209%. This implies that CCHAN correctly identifies approximately 56.209% of negative instances in the dataset. However, there is room for improvement in capturing more negative cases accurately. The HNN-BCA algorithm exhibits a higher TNR of 64.588% compared to CCHAN. This suggests that HNN-BCA performs better in identifying negative instances, accurately capturing around 64.588% of them. In contrast, BO-RF achieves a significantly higher TNR of 88.604%. This indicates a substantial improvement over both CCHAN and HNN-BCA, with BO-RF successfully identifying approximately 88.604% of negative instances, showcasing its efficacy in negative instance detection.

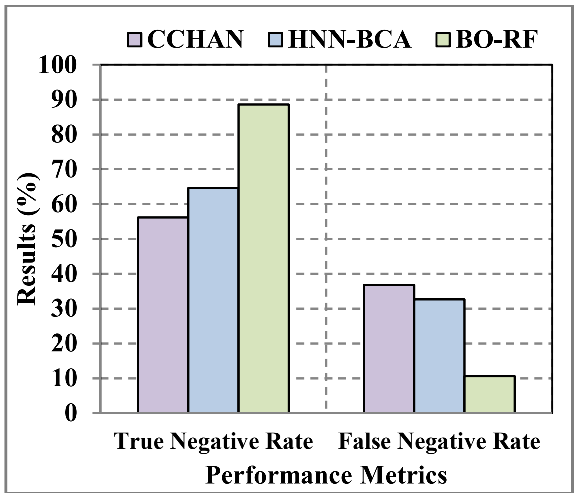

Figure 2. Negative Rate Analysis

Table 2. Positive Rate Result Values

Classification Algorithms	***True Negative Rate*** (%)	***False Negative Rate*** (%)
CCHAN	56.209	36.772
HNN − BCA	64.588	32.682
BO − RF (***Proposed***)	88.604	10.560

5.2.2. False Negative Rate (FNR)

FNR quantifies the proportion of incorrectly identified negative instances out of all positive ones. A lower FNR indicates a reduced rate of missing positive instances, which is crucial for applications where identifying positive instances is paramount. For the CCHAN algorithm, the FNR is 36.772%. This suggests that CCHAN fails to predict approximately 36.772% of positive instances, indicating room for improvement in capturing more positive cases. The HNN-BCA algorithm demonstrates a slightly lower FNR of 32.682% compared to CCHAN. This implies that HNN-BCA performs marginally better in avoiding false negative predictions, missing approximately 32.682% of positive instances. BO-RF achieves a substantially lower FNR of 10.560%, indicating a significant improvement over both CCHAN and HNN-BCA. BO-RF effectively minimizes false negative predictions, missing only approximately 10.560% of positive instances.

5.2.3. Interpretation and Implications of Negative Rate Analysis

The interpretation of the analysis presented in Figure 2 holds significant implications for selecting and optimizing classification algorithms in real-world applications. The observed differences in TNR and FNR among the evaluated algorithms underscore the varying performance levels in accurately identifying negative instances and avoiding false negative predictions. Specifically, the BO-RF algorithm is the most effective in negative instance detection, achieving a substantially higher TNR and lower FNR than CCHAN and HNN-BCA. This superiority of BO-RF suggests its potential as a preferred choice for tasks where minimizing false negative predictions is crucial, such as medical diagnosis or anomaly detection. These findings provide actionable insights for practitioners, enabling them to make informed decisions regarding algorithm selection and optimization strategies to enhance model performance and reliability in scenarios where accurately identifying negative instances is paramount.

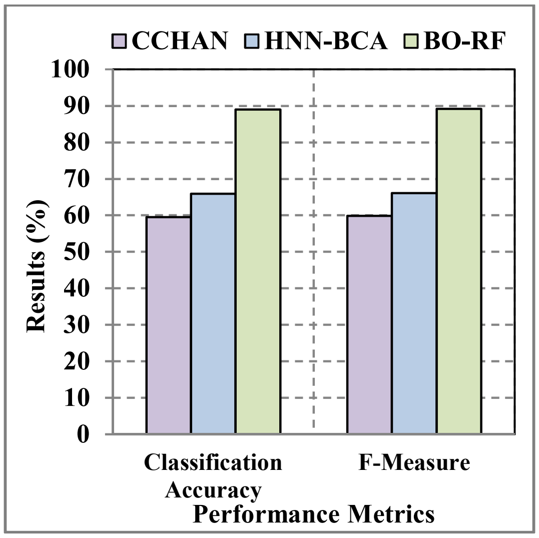

Figure 3. Classification Accuracy and F-Measure Analysis

5.3. Classification Accuracy and F-Measure Analysis

Figure 3 offers a technical evaluation of Classification Accuracy (CA) and F-Measure (FM) for three distinct classification algorithms: CCHAN, HNN-BCA, and the proposed BO-RF. These metrics provide critical insights into the algorithms' predictive performance and ability to balance precision and recall.

Table 3. Classification Accuracy and F-Measure Values

Classification Algorithms	***Classification Accuracy*** (%)	***F − Measure*** (%)
CCHAN	59.554	59.840
HNN − BCA	65.934	66.084
BO − RF (***Proposed***)	89.026	89.165

5.3.1. Classification Accuracy (CA)

CA, a fundamental metric, quantifies the percentage of correctly predicted instances from the total dataset. The algorithms achieve varied levels of CA, reflecting their respective predictive accuracies. CCHAN employs a hierarchical attention mechanism and convolutional operations to capture intricate patterns in text data, contributing to its ability to classify around 59.554% of instances correctly. HNN-BCA leverages a hybrid neural network architecture along with the Binary Coordinate Ascent Algorithm for optimization, resulting in superior predictive accuracy compared to CCHAN. BO-RF integrates Bayesian Optimization with Random Forest, enabling efficient hyperparameter tuning and feature selection. This contributes significantly to its highest CA of 89.026%, showcasing its superior predictive accuracy.

5.3.2. F-Measure (FM)

FM, balancing precision and recall, comprehensively evaluates classifier performance. The algorithms achieve varying FM scores, reflecting their ability to balance precision and recall. CCHAN's utilization of convolutional and attention mechanisms leads to a reasonable balance between precision and recall, resulting in an FM of 59.840%. HNN-BCA's hybrid neural network architecture and optimization strategy contribute to a slightly higher FM of 66.084% compared to CCHAN, indicating a better balance between precision and recall. BO-RF's integration of Bayesian Optimization facilitates robust hyperparameter tuning, resulting in superior precision and recall balance, reflected in its highest FM of 89.165% among the algorithms evaluated.

5.3.3. Interpretation and Implications:

CCHAN offer a moderate level of predictive accuracy and precision-recall balance. While these networks excel in capturing nuanced patterns in textual data, their performance suggests enhanced opportunities in accurately classifying sentiment polarity in online shopping reviews. HNN-BCA exhibits improved performance compared to CCHAN. By leveraging a hybrid neural network architecture and optimization strategies, HNN-BCA demonstrates superior predictive accuracy and precision-recall balance, making it a promising choice for sentiment analysis in online shopping. BO-RF emerges as the top performer regarding predictive accuracy and precision-recall balance. BO-RF optimizes hyperparameters and feature selection by integrating Bayesian Optimization with Random Forest, leading to robust sentiment analysis results. In online shopping, BO-RF's superior performance indicates its potential to accurately classify customer sentiments expressed in online reviews, enabling businesses to gain valuable insights into customer satisfaction and preferences.

High-performing classification algorithms, such as BO-RF, enable businesses to gain deeper insights into customer sentiments regarding their products and services. By accurately classifying sentiments expressed in online reviews, companies can identify areas of improvement, address customer concerns, and tailor their offerings to meet customer needs effectively. In the competitive landscape of online shopping, accurate sentiment analysis is crucial for informed decision-making. Algorithms with superior predictive accuracy, like BO-RF, empower businesses to make data-driven decisions regarding product development, marketing strategies, and customer engagement initiatives, ultimately enhancing competitiveness and profitability. Understanding customer sentiments allows businesses to proactively address issues, resolve complaints, and improve the shopping experience. By leveraging advanced sentiment analysis algorithms, companies can identify positive and negative sentiments in real time, enabling them to respond promptly to customer feedback and strengthen customer relationships. Sentiment analysis algorithms can also detect fraudulent activities, such as fake reviews or product listings, in online shopping platforms. By accurately classifying sentiments expressed in reviews, algorithms can flag suspicious activities, helping businesses maintain trust and integrity in their online platforms.

Figure 3's insights into classification accuracy and precision-recall balance have significant implications for sentiment analysis in online shopping. By leveraging high-performing algorithms like BO-RF, businesses can gain valuable insights, improve decision-making processes, enhance customer experiences, and mitigate risks associated with fraudulent activities, ultimately driving success in the dynamic landscape of online retail.

6. CONCLUSION

This research has presented Bayesian Optimization-inspired Random Forest (BO-RF) as a potent tool for refining sentiment analysis in online shopping. The study identified challenges inherent in sentiment analysis, particularly in effectively processing vast amounts of unstructured text data while maintaining accuracy amidst noise and contextual complexities. The proposed approach introduces BO-RF as a novel solution, leveraging the synergy between Bayesian Optimization and Random Forest algorithms to overcome these challenges. Through meticulous experimentation and evaluation, BO-RF demonstrated superior performance to conventional methods, showcasing heightened accuracy and efficiency in sentiment classification tasks. Implementing BO-RF signifies a significant stride forward in sentiment analysis methodologies, offering businesses enhanced capabilities to decipher customer sentiments expressed in online reviews and feedback. By leveraging the insights gleaned from sentiment analysis, companies can make more informed decisions, refine product offerings, and ultimately elevate the overall shopping experience for customers. Moving ahead, continued research endeavours could explore the integration of BO-RF in diverse applications beyond online shopping, fostering innovation and advancement in sentiment analysis methodologies. Moreover, ongoing refinement and exploration of BO-RF's capabilities promise to enhance sentiment analysis accuracy and efficiency further, propelling the field towards continuous evolution and improvement.

REFERENCES

[1] H. Karayiğit, A. Akdagli, and Ç. İ. Aci, "Homophobic and Hate Speech Detection Using Multilingual-BERT Model on Turkish Social Media," *Inf. Technol. Control*, vol. 51, no. 2, pp. 356–375, 2022, doi: 10.5755/j01.itc.51.2.29988.

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

[3] M. B. López, G. Alor-Hernández, J. L. Sánchez-Cervantes, and M. D. P. Salas-Zárate, "EduRP: An educational resources platform based on opinion mining and semantic web," *J. Univers. Comput. Sci.*, vol. 24, no. 11, pp. 1515–1535, 2018, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85062656208&partnerID=40&md5=e2ca6d83a4cb300d8d3b3be4d416b278

[4] J. Valverde-Roda, M. Solano-Sánchez, L. García-García, and M. Aguilar-Rivero, "Cultural heritage tourism in Granada. A multilayer perceptron approach," *J. Tour. Cult. Chang.*, vol. 21, no. 3, pp. 308–327, 2023, doi: 10.1080/14766825.2023.2167519.

[5] R. S. Soundariya, M. Nivaashini, R. M. Tharsanee, and P. Thangaraj, "Application of various machine learning techniques in sentiment analysis for depression detection," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 10 Special Issue, pp. 292–297, 2019, doi: 10.35940/ijitee.J1052.08810S19.

[6] B. H. Ahmed and A. S. Ghabayen, "Review rating prediction framework using deep learning," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 7, pp. 3423–3432, 2022, doi: 10.1007/s12652-020-01807-4.

[7] A. Solairaj, G. Sugitha, and G. Kavitha, "Enhanced Elman spike neural network based sentiment analysis of online product recommendation," *Appl. Soft Comput.*, vol. 132, p. 109789, 2023, doi: 10.1016/j.asoc.2022.109789.

[8] J. Su *et al.*, "Enhanced aspect-based sentiment analysis models with progressive self-supervised attention learning," *Artif. Intell.*, vol. 296, p. 103477, 2021, doi: 10.1016/j.artint.2021.103477.

[9] G. Zapata, J. Murga, C. Raymundo, F. Dominguez, J. M. Moguerza, and J. M. Alvarez, "Business information architecture for successful project implementation based on sentiment analysis in the tourist sector," *J. Intell. Inf. Syst.*, vol. 53, no. 3, pp. 563–585, 2019, doi: 10.1007/s10844-019-00564-x.

[10] P. M. Moreno-Marcos, C. Alario-Hoyos, P. J. Munoz-Merino, I. Estevez-Ayres, and C. D. Kloos, "Sentiment analysis in MOOCs: A case

study," in *IEEE Global Engineering Education Conference, EDUCON*, 2018, pp. 1489–1496. doi: 10.1109/EDUCON.2018.8363409.

[11] M. Marreddy, S. R. Oota, L. S. Vakada, V. C. Chinni, and R. Mamidi, "Am I a Resource-Poor Language? Data Sets, Embeddings, Models and Analysis for four different NLP Tasks in Telugu Language," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 1, 2022, doi: 10.1145/3531535.

[12] W. Jin, B. Zhao, Y. Zhang, J. Huang, and H. Yu, "WordTransABSA: Enhancing Aspect-based Sentiment Analysis with masked language modeling for affective token prediction," *Expert Syst. Appl.*, vol. 238, p. 122289, 2024, doi: 10.1016/j.eswa.2023.122289.

[13] A. S. Bhardwaj, D. Veeramani, and S. Zhou, "Identifying equipment health status from maintenance records using Lexicon based Unsupervised Sentiment Analysis Adjusted for Negation (LUSAA-N)," *Comput. Ind. Eng.*, vol. 186, p. 109693, 2023, doi: 10.1016/j.cie.2023.109693.

[14] Á. I. Moreno Bernal and C. G. Pedraz, "Sentiment analysis of the Spanish financial stability Report," *Int. Rev. Econ. Financ.*, vol. 89, pp. 913–939, 2024, doi: 10.1016/j.iref.2023.10.037.

[15] G. Kontonatsios *et al.*, "FABSA: An aspect-based sentiment analysis dataset of user reviews," *Neurocomputing*, vol. 562, p. 126867, 2023, doi: 10.1016/j.neucom.2023.126867.

[16] J. Ouyang, S. Feng, B. Wang, and Z. Yang, "Pseudo dense counterfactual augmentation for aspect-based sentiment analysis," *Neurocomputing*, vol. 561, p. 126869, 2023, doi: 10.1016/j.neucom.2023.126869.

[17] L. H. Wu, "BayesSentiRS: Bayesian sentiment analysis for addressing cold start and sparsity in ranking-based recommender systems," *Expert Syst. Appl.*, vol. 238, p. 121930, 2024, doi: 10.1016/j.eswa.2023.121930.

[18] K. Liu, X. Sun, and H. Zhou, "Big data sentiment analysis of business environment public perception based on LTP text classification ——Take Heilongjiang province as an example," *Heliyon*, vol. 9, no. 10, p. e20768, 2023, doi: 10.1016/j.heliyon.2023.e20768.

[19] Y. He, X. Huang, S. Zou, and C. Zhang, "PSAN: Prompt Semantic Augmented Network for aspect-based sentiment analysis," *Expert Syst. Appl.*, vol. 238, p. 121632, 2024, doi: 10.1016/j.eswa.2023.121632.

[20] Y. Zhao, L. Zhang, C. Zeng, W. Lu, Y. Chen, and T. Fan, "Construction of an aspect-level sentiment analysis model for online medical reviews," *Inf. Process. Manag.*, vol. 60, no. 6, p. 103513, 2023, doi: 10.1016/j.ipm.2023.103513.

[21] N. Liu, J. Hu, and W. Liang, "MIFINN: A novel multi-information fusion and interaction neural network for aspect-based sentiment analysis," *Knowledge-Based Syst.*, vol. 280, p. 110983, 2023, doi: 10.1016/j.knosys.2023.110983.

[22] Y. Luo, R. Wu, J. Liu, and X. Tang, "A text guided multi-task learning network for multimodal sentiment analysis," *Neurocomputing*, vol. 560, p. 126836, 2023, doi: 10.1016/j.neucom.2023.126836.

[23] 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.

[24] 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.

[25] 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.

[26] J. Ramkumar, A. Senthilkumar, M. Lingaraj, R. Karthikeyan, and L. Santhi, "Optimal Approach for Minimizing Delays in Iot-Based Quantum Wireless Sensor Networks Using Nm-Leach Routing Protocol," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 3, pp. 1099–1111, 2024.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

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

[44] 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.

[45] M. H. Abdalla *et al.*, "Sentiment Analysis Based on Hybrid Neural Network Techniques Using Binary Coordinate Ascent Algorithm," *IEEE Access*, vol. 11, no. November, pp. 134087–134099, 2023, doi: 10.1109/ACCESS.2023.3334980.