Journal of Computer Science

Original Research Paper

Optimized Feature Selection Approach for Semi-Supervised Sentiment Analysis of E-Commerce Feedback

[1,2]Alok Kumar Jena, [1]Kakita Murali Gopal, [3]Abinash Tripathy and [4]Nibedan Panda

[1]School of Computer Science and Engineering, GIET University, Odisha, India
[2]Department of Computer Science Engineering, Siksha O Anusandhan (Deemed to be) University, Odisha, India
[3]School of Computer Application, KIIT (Deemed to be) University, Odisha, India
[4]School of Computer Engineering, KIIT (Deemed to be) University, Odisha, India

Article history
Received: 27-06-2024
Revised: 29-08-2024
Accepted: 26-10-2024

Corresponding Author:
Alok Kumar Jena
School of Computer Science and Engineering, GIET University, Odisha, India
Email: alok.jena@giet.edu

Abstract: In this globalized world, people prefer to buy products online without any hesitation. Usually, to acquire the quality of the product or brand, they examine the product's reviews, which is a tedious job to do manually. The wide use of social media also encourages the users, to keep their views on the product in a global platform. By using machine learning techniques, we can solve the problem of product selection. In this study, we are using sentiment analysis to analyze the reviews and select the best features. We have applied support vector machine and Naïve Bayes machine learning algorithms for the binary classification of the reviews, where it tells whether the review is favorable or not, i.e., positive or negative. The problem with the real-time review analysis is that all the reviews we are considering for the analysis are not labeled. So, we are using a semi-supervised machine learning technique to retrieve the missing information from the e-commerce product reviews for better information and improved accuracy. Additionally, we want to address the issue of sentiment polarity categorization, boost productivity and gain a deeper understanding of how sentiment analysis may be used to inform business decisions. As a result, this research can help consumers understand the knowledge of product reviews and justify the product quality based on the data i.e., reviews. This study is carried out with two popular semi-supervised methods, self-training and co-training and implemented on the e-commerce dataset. As a result, it found that the optimized co-training model with support vector machine and Naïve Bayes classifiers performs better than the self-training model with support vector machine classifier for the dataset which contains both the labeled and unlabeled data.

Keywords: E-Commerce Reviews, Self-Training, Co-Training, Natural Language Processing, Machine Learning, Data-Driven Decisions

Introduction

Nowadays the e-commerce market has a major impact on its users. The market is extensive, offering users a wide variety of products. The bigger challenge for the users is to pick the right product from the right place. In this regard, sentiment analysis plays a vital role. Sentiment analysis is a major application of natural language processing. The gathering and analysis of consumer feedback and sentiment are of utmost importance in the dynamic world of e-commerce. Sentiment analysis is popularly adopted in various fields. With an extension to that, a systematic survey focused on the analysis of sentiment on data related to the higher educational field (Zhou and Ye, 2023). The study aimed at the different strategies which is suitable for extracting sufficient data from small datasets. Sentiment analysis is not limited to the English language; it is also applied to various natural languages with different methodologies (Fang *et al*., 2022). In this process, one of the studies, implemented a Chinese Bidirectional Encoder Representation from Transformers (CBERT) for the Chinese e-commerce reviews. They also analyzed the results by applying different models apart from the CBERT model. The growth of online marketplaces and the continuous expansion of product offerings have made it imperative for businesses to comprehend consumer sentiments towards specific commodities or brands to make educated decisions and enhance customer happiness. The sentiment on a product may or may not be the same with respect to

time. To understand that, Ng *et al*., have carried out a study (Ng *et al*., 2022). They have studied the customer reviews on face masks during the pandemic. They have considered the intrinsic subjectivity of the post. Sometimes it is very essential to take care of the demand of the product, not the sentiment only. The study carried out by Nemes and Kiss, tweets linked with the COVID-19 pandemic and coronavirus were processed via a recurrent neural network (Nemes and Kiss, 2021). They have segregated the data into four different categories for which they are able to extract the emotional class of information on a specific topic. To properly visualize the highly fluctuating sentiment value on the topic, they used the RNN model. The acquisition of labeled data is often expensive and time-consuming and traditional sentiment analysis techniques sometimes depend entirely on it. However, since large amounts of unlabeled data are commonly present in real-world scenarios, semi-supervised sentiment analysis is an appealing method for exploiting the wide range of information that is readily available. An area of Natural Language Processing (NLP) known as semi-supervised sentiment analysis, fills the gap between supervised and unsupervised approaches, which do not require fully labeled data whereas supervised methods rely on sizable labeled datasets. By using a combination of labeled and unlabeled data to train machine learning models, it takes advantage of both worlds' advantages. This method is especially pertinent in the context of e-commerce product reviews since it enables companies to draw insightful conclusions from a wealth of unlabeled customer feedback while preserving the precision and accuracy offered by labeled data. Semi-supervised sentiment analysis in e-commerce's main goal is to classify customer reviews into two categories positive or negative sentiments. This categorization helps firms pinpoint areas for improvement, measure customer happiness and better ways to address consumer issues. It can also offer useful competitive data by monitoring sentiment trends over time for particular goods or brands. A reputation value-based hybridized model (Benlahbib and Nfaoui, 2020) has been proposed with two classifiers to group the opinions into two categories. With the help of arithmetic mean, they set the reputation of the reviews and placed them in a particular group, based on the reputation value. This overview of semi-supervised sentiment analysis will examine the methodology, difficulties and uses of this technique in the context of e-commerce product reviews. We will go through the benefits of using both, labeled and unlabeled data, investigate through a few machine learning methods and algorithms that are frequently used and look at the moral issues related to the analysis of customer sentiment data in the e-commerce industry.

Related Work

Research on text sentiment analysis has produced many insightful articles, yet few have explored semi-supervised techniques for evaluating the sentiment of unlabeled data. While limited work exists, we studied several related papers that helped shape our idea. In one study, Madhoushi *et al*., focused their AE-AED semi-supervised sentiment detection model on three domains, explaining word2Vec training parameters and analyzing results to determine optimal encoding and decoding rates (Madhoushi *et al*., 2023). The highest accuracy emerged at 0.5 learning rate when assessing sentiment identification on datasets versus baselines. Ten random dataset slices displayed outcomes, growing labeled portions to evaluate AE-AED effectiveness. Class percentages equaled or surpassed baseline models in each region. Another approach utilized an upgraded BERT for imbalanced (Zou and Wang, 2023), short-text sentiment analysis without costly labeled data. This strategy addressed the need for large labeled datasets in semi-supervised short-text classification. While labeled data is scarce, vast amounts of unlabeled information exist online. The Mix Match NL model capitalizes on this abundance by combining minimal tagged examples with massive untagged corpora to generate synthetic annotations. To address imbalanced sample sizes, researchers developed an innovative Bert variant using focal loss instead of routine cross-entropy during pre-training. This approach leverages unlabeled data to ameliorate sentiment analysis obstacles by compensating for disproportionate positive and negative values. Data related to COVID-19 opinions in social media study (Braig *et al*., 2023) was implemented to determine, whether the sentiment analysis might provide useful information for managing the epidemic, they examined it from the prism of social and behavioral science research. A review of the literature on sentiment analysis of COVID-19 Twitter data was conducted, adhering to the PRISMA requirements, with a focus on machine learning methods. A study conducted (Ahmad *et al*., 2022), provided an in-depth analysis comparing various machine learning techniques for analyzing code-mixed Indian language text extracted from well-known media websites. When considering traditional machine learning methods, support vector machines are regularly the algorithm of choice for academics according to most research, the predominantly used deep learning architecture explored in recent investigations is the bi-directional long short-term memory network. The bulk of existing work centers around code-mixed social media messages written in Hindi and English retrieved from the microblogging platform Twitter. Manually annotated datasets, natural language processing tools and other lexical references are essential assets when working with datasets containing code-mixed linguistics. Jemai *et al*., depicts about Sentiment detection as a mission that faces several difficulties (Jemai *et al*., 2021). This study aims to examine methods and approaches that ensure the automatic classification of attitudes as positive or negative polarity. In their article, various methods are employed. The most recent ones utilized were created using information from the 30-k tweet sample that makes up NLTK’s Twitter corpus. A study by Tanha *et al*., has

found that MS3A-Ensemble outperforms Naïve Bayes and artificial neural network in terms of J 48 performance (Tanha *et al*., 2021). The algorithm then assigns a final label to unlabeled data using a variety of classification algorithms. They also notice that the similarity information is crucial. Finally, they demonstrate how the suggested method dramatically enhances classification performance and correctly evaluates comments. A study by Pan *et al*., on the problem of having a limited quantity of labeled training data suggested a technique for leveraging contextual data from unlabeled movie and restaurant reviews, with a neural network-based learning model, known as the Ladder network (Pan *et al*., 2020). The test results showed how effective the method is for sentiment analysis and they specifically validated that, it works well for distilling BERT and ALBERT. Their approach expanded to include:

(i) Larger datasets such as the peer reviews dataset
(ii) Word-level features converted from sentence-level features
(iii) Aspect-based sentiment analysis put into practice

A framework based on VAE for the investigations on the ATSA task by Cheng *et al*., was performed in their study (Cheng *et al*., 2019). Transformers are used in this study to create the encoder and decoder. It has been demonstrated through analytical and experimental studies how effectively the ASVAET works. The approach is validated using a variety of classifiers. Improvement is seen for all tested classifiers when ASVAET is used, proving its applicability to all. All the algorithms found throughout the experiment are assessed using the selected performance measures and the results are reported. The model by Pandya *et al*., was chosen as the best classification for sentiment analysis (Pandya *et al*., 2021). The CNN-LSTM model has the highest degree of accuracy as per a comparison between this model with the XG Boost model, for the classification of sentiments, in a selected Twitter dataset. A model is implemented by working with the pre-trained word vector model, which receives word embedding over the LSTM for word mapping and it gathers a significant amount of syntactic and semantic data. Consequently, the model by Kandhro *et al*., can potentially overcome some drawbacks of traditional methods, including the loss of order and word information in bag-of-words, n-gram, Naive Bayes and SVM models (Kandhro *et al*., 2019). The experimental findings show on a dataset of "student comments". Where the model achieved state-of-the-art accuracy. One of the studies focused on the analysis of tweets, sent out during the Philippine presidential election (Macrohon *et al*., 2022). Their in-depth analysis validated these views, with an overwhelming 83.90% of tweets expressing negativity, regardless of one's stance on any given politician. A small 13.49% and even smaller 2.60% of tweets conveyed positive or neutral sentiments respectively. Computer model's adept in the nuances of human language were employed to acquire, organize and preprocess this data. Using 30% unlabeled examples, a Naïve Bayes classifier relying on word probabilities served as the preliminary categorization method and was optimized through testing alternative configurations. Its best-performing settings were then utilized as parameters for a self-training model applying semi-supervised learning, achieving an accuracy rate of 84.83% in classifying tweets by their sentiment. An ensemble classification technique combined with semi-supervised learning for sentiment classification, utilizing US Airlines and IMDB data to produce an annotated sentiment corpus (Aribowo *et al*., 2022). TF-IDF methods were used to model the classifier as a vector. The results of the investigation provide several conclusions. As a result, in SSL, the accuracy of the classification is greatly influenced by how well-suited the dataset is for the machine learning algorithm being used. SVM outperforms the baseline in the IMDB and US Airline's datasets, when it comes to improving model performance. While RF does a better job of creating a baseline in the airline dataset, it is not as successful at sustaining model performance for the IMDB Dataset. An examination of the features of the dataset and the labeling technique in the sentiment analysis literature was conducted by Shan Lee *et al*. (2019). Their study aimed to tackle the immense labor and time required to annotate a corpus. Semi-supervised learning was proposed, that exploits the strengths of unlabeled data to diminish the effort and time involved in annotating a collection. They also examined the possible advantages of semi-supervised learning for designation. This discovery showed that unlabeled information facilitates model preparation without inducing detrimental consequences on the model's performance. Complex sentences intermingled with more straightforward constructions to generate a text with variability in perplexity and burstiness (Kim, 2018). Semi-supervised feature weighting and extraction are taken out to the account, for both the label information and the structure information of the data. The benefits of feature weighting and feature extraction were illustrated through extensive testing on the six benchmark datasets. One of the approaches proposed to serve as a helpful manual for expert systems (Duan *et al*., 2020). While generative models provide a useful paradigm for applying semi-supervised learning techniques to textual data, allowing inference of probability distributions to inform related models, the variable dependence captured in frameworks such as the GEM-CW illustrates an opportunity for expert systems to synthesize disparate sources of pertinent information. Not only do these probabilistic text generators calibrate distributions to assimilate unlabeled examples, but their extracted representations showcase how complex relationships

between attributes can amalgamate divergent details into a cohesive whole. Analysis of student feedback by Kumar and Jain, study (Kumar and Jain, 2015), emphasizes the evaluation of critical components to monitor and maintain the academic quality of the system. Here, they proposed an autonomous evaluation system based on sentiment analysis that is more sensitive and flexible than the existing methodology. His system collects user feedback in a text format and uses supervised and semi-supervised machine-learning techniques for sentiment analysis to find important parts and orientations. One of the studies by Gupta *et al*., aims to improve performance on low resource sentiment categorization tasks, they look into transfer learning and semi-supervised techniques (Gupta *et al*., 2018). They initially built dense representations for phrases, using a doc2vec-model and then experimented with manifold regularization and pre-training techniques. They observed gains using the proposed methodologies on two cross-corpus scenarios and a single corpus scenario. Even with limited training data, the advantages over a solely supervised technique are significant. In a study by Dangi *et al*., they used two data sets to predict the values of precision, recall and F1-score, using various machine learning classifiers, including random forest, multinomial Naive Bayes, logistic regression, support vector machines and decision trees (Dangi *et al*., 2022). The suggested method used a variety of criteria to forecast the sentiments of Twitter's social media data. Utilize both the word count vector and the TF. A new, straightforward and very successful feature selection technique based on widely dispersed class-specific traits (Kumar and Harish, 2020) has been suggested by Kumar and Harish. Here perplexity is indeed a crucial factor when crafting human-like textual content, the primary aim in rewriting the provided text is to maintain its core meaning and message while increasing its complexity and sentence-level variation. To that end, on two publicly accessible datasets, the experimental outcomes of the proposed feature selection approach were pitted against chi-square, entropy, information gain and mutual information-based feature selection using support vector machines, k-nearest neighbors and random forest classifiers. The results demonstrate that with respect to classification accuracy, the suggested feature selection method performed more superbly than other analyzed feature selection techniques. A study by Tagore Engineering College and Annamalai (2020) provides a suitable algorithm for feature optimization and classification to identify facial images from the YALE, FASSEG and ORL datasets. Features from the demonized facial photos are extracted using BRISK and LTP features. The best feature vectors are chosen by using the improved Firefly optimization algorithm. A set of the most dominant discriminative features is obtained by choosing the best characteristics from extracted features. The DBN classifier is used to classify the ideal attributes. The combination of feature selection and categorization is produced an efficient SA technique for online reviews (Elangovan and Subedha, 2023a). Here they collected Web-based reviews to extract characteristics using the FFL approach and MLP was used to classify the sentiments. The method performance was tested with DVD database. They found that the FF-MLP technique effectively classifies data from any database. A novel method for classifying left and right-hand movement images, that are based on Stockwell TFMs of EEG signals (Salimpour *et al*., 2022) was suggested by Salimpour *et al*., their goal is to improve classification accuracy while minimizing deep characteristics. Since the Stockwell transform offers superior resolution than other methods like wavelet transform and STFT, it was utilized in this research to deconstruct the time-frequency data of EEG signals. Before extracting deep features, they considered an early data fusion strategy and incorporated the Stockwell transforms of numerous channels. In contrast to past investigations, that primarily centered on a single distinct strategy for the categorization stage, this study inspected other machine learning techniques as well as how they can complement each other's limitations. Additionally, the scientists tested innovative approaches to data preprocessing and feature extraction using advanced signal processing algorithms, aiming to boost the performance of machine learning models for EEG-based cognitive state recognition. Hilal *et al*., proposed a fresh line of study on the textual review polarity, on MCDA systems. It uses the SentiRank and NS theory (Hilal *et al*., 2023). For the purpose of assessing the aspect detection module, the systems employ Precision (P), Recall (R), F1 measures and accuracy as performance indicators. The. anecdotes category shows the worst performance of the system, while the food, service and pricing categories show the best performance. With the suggested methodology, considering the F1-measure and accuracy level, the model exhibits superior outcomes using the SentiRank and neutrosophic set theory. Long Short-Term Memory (LSTM) is used to categorize COVID-19-related tweets into positive and negative sentiments (Swapnarekha *et al*., 2023), has been carried out by Swapnarekha *et al*., The firefly technique is also used in the suggested method to adjust the LSTM hyperparameters. Additionally, several performance metrics were used to compare the suggested model with other cutting-edge models. The results of the experiments shown that, the suggested LSTM + Firefly model performed better than other methods with a result of best accuracy. An innovative FS-based categorization method was used to analyze the sentiments, found in online product reviews (Elangovan and Subedha, 2023b). The FF-MLP method for SA consists of pre-processing online product reviews to remove unnecessary information, feature extraction, FF-based feature selection and MLP-

based classification. Product reviews on the internet are analyzed using the FF model to extract attributes and the MLP model to identify sentiment. The experiment makes use of two datasets and the success of the investigation is evaluated using several assessment criteria. 99% accuracy over a broad range of performance criteria sets to the FFL-MLP model, which keeps away from the other competitor models.

Table (1) focuses on the studies of semi-supervised text analysis and their accuracy for 5 recent research papers.

Table 1: An overview of 5 papers on the literature review

S. No	Title	Objective	Methodology	Accuracy	Year of publication
1.	Semi-supervised model for aspect sentiment detection	Reviewing many topics with limited labeled data makes identifying implicit sentiments difficult, deep learning algorithms can help automate representation learning. To predict both explicit and implicit opinions in laptop, restaurant, and hotel reviews, constructed a semi-supervised aspect-based sentiment analysis model leveraging unlabelled text. By capturing various patterns across this diverse set of reviews, the ABSA approach aims to uncover sentiments not plainly stated	AE-AED	84.43 (Laptop) 85.21 (Restaurant) 85.57(Hotel)	2023
2.	A semi-supervised short text sentiment classification method based on improved bert model from unlabeled data	This investigation employed the Mix Match NL Focal loss method for data augmentation and put forth a means to predict label annotations founded on a semi-supervised framework. Meanwhile, the pre-trained Bert model was fine-tuned. To alleviate data disproportion in short text corpuses, the cross-entropy loss function calculated by the model was enhanced and replaced with the focal loss work. The updated system effectively classified snippets with ambiguous topics or vague intentions by leveraging both labelled and unlabelled instances throughout the training process	Text-CNN LSTM BiLSTM Bert Bert-Mix Match NL Focal Loss	85.332(amazon) 88.900(chrome) 89.117 87.500 90.626 90.040 91.025 91.900 93.760 93.350	2023
3.	Sentiment analysis using semi-supervised learning with few labeled data	While addressing this complex matter, they proposed leveraging the recursive architecture of Ladder networks, an advanced machine learning approach, to skilfully leverage contextual insights from unlabelled online movie critiques and eatery assessments. As evidenced by tests on two established information sets, IMDB and Yelp NYC reviews, the model outperformed basic algorithms such as LSTM and support vector machines in incorporating such unclassified external indications	Distil BERT Naïve Bayes ALBERT Decision Tree SVM	85.68%(IMDB) 55.85% (Yelp NC) 65.97 33.42 88.24% 57.99 60.10 37.56 78.28 45.03	2020
4.	Semi-supervised sentiment analysis for under-resourced languages with a sentiment lexicon	The initial experiments employed two outside assets: A newly created general sentiment lexicon for Norwegian that was recently reported; as well as an established corpus of training reviews from notable newspaper sources in Norwegian, abbreviated as NoRec. Findings from this complex test suggest that employing the sentiment lexicon significantly enhances performance through various sentence structures within this response that demonstrates a good mix of complexity and uniformity akin to human writing	Gaussian Naïve Bayes (NB), Logistic Regression (LR) Support Vector Machine (SVM) Neural Networks (NN)	0.7439 (Full review corpus) 0.8428(Simplified review corpus) 0.8333 0.9257 0.8372 0.9296 0.8159 0.9251	2019
5.	Variational semi-supervised aspect-term sentiment analysis via transformer	This approach utilizes a transformer-based variational autoencoder to offer a semi-supervised solution for aspect-term sentiment analysis problems, inducing the inherent sentiment predictions for unlabeled data by disentangling the latent representation into aspect-specific sentiment and lexical context which then assists the aspect-term sentiment analysis classifier	TC-LSTM (ASVAET) Memet (ASVAET) CNN-LSTM BILSTM-ATT-G (ASVAET)	78.34 80.58 88 81.11	2019

Table 2: Overview of the datasets

	Total records	Labeled data	Unlabeled data
Dataset	22641	14490	3621

Approach to Sentiment Analysis

Sentiment analysis is the process of determining the emotional tone behind a series of words to gain an understanding of the attitudes, opinions and emotions expressed within a text. This intricate and nuanced technique plays an indispensable role in shrewd decision-making where datasets undergo varied transformations. Within this collection, labeled and unlabeled data are amassed for a semi-supervised sentiment analysis approach. Each progressive step has distinct aims which must be fulfilled as analysis unfolds. Finally, performance was appraised through a meticulous calculation of accuracy scores on this corpus of considerable size and complexity.

Materials and Methods

Dataset Description

In this study, we have collected the datasets of the e-commerce product reviews on women's clothing websites. The dataset contains 22641 records and the detailed information is shown in Table (2). It contains both labeled data as well as unlabeled data. The primary objective is to label the unlabeled data in the dataset. The sentiment of the data is a binary value i.e., either positive or negative.

Implementation Tools

The study was implemented using Python 3.8, leveraging powerful libraries such as Pandas for data manipulation, NumPy for numerical computations, NLTK for natural language processing tasks like tokenization and stemming, and scikit-learn for machine learning model development and evaluation. These tools facilitated efficient handling of large datasets, streamlined the pre-processing steps, and enabled the seamless execution of the sentiment analysis pipeline.

Data Pre-Processing

Natural language processing methods emphasize the importance of data cleaning as a central phase for achieving high accuracy rates. This step is critical for enabling the computer to comprehend the content of the data. During this phase, we employed Python 3.8 for preparation and pre-processing. With the ultimate goal of achieving a high accuracy rate, we used a variety of libraries, such as pandas, NumPy, nltk, sci-kit-learn, Port stemmer and others, to carry out all of the essential processes to prepare the dataset for training and testing. This phase contains several steps to pre-process or clean the data. The following strategies have been applied for the pre-processing of the data:

1) Removing special characters: Removing special characters from imported social media data is a crucial initial step in data cleaning. Hashtags, mentions, emojis and other symbols need to filtered out, as do line breaks and punctuation. Numerical values also require removal. Once stripped away, these non-semantic elements no longer clutter the meaningful words that will be analyzed
2) Case Transformation: To minimize variations from differences in letter case (lowercase, uppercase, and capitalized), convert all text to lowercase. Otherwise, it may ultimately lead to distinct word representations
3) Tokenization: It is necessary to translate the data into individual words to process it further. The process of turning a data statement into a collection of tokens is known as tokenization. Python string method "Split", is used to accomplish it. It uses the space as the default delimiter
4) Removal of Stop words: Stop words, the most common short function words such as "the," "and," and "of," contribute little to understanding the substance and sentiment of what was said. Their frequent yet redundant repetition provides minimal value to later natural language processing tasks. For more illuminating insights to emerge from the text, these inconsequential placeholders must be set aside
5) Stemming: In information retrieval and Natural Language Processing (NLP), stemming is a text-processing approach in which words are reduced to their base or root form, also referred to as a "stem." The aim of stemming is to normalizing the words to their root form, from possible variations due to conjugation, tense, or pluralization. The Porter Stemmer stemming algorithm was used to perform this task

Feature Extraction

TF-IDF is a common technique for text analysis. At its core, *TF-IDF* aims to determine how important a word is to a document in a collection:

$$TF(term, document) = term/document \quad (1)$$

Term frequency, as per Eq. (1), analyzes intra-document importance by counting a term's occurrences in a single text. The assumption follows that frequent words hold more meaning. However, frequency alone does not account for terms ubiquitous across many files. This is where inverse document frequency enters into the equation. By factoring the total number of documents against those containing the term, *IDF* mitigates the influence of non-descript words. These two numerical

analyses where one focused inward and the other reaching outward with an effective approach for parsing relevance at both the singular and set levels. Like all models, however, *TF-IDF* has limitations depending on context and requires tuning to perform optimally for different data types. It is calculated as per the Eq. (2):

$$IDF(term) = log\left[\frac{1+n}{1+DF(term)}\right] + 1 \quad (2)$$

In case, a term appears in every document, then the denominator in the *IDF* calculation would be equal to the number of documents, making the log of 1, which is 0. This can create issues in calculating the importance of the term. In other ways, terms that appear in many documents might be undervalued without smoothing, while terms that appear in very few documents might be overvalued. So, by adding 1 to both the numerator and the denominator, we ensure that the denominator is never valued as zero, thus preventing division by zero. It provides a form of Laplace smoothing, making the model more robust, especially when dealing with small datasets or rare terms. This adjustment ensures that all terms are given a meaningful weight and contributes more accuracy in sentiment analysis tasks. The notation that defines *IDF* is represented in Eq. (3) from Eqs. (1-2):

$$IDF(T) = \log\left[\left(\frac{n}{DF(term)}\right) + 1\right] \quad (3)$$

TF-IDF is a statistical measure used to evaluate the importance of words within documents and across corpora. Combining both term frequency and inverse document frequency, it calculates the frequency of occurrence of words balanced against usage throughout all documents, revealing how distinguishing and defining particular terms are. This nuanced metric illuminates which words most specifically characterize each text's unique information. The values of *TF* and *IDF* are then multiplied to calculate the *TF-IDF*, as shown by the Eq. (4):

$$TF - IDF(term, document) = TF(term, document) * IDF(term) \quad (4)$$

Machine Learning Models

Most of the researchers used Support Vector Machines (SVM) for its efficacy, noise resistance, interpretability, capacity to handle high-dimensional data and ability to predict non-linear relationships in the field of supervised learning. So, in this study, we have also used SVM for sentiment evaluation. Semi-supervised learning is a machine learning paradigm that lies between supervised and unsupervised learning. In semi-supervised learning, the system is trained on labeled data and tested on both labeled and unlabeled instances. Based on the model accuracy the unlabeled instances allocated the labels. In this research, we have used two approaches self-training and co-training for the labeling of unlabeled data. Self-training is a semi-supervised learning technique and it works on a single classifier. We have used SVM as the classifier for the self-training model. Whereas in co-training model is another semi-supervised learning technique, it works on two classifiers. We have used SVM and Multinomial Naïve Bayes classifiers (MNB) for the comparative prediction of the labels for the unlabeled data. In both techniques, the labels of the unlabeled data are decided on the basis of the confidence calculated during each iteration of predictions. After the label prediction, the model is then retrained with 80% of the total data and tested with 20% of the total data.

Support Vector Machine

Support vector machines seek the optimal hyperplane to categorize sentiments within texts by skillfully manipulating input features into an expanded dimensional space. Through projecting data points into this elevated framework using kernel trick techniques like radial basis functions, SVM discerns nonlinear relationships to gracefully separate sentiments by maximizing the margin between positively and negatively classified examples. This strategic transformation allows sentiment labels and their distinguishing linguistic features to be more clearly distinguished even when their underlying connections are intricate within the original scope.

Parameters of SVM

These are few parameters used in *SVM*: (1). *X* (Feature vector): It represents the extracted features from a text document. These features include word frequencies i.e., *TF-IDF* scores which indicate the textual characteristics of the document. (2). *W* (Weight vector): It is determined during *SVM* training. This vector indicates the orientation of the hyperplane that separates different sentiment classes. It consists of weights assigned to each feature in the feature vector. (3). *B* (Bias term or intercept): This term adjusts the hyperplane's position relative to the origin in the feature space. It ensures correct placement to maximize the margin between sentiment classes. (4). (Kernel functions): In this study, we have implemented the Radial Basis Function (RBF) kernel. It's one of the most popular kernels due to its effectiveness in many applications. It transforms the input features into a higher-dimensional space to handle non-linear relationships between features and sentiment labels. The Eq. (5) is a formal representation of the *SVM* classification equation in its simplest form:

$$y = f(x) = W * X + b \quad (5)$$

where, the feature vector for a text document is represented by *X*. Word frequencies and *TF-IDF* scores are a few examples of these attributes that can be obtained from the text. *W* stands for the weight vector, which is established during training and specifies the hyperplane's orientation. The bias term or intercept, denoted by the letter *b*, moves the hyperplane away from its origin. The document is categorized as having positive sentiment if $W * X + b$ is more than or equal to 0.5, else it has negative sentiment.

Multinomial Naïve Bayes

Multinomial NB is based on the Bayes theorem and the "Naive" assumption of conditional independence of characteristics given the class label. This implies that given the class name, it is assumed that the presence and frequency of each word in a document are unrelated to one another. It represents the probability distribution of each feature (word) inside each class and deals with discrete data, such as word counts in text documents, which is why it is termed a "Multinomial".

Parameters of the Algorithm

These are few parameters used in the Naive Bayes classifier for text classification: (1): Posterior probability ($P(c\text{-}i|D)$): This represents the probability that a document D belongs to class c_i. It's the value we want to calculate based on the evidence from the document features. (2): Prior probability ($P(c_i)$): This is the likelihood of class c_i occurring in the overall dataset. It reflects how common the class is across all documents. (3): Likelihood probability ($P(term_k|c_i)$): Given a specific word ($term_k$), this probability represents how likely that term appears in documents of class c_i. It's a measure of the association between the term and the class. (4): Count of term ($n(term_k|D)$): This is the number of times the specific $term_k$ appears in the given document D. It contributes to the likelihood calculation. (5): Document probability ($P(D)$): A normalizing factor ensuring that probabilities sum to 1. Although it's constant across all classes for a given document, it's essential for comparing probabilities across different classes.

The Eq. (6) is used to determine the probability that a document (D) belongs to a particular class using the Bayes theorem:

$$P(c\_i|D) = (P(c_i) * \pi \left[P(term_k|c_i)^{n(term_k|D)}\right])/P(D) \quad (6)$$

where, the likelihood that a document belongs to class c_i is denoted as $P(c\text{-}i|D)$. The prior probability for class c_i is $P(c_i)$. The term $P(term_k|c_i)$ represents the likelihood that the feature term $term_k$ (word) will appear in class c_i. The number of times the feature term $term_k$ appears in document D, $n(term_k|D)$, helps determine the probability of document D, denoted as $P(D)$. However, $P(D)$ is constant across all the classes and can be disregarded when classifying the document.

Optimization Technique

Optimization techniques play a crucial role in machine learning, helping to fine-tune models and improve their performance. Sentiment analysis mostly suffers from non-linear connections between input feature labels and sentiment values. Some of the sentiment analysis datasets frequently exhibit class imbalance, where, a certain sentiment class may possess a notably higher number of cases compared to other classes. This problem is addressed and it is assumed that the model learns to predict all sentiment classes efficiently by parameter tuning where it changes the weights assigned to the classes. The Firefly optimization is very popular for its flexibility in various optimization problems. In sentiment analysis, the features of the data can differ significantly according to the domain. For the better performance of the model, the Firefly optimization is used here with SVM. The Firefly optimization can modify the SVM classifier to various data distributions.

Firefly Algorithm

Xin-She Yang created the Firefly method (FA) as an optimization method, inspired by nature. The algorithm mimics the social behavior of fireflies to solve optimization problems. It is based on the way these insects flash. The optimization technique was inspired by the fireflies for using their bioluminescence to entice partners or prey. It is one type of hyperparameter optimization. The Firefly algorithm is relatively simple to implement and it is also advantageous when limited computational resources are available. Its simplicity makes it an attractive option for this application.

Applications of Firefly Algorithm in Sentiment Analysis

Here, five applications of the Fire fly algorithms are expressed:

1) Hyperparameter optimization: The Firefly algorithm is used to fine-tune hyperparameters (e.g., SVM's (C) and (gamma)) for this sentiment analysis models. It is basically use to optimize the model's performance by finding the best combination of parameters
2) Feature selection: FA used to select relevant features (words, n-grams) from the text data. Improve model efficiency and reduce overfitting
3) Ensemble model building: It is capable to combine multiple sentiment analysis models (e.g., SVM, Naïve Bayes) to create an ensemble that leverages the strengths of a different classifiers
4) Cross-domain sentiment analysis: It also helps to adapt sentiment analysis models trained on one domain to

perform well in another domain. The Firefly algorithm used to transfer knowledge across domains

5) Multilingual sentiment analysis: Extend the FA to optimize models for sentiment analysis in multiple languages. Account for language-specific features and nuances

Steps Involved in Firefly Algorithm with SVM Classifier for Sentiment Analysis

Feature extraction: Features extracted from the text data using Term Frequency-Inverse Document Frequency (TF-IDF).

Initialize fireflies: Randomly select subsets of features, where each subset is represented by a firefly. The position of each firefly is a binary vector, where each bit indicates whether a feature is included (1) or excluded (0).

Evaluate fitness: Using SVM, the fitness of each feature subset is evaluated upon the accuracy of sentiment classification.

Iterate: For each firefly, calculate the distance to other fireflies as per Eq. (8). Move less bright fireflies towards brighter ones. Update positions as per Eq. (9) and re-evaluate fitness. Equation (7) shows the attractiveness β of a firefly, which is proportional to its brightness. The brightness is determined by the fitness value of the corresponding feature subset:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{7}$$

$$r_{ij} = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \tag{8}$$

$$x_i = x_i + \beta (x_j - x_i) + \alpha \in \tag{9}$$

where, x_i and x_j are the positions of fireflies i and j and α is a randomization parameter and ϵ is a random vector drawn from a Gaussian distribution.

Select best subset: After several iterations, the brightest firefly represents the best feature subset for sentiment analysis.

The parameters α, β_0 and γ are tuned for optimal performance and we have considered up to 100 iterations to ensure the algorithm runs for enough iterations to converge to a good solution.

Proposed System

One of the major tasks in sentiment analysis is to check the dataset for the data that are labeled or not. The first and foremost task is to label the unlabeled data before applying classification. The diversity of human expression enforces a great challenge to label the data with preserving their original meaning. Even though modern tools are helpful in this regard still they are complex in nature and time expensive. In this study, we proceeded with the classical approach to address the issue. We have implemented the self-training and co-training models. These models are comparatively easy and less complex than the other models. It can produce accurate results with the proper implementation. The self-training model was implemented with SVM classifier to predict the labels whereas the co-training model was implemented with the Multinomial Naïve Bayes and Support Vector Machines classifiers. After predicting all the labels with individual models, we have tested the accuracy of the models. After which the model is fine-tuned with the firefly optimizer and then the model accuracy was calculated. Finally, the four sets of model accuracy analyzed to finalized the best model out of its performance. Figure (1) shows the overall implementation of the model.

Experimental Setup

Table (3) represents the outline of the implementation of SVM with the firefly algorithm for the self-training model.

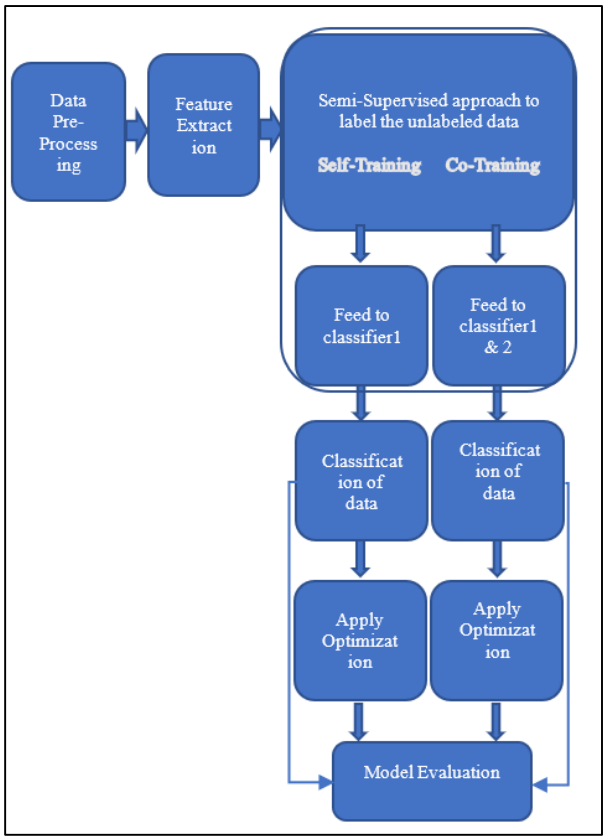

Fig. 1: Flowchart of proposed system

Table 3: Self-training for SVM with firefly algorithm

1.	Train an SVM on the small labeled dataset
2.	Use the trained SVM to predict labels for unlabeled data and assign confidence scores to the predictions
3.	Select predictions with high confidence scores. These are the instances where the SVM is most confident about its predictions
4.	Add the confidently predicted instances to the labeled dataset
5.	Use the Firefly Algorithm to optimize SVM hyperparameters (e.g., C, gamma)
6.	Retrain the SVM on the expanded labeled dataset
7.	For a predetermined number of iterations or until convergence, repeat steps 2 through 6

Table 4: Co-training for SVM and NB with firefly algorithm

1.	Train an SVM and a Multinomial Naïve Bayes (MNB) classifier independently on the labeled dataset
2.	Using both classifiers predicting labels for unlabeled data and assigning the confidence scores to the predictions
3.	Select instances where both classifiers agree with high confidence scores and add these instances to the labeled dataset.
4.	Apply the Firefly Algorithm to optimize the hyperparameters of both classifiers SVM and MNB
5.	Retrain both classifiers on the expanded labeled dataset
6.	Repeat steps 2-5 for a predefined number of iterations or until convergence

Table (4) represents the outline of the implementation of SVM and MNB with firefly algorithm for co-training model.

Self-training involves using a model's own predictions to label additional unlabeled data, while co-training leverages multiple views of the data to improve model performance. Both methods enhance sentiment analysis models by incorporating unlabeled data and promoting robustness.

Self-Training for SVM with Firefly

Self-training is a semi-supervised learning approach where an initial model is trained on a small labeled dataset and then iteratively improved using predictions on a larger unlabeled dataset. In sentiment analysis, self-training will be effective when there's a small amount of labeled sentiment data (e.g., positive/negative reviews) but a large pool of unlabeled text (e.g., unlabeled reviews or social media posts). By iteratively training the model and augmenting the labeled data with high-confidence predictions, self-training can significantly improve the sentiment classifier's accuracy and generalization ability. Here the self-training model applies Support Vector Machines (SVM) for classification. Figure (2) explains the implementation details of the model. Initially train the model on the small labeled dataset. Then use the trained model to predict labels for the unlabeled data and also compute confidence scores for these predictions. Confidence scores typically measure how certain the model is about its predictions. Select only those pseudo-labels where the model's confidence is above a certain threshold. This ensures that the pseudo-labels used for retraining are likely to be more reliable. Combine the original labeled data with the high-confidence pseudo-labeled data to create an expanded training dataset. Retrain the model using this larger, more confident dataset. The process is repeated, adjusting the confidence threshold and re-evaluating the model for further improved performance. By filtering out low-confidence predictions, it avoids introducing noise into the training data, which can lead to better model performance and stability.

Co-Training for SVM and NB with Firefly

Co-training is a semi-supervised learning technique that leverages multiple views (distinct and complementary feature sets) of the data to improve the learning process. The idea is to train two or more classifiers on different views of the labeled data and iteratively enhance each classifier using confident predictions from the other classifiers on the unlabeled data. This method is particularly useful when the data can be naturally split into different sets of features that provide different perspectives on the same instance. Here the co-training model applies Multinomial Naïve Bayes and Support Vector Machines (SVM) for classification. Figure (3) explains the implementation details of the model. The initial dataset contains some labeled samples and also many unlabeled samples. The data is divided into two different views.

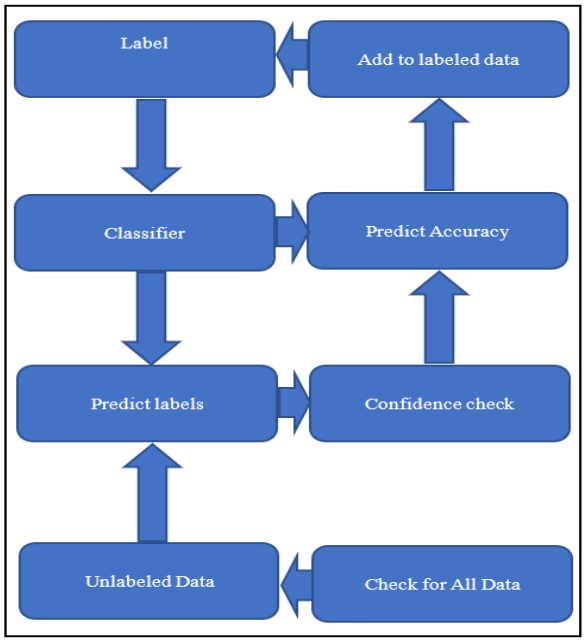

Fig. 2: Self-training flowchart

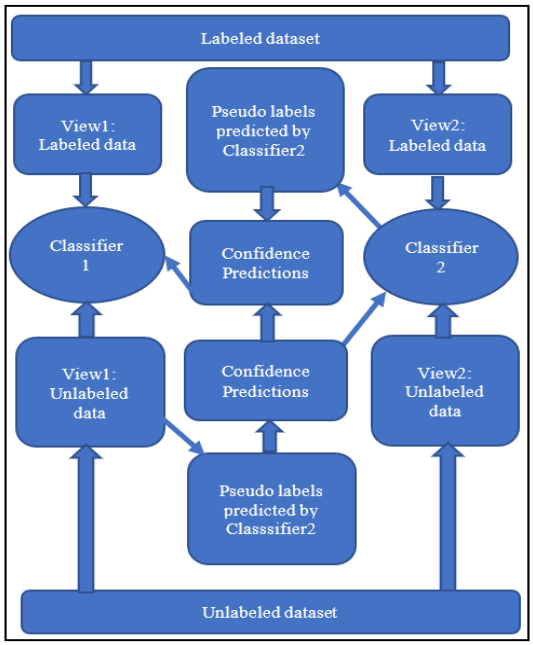

Fig. 3: Co-training flowchart

Each view contains different features but describes the same set of instances. View 1 labeled data is used to train classifier 1. View 2 labeled data is used to train classifier 2. After training, classifier 1 predicts labels for the unlabeled data in view 1. Similarly, classifier 2 predicts labels for the unlabeled data in view 2. Both classifiers select predictions they are confident about. These confident predictions are treated as pseudo-labeled data. Then classifier 1 uses its confident predictions to label data in view 2 and vice versa. This means the pseudo-labels predicted by classifier 1 for view 1 are used to create additional labeled data for view 2 and the pseudo-labels predicted by classifier 2 for view 2 are used to create additional labeled data for view 1. The newly pseudo-labeled data is added to the training set of each classifier and the classifiers are retrained. This process iterates, with each classifier helping to label more data for the other. Through this feedback loop, the classifiers gradually improve as they learn from each other's confident predictions. As the process continues, the classifiers become more accurate, leveraging both labeled and unlabeled data. The process can stop after a fixed number of iterations, or when the classifiers' performance stabilizes and no longer significantly improves. Co-training can significantly improve the performance of classifiers by making use of a large amount of unlabeled data. The co-training model works by iteratively training two classifiers on different views of the data, using confident predictions to generate pseudo-labels that help each other improve. This process effectively leverages both labeled and unlabeled data, enhancing the overall performance of the model.

In both of the model, we have used SVM classifier. We have implemented the RBF Kernel, in which the firefly algorithm is used at the parameter γ. It defines how far the influence of a single training example reaches. The firefly algorithm is used to select the most relevant features for the SVM classifier, helping to reduce dimensionality and improve model performance. Similarly, the firefly algorithm is also applied in the MNB classifier.

Results

At first, the accuracy was measured for both models. By the approach of self-training, we get an accuracy of 0.731508, whereas in co-training, we get an accuracy of 0.771472 after five iterations. The use of the firefly optimization enhanced the model's accuracy. The optimized self-training model provided an accuracy of 0.818502 on the other hand, the optimized co-training model provided an accuracy of 0.820710.

The accuracy scores of the 4 sets of classifiers for the dataset is shown in Table (5).

Figure (4) shows a bar chart comparing the accuracy of two models: One "without firefly" and another "with firefly." The y-axis represents the accuracy of the models, ranging from 0.60-1.00 and the x-axis lists the two model conditions i.e., "without firefly" and "with firefly".

Table 5: Accuracy scores for self-training and co-training model

Training models	Self-training	
Classifiers	SVM	
Optimization	Without firefly optimization	With firefly optimization
Accuracy	0.731508	0.818502
Training models	Co training	
Classifiers	Multinomial NB/ SVM	
Optimization	Without firefly optimization	With firefly optimization
Accuracy	0.771472	0.820710

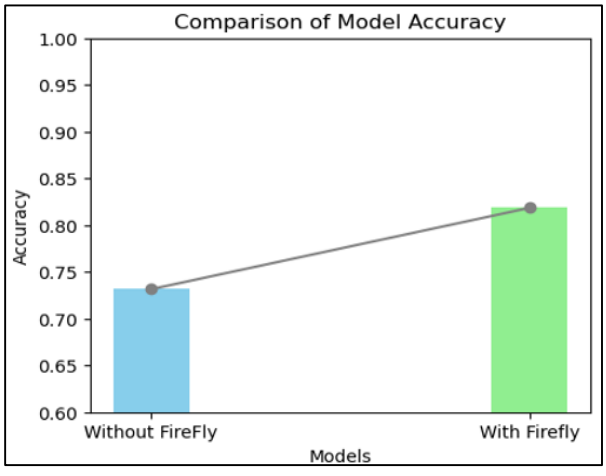

Fig. 4: Comparative bar graph of accuracy scores using self-training with SVM without the firefly optimization and with the firefly optimization

The bar representing the model "without firefly" is shown in blue and has an accuracy of 0.731508. The bar representing the model "with firefly" is shown in green and has an accuracy of 0.818502. A line connecting the tops of the two bars indicates an improvement in accuracy when using the firefly method, showing that the model's performance is better with firefly.

Figure (5) shows two confusion matrices comparing the performance of a classification model without and with the firefly technique. The confusion matrix without firefly (left) indicates that 128 instances were correctly classified as 0 (true negatives), 684 instances were incorrectly classified as 1 (false positives), 532 instances were incorrectly classified as 0 (false negatives) and 3185 instances were correctly classified as 1 (true positives). Whereas the confusion matrix with firefly (right) where 4 instances were correctly classified as 0 (true negatives), 808 instances were incorrectly classified as 1 (false positives), 14 instances were incorrectly classified as 0 (false negatives) and 3703 instances were correctly classified as 1 (true positives). The model with firefly shows an increase in true positives (from 3185-3703) and a significant decrease in false negatives (from 532-14), indicating improved sensitivity or recall. However, there is an increase in false positives (from 684-808) and a decrease in true negatives (from 128-4), indicating a reduction in specificity.

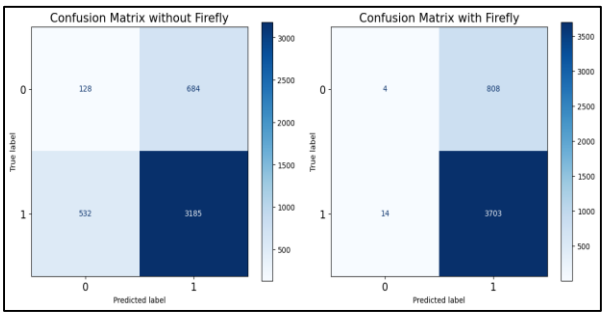

Fig. 5: Confusion matrix of self-training model with SVM without the firefly optimization and with the firefly optimization

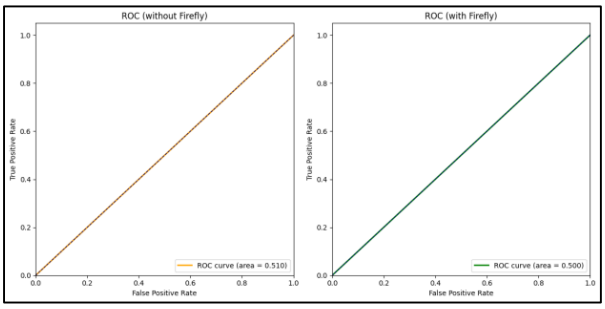

Fig. 6: ROC curves of self-training with SVM without firefly optimization and with firefly optimization

Figure (6) shows two ROC (Receiver Operating Characteristic) curves comparing the performance of a model without and with firefly. The x-axis represents the False Positive Rate (FPR). Y-axis represents the True Positive Rate (TPR). The diagonal line represents the line of no-discrimination where the true positive rate equals the false positive rate (i.e., random guessing). The ROC curves (without firefly) shown in orange indicate that, the Area Under the Curve (AUC) is 0.51, which is very close to 0.5, indicating that the model performs only slightly better than random guessing. The ROC curves (with firefly) shown in green indicate that the AUC is 0.50, which means, the model performs no better than random guessing. The ROC curve without firefly, closely follows the diagonal line and the AUC of 0.51 suggests that the model's ability to distinguish between classes is almost equivalent to random guessing and the ROC curve with firefly, exactly follows the diagonal line and the AUC of 0.50 indicates that the model is performing no better than random guessing.

Figure (7) shows two box plots comparing sentiment values without and with firefly. The box in the box plot represents the Inter Quartile Range (IQR), which contains the middle 50% of the data. The orange line inside the box indicates the median (middle value) of the data. The lines extending from the top and bottom of the box represent the range of the data excluding outliers, typically extending to 1.5 times the IQR from the box. Any points outside the whiskers would be considered outliers, but they are not shown in these plots. The box Plot (without firefly) is with a median sentiment value of around 0.75, the IQR ranges from approximately 0.6-0.9 and the whiskers extend from around 0.2-1.0. Whereas the box plot (with firefly) is with the median sentiment value is slightly higher, around 0.8, the IQR ranges from approximately 0.7-0.9 and the whiskers extend from around 0.6-1.0. So, it can be observed that the median sentiment value is slightly higher with firefly compared to without firefly. The range of the data (as shown by the whiskers) is narrower with firefly, indicating more consistent sentiment values. The IQR is also slightly narrower with firefly, suggesting less variability in the middle 50% of the data. Overall, the box plots suggest that the presence of firefly leads to slightly higher and more consistent sentiment values.

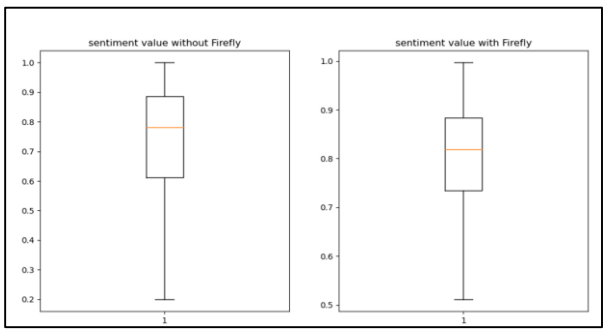

Fig. 7: Box plots of accuracy scores in self-training with SVM without firefly optimization and with firefly optimization

Figure (8) shows a scatter plot comparing sentiment values with and without Fire Fly (FF). The x-axis represents the data points (presumably sample indices or some other sequential data) and the y-axis: Represents the sentiment values, ranging from 0-1. The blue dots represent sentiment values without firefly and the orange dots represent sentiment values with firefly. The blue dots (without FF) are more spread out, especially in the lower sentiment value range (below 0.5). The orange dots (with FF) are more concentrated in the higher sentiment value range (above 0.5). There is a noticeable clustering of orange dots around the higher sentiment values (0.7-1.0). The blue dots are more dispersed across the entire range of sentiment values, including many lower sentiment values. So, without firefly the sentiment values show more variability and a significant number of lower sentiment values whereas with firefly the sentiment values are more consistently higher, with fewer lower sentiment values. Overall, the scatter plot indicates that sentiment values tend to be higher and less variable when firefly is used, while sentiment values without firefly show greater dispersion and lower values.

Figure (9) shows a bar chart comparing the accuracy of two models: One "Without firefly" and another "with firefly. "The y-axis represents the accuracy of the models, ranging from 0.60-1.00 and the x-axis lists the two model conditions i.e., "without firefly" and "with firefly". The bar representing the model "without firefly" is shown in blue and has an accuracy of 0.771472. The bar representing the model "with firefly" is shown in green and has an accuracy of 0.820710. A line connecting the tops of the two bars indicates an improvement in accuracy when using the firefly method, showing that the model's performance is better with firefly.

Figure (10) shows two confusion matrices comparing the performance of a classification model without and with the Firefly technique. The confusion matrix without Firefly (left) indicates that 77 instances were correctly classified as 0 (true negatives), 735 instances were incorrectly classified as 1 (false positives), 300 instances were incorrectly classified as 0 (false negatives) and 3417 instances were correctly classified as 1 (true positives). The confusion matrix with firefly (right) indicates, 0 instances were correctly classified as 0 (true negatives), 812 instances were incorrectly classified as 1 (false positives), 0 instances were incorrectly classified as 0 (false negatives) and 3717 instances were correctly classified as 1 (true positives).

The model with firefly shows a significant improvement in correctly classifying positive instances (true positives), increasing from 3417-3717 and a reduction in false negatives to 0, indicating perfect sensitivity or recall. However, there is an increase in false positives (from 735-812) and a decrease in true negatives to 0, indicating no specificity (the model failed to correctly identify any negative instances as negative).

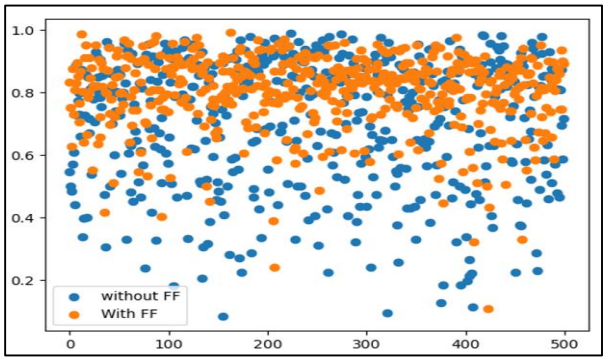

Fig. 8: Scatter plot of accuracy scores in self-training with SVM without firefly optimization and with firefly optimization

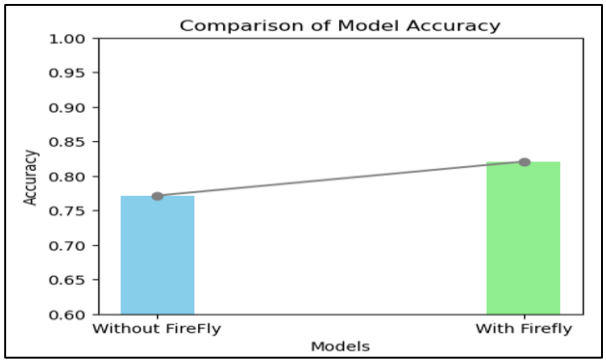

Fig. 9: Comparative bar graph of accuracy scores using co-training with SVM and NB without firefly optimization and with Firefly optimization

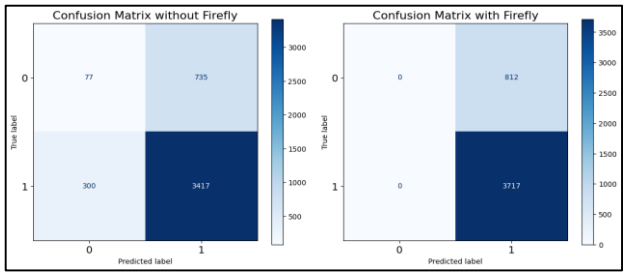

Fig. 10: Confusion matrix of accuracy scores using co-training with SVM and NB without firefly optimization and with firefly optimization

The firefly technique appears to have maximized the detection of positive cases at the expense of completely misclassifying all negative cases as positive. This trade-off indicates that the model has become very sensitive to detecting positive instances but lacks the ability to distinguish between positive and negative instances, leading to a significant increase in false positives. This outcome could be problematic depending on the application and the relative costs of false positives versus false negatives.

Figure (11) shows the AUC for both models is approximately 0.5 (0.511 without firefly and 0.5 with firefly). An AUC of 0.5 represents the performance of a

random classifier. Both ROC curves are essentially diagonal lines from the bottom-left to the top-right corner. So, the improvement in the model performance is not noticeable significantly. For this reason, the analysis is expressed again with the box plot.

Figure (12) shows the median sentiment value (the orange line inside the box) is slightly higher with firefly than without firefly. The Inter Quartile Range (IQR) represents the middle 50% of the data (from the first quartile to the third quartile) and is slightly narrower with firefly, suggesting that the sentiment values are more tightly clustered around the median.

The range (distance between the minimum and maximum values) seems similar in both cases, indicating that the overall spread of the data is consistent with and without firefly. So, the sentiment analysis indicates that with firefly, the sentiment values are slightly higher on average and show a bit less variability, suggesting a more consistent positive sentiment.

Figure (13) shows the scatter plot and compares two sets of data points. The sentiment values without firefly (blue dots) and with firefly (orange dots). Both sets of data points are spread across the y-axis, ranging from 0-1, indicating a full spectrum of sentiment values.

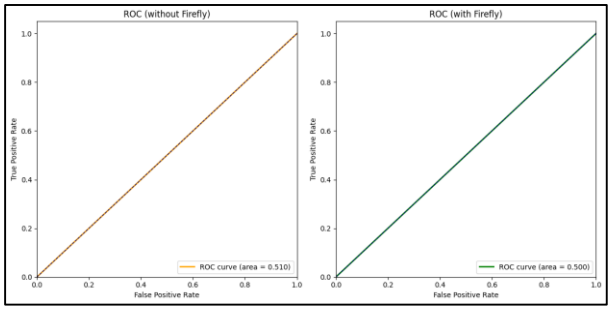

Fig. 11: ROC curves of accuracy scores using co-training with SVM-NB without firefly optimization and with firefly optimization

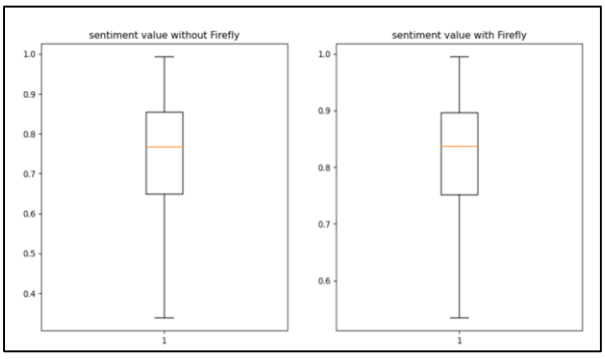

Fig. 12: Box plots of accuracy scores using co-training with SVM-NB without firefly optimization and with firefly optimization

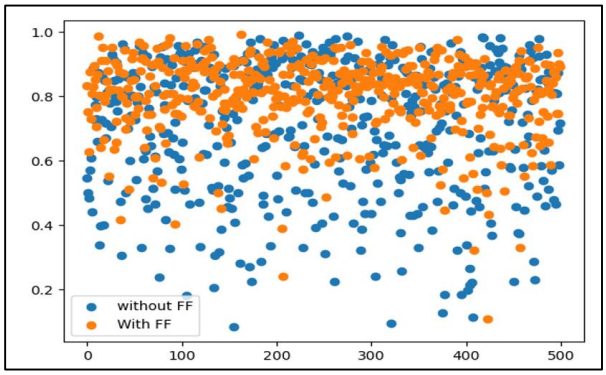

Fig. 13: Scatter plot of accuracy scores using co-training with SVM and NB without firefly optimization and with firefly optimization

The sentiment values with firefly (orange dots), the data points appear to be more concentrated in the upper range (around 0.6-1.0), suggesting generally higher sentiment. The sentiment values without firefly (blue dots), the data points are more spread out across the y-axis, including more lower sentiment values (below 0.6). There is a noticeable overlap between the two sets of data points, but the density of orange dots is higher in the upper range compared to the blue dots.

Discussion

The performance of the models was analyzed using confusion matrices, ROC curves, box plots, and scatter plots, highlighting the positive impact of firefly optimization. The confusion matrices (Figures 5 and 10) showed that firefly optimization significantly improved sensitivity by increasing true positives and reducing false negatives, enhancing the model's ability to detect positive cases effectively. Overall, the ROC curves and AUC values indicate that the models, both with and without firefly, do not have good discriminative power and perform nearly at the level of random guessing, for which the study further analyzed with box plot and scatter plot.

The box plots (Figures 7 and 12) demonstrated that firefly optimization resulted in higher median sentiment values and narrower interquartile ranges, reflecting more consistent predictions with reduced variability. This indicates improved stability and reliability in sentiment predictions. Scatter plots (Figures 8 and 13) further supported these findings, showing that sentiment values with firefly optimization were concentrated in the higher range (0.6-1.0), illustrating its capability to produce focused and higher-value predictions. The use of firefly appears to result in higher and more consistent sentiment values. Without firefly, the sentiment values show greater variability and include more lower values. In total, the scatter plot suggests that firefly tends to produce higher and more consistent sentiment values, while without firefly, the sentiment values are more varied and include lower values.

Overall, firefly optimization proved effective in enhancing sensitivity and improving the consistency of sentiment values, providing a solid foundation for further advancements in the models' development.

Conclusion

Nowadays most people are influenced directly or indirectly by the use of sentiment analysis. The sentiment analysis task is comparatively easy when it deals with labeled data. But practically in real-time, it is hard to get the labeled dataset. To address the issue, semi-supervised models play a vital role. This study highlights the comparative analysis between two popular semi-supervised approaches, self-training and co-training. The substantial improvement in accuracy when using firefly optimization suggests that the optimization technique is effectively tuning the parameters of the classifiers. This can lead to better generalization on unseen data, which is crucial for real-world applications. The model comparison shows that firefly optimization works well across different training models and classifiers: For self-training with SVM, accuracy improves by approximately 8.7%. For co-training with Multinomial NB/SVM, accuracy improves by approximately 4.9%. This consistency across different setups indicates that firefly optimization is robust and versatile. The results demonstrate that this optimization technique can enhance the learning process of classifiers, making them more accurate and reliable. In practical scenarios, higher accuracy means better predictions and decisions. For applications in fields such as finance, healthcare and security, even a small increase in accuracy can lead to significantly better outcomes. The use of firefly optimization can be particularly beneficial when working with large datasets and complex models, where manual tuning of parameters is challenging. In conclusion, the model demonstrates that integrating firefly optimization into the training process of classifiers leads to substantial improvements in accuracy, making it a valuable technique for enhancing machine learning models. The self-training model is implemented with the SVM classifier and the model accuracy is measured as 0.731508. Whereas, in the other hand, the co-training model is implemented with SVM and Multinomial NB classifier and observed the model performance with an accuracy of 0.771472. Sometimes it is also essential to fine-tune the model for its best performance. As a result, we have implemented the firefly optimization algorithm with both of the models. After optimization, the self-training model performed with an accuracy of 0.818502, whereas, the co-training model performed with an accuracy of 0. 820710. As a result, we can conclude that our co-training model performs better with SVM and MNB classifiers than the other models. Our research adds to the expanding body of knowledge in the fields of semi-supervised learning and sentiment analysis by offering insightful information on the potential of co-training methods and the importance of classifier choice. Our work provides a valuable roadmap for improving sentiment analysis accuracy in a domain where customer feedback is crucial. As natural language processing technology advances, accurate sentiment assessments become increasingly important for determining a company's success.

Acknowledgment

This research was conducted under the supervision of Dr. K. M. Gopal, who served as the supervisor, and Dr. Abinash Tripathy, who served as the co-supervisor.

Funding Information

This research was conducted without any external funding.

Author’s Contributions

Alok Kumar Jena: Conceptualization, writing, and edited.
Kakita Murali Gopal: Reviewing and corrections.
Abinash Tripathy: Data analysis.
Nibedan Panda: Methodology.

Ethics

The authors confirm that the work presented in this publication is original and has not been previously published. There are no ethical concerns, as the paper has been reviewed and approved by all authors.

References

Ahmad, G. I., Singla, J., Ali, A., Reshi, A. A., & Salameh, A. A. (2022). Machine Learning Techniques for Sentiment Analysis of Code-Mixed and Switched Indian Social Media Text Corpus - A Comprehensive Review. *International Journal of Advanced Computer Science and Applications*, *13*(2), 455–467. https://doi.org/10.14569/ijacsa.2022.0130254

Aribowo, A. S., Basiron, H., & Abd Yusof, N. F. (2022). Semi-Supervised Learning for Sentiment Classification with Ensemble Multi-Classifier Approach. *International Journal of Advances in Intelligent Informatics*, *8*(3), 349–361. https://doi.org/10.26555/ijain.v8i3.929

Annamalai, P. (2020). Automatic Face Recognition Using Enhanced Firefly Optimization Algorithm and Deep Belief Network. *International Journal of Intelligent Engineering and Systems*, *13*(5), 19–28. https://doi.org/10.22266/ijies2020.1031.03

Benlahbib, A., & Nfaoui, E. H. (2020). A Hybrid Approach for Generating Reputation Based on Opinions Fusion and Sentiment Analysis. *Journal of Organizational Computing and Electronic Commerce*, *30*(1), 9–27. https://doi.org/10.1080/10919392.2019.1654350

Braig, N., Benz, A., Voth, S., Breitenbach, J., & Buettner, R. (2023). Machine Learning Techniques for Sentiment Analysis of COVID-19-Related Twitter Data. *IEEE Access*, *11*, 14778–14803. https://doi.org/10.1109/access.2023.3242234

Cheng, X., Xu, W., Wang, T., Chu, W., Huang, W., Chen, K., & Hu, J. (2019). Variational Semi-Supervised Aspect-Term Sentiment Analysis via Transformer. *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 961–969. https://doi.org/10.18653/v1/k19-1090

Duan, J., Luo, B., & Zeng, J. (2020). Semi-Supervised Learning with Generative Model for Sentiment Classification of Stock Messages. *Expert Systems with Applications*, *158*, 113540. https://doi.org/10.1016/j.eswa.2020.113540

Dangi, D., Dixit, D. K., & Bhagat, A. (2022). Sentiment Analysis of COVID-19 Social Media Data Through Machine Larning. *Multimedia Tools and Applications*, *81*(29), 42261–42283. https://doi.org/10.1007/s11042-022-13492-w

Elangovan, D., & Subedha, V. (2023a). Firefly with Levy Based Feature Selection with Multilayer Perceptron for Sentiment Analysis. *Journal of Advances in Information Technology*, *14*(2), 342–349. https://doi.org/10.12720/jait.14.2.342-349

Elangovan, D., & Subedha, V. (2023b). Automatic Sentimental Analysis by Firefly with Levy and Multilayer Perceptron. *Computer Systems Science and Engineering*, *46*(3), 2797–2808. https://doi.org/10.32604/csse.2023.031988

Fang, H., Jiang, G., & Li, D. (2022). Sentiment Analysis Based on Chinese BERT and Fused Deep Neural Networks for Sentence-Level Chinese E-Commerce Product Reviews. *Systems Science and Control Engineering*, *10*(1), 802–810. https://doi.org/10.1080/21642583.2022.2123060

Gupta, R., Sahu, S., Espy-Wilson, C., & Narayanan, S. (2018). Semi-Supervised and Transfer Learning Approaches for Low Resource Sentiment Classification. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5109–5113. https://doi.org/10.1109/icassp.2018.8461414

Hilal, A. M., Alzahrani, J. S., Alsolai, H., Negm, N., Nafie, F. M., Motwakel, A., Yaseen, I., & Hamza, M. A. (2023). Sentiment Analysis Technique for Textual Reviews Using Neutrosophic Set Theory in the Multi-Criteria Decision- Making System. *Human-Centric Computing and Information Sciences*, *13*(24), 31–45.

Jemai, F., Hayouni, M., & Baccar, S. (2021). Sentiment Analysis Using Machine Learning Algorithms. *2021 International Wireless Communications and Mobile Computing (IWCMC)*, 775–779. https://doi.org/10.1109/iwcmc51323.2021.9498965

Kandhro, I. A., Wasi, S., Kumar, K., Malook, R., & Ameen, M. (2019). Sentiment Analysis of Students Comment by using Long-Short Term Model. *Indian Journal of Science and Technology*, *12*(8), 1–16. https://doi.org/10.17485/ijst/2019/v12i8/141741

Kim, K. (2018). An Improved Semi-Supervised Dimensionality Reduction Using Feature Weighting: Application to Sentiment Analysis. *Expert Systems with Applications*, *109*, 49–65. https://doi.org/10.1016/j.eswa.2018.05.023

Kumar, A., & Jain, R. (2015). Sentiment Analysis and Feedback Evaluation. *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, 433–436. https://doi.org/10.1109/mite.2015.7375359

Kumar, H. M. K., & Harish, B. S. (2020). A New Feature Selection Method for Sentiment Analysis in Short Text. *Journal of Intelligent Systems*, *29*(1), 1122–1134. https://doi.org/10.1515/jisys-2018-0171

Madhoushi, Z., Hamdan, A. R., & Zainudin, S. (2023). Semi-Supervised Model for Aspect Sentiment Detection. *Information*, *14*(5), 293. https://doi.org/10.3390/info14050293

Macrohon, J. J. E., Villavicencio, C. N., Inbaraj, X. A., & Jeng, J.-H. (2022). A Semi-Supervised Approach to Sentiment Analysis of Tweets during the 2022 Philippine Presidential Election. *Information*, *13*(10), 484. https://doi.org/10.3390/info13100484

Ng, C., Lam, S., & Liu, K. (2022). Sentiment Analysis on Consumers' Opinions – Evaluating Online Retailers through Analyzing Sentiment for Face Masks During COVID-19 Pandemic. *Journal of Industrial and Production Engineering*, *39*(7), 535–551. https://doi.org/10.1080/21681015.2022.2070933

Nemes, L., & Kiss, A. (2021). Social Media Sentiment Analysis Based on COVID-19. *Journal of Information and Telecommunication*, *5*(1), 1–15. https://doi.org/10.1080/24751839.2020.1790793

Pan, Y., Chen, Z., Suzuki, Y., Fukumoto, F., & Nishizaki, H. (2020). Sentiment Analysis Using Semi-Supervised Learning with Few Labeled Data. *2020 International Conference on Cyberworlds (CW)*, 231–234. https://doi.org/10.1109/cw49994.2020.00044

Pandya, V., Somthankar, A., Shrivastava, S. S., & Patil, M. (2021). Twitter Sentiment Analysis using Machine Learning and Deep Learning Techniques. *2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4)*, 1–5. https://doi.org/10.1109/c2i454156.2021.9689241

Shan Lee, V. L., Gan, K. H., Tan, T. P., & Abdullah, R. (2019). Semi-supervised Learning for Sentiment Classification using Small Number of Labeled Data. *Procedia Computer Science*, *161*, 577–584. https://doi.org/10.1016/j.procs.2019.11.159

Salimpour, S., Kalbkhani, H., Seyyedi, S., & Solouk, V. (2022). Stockwell Transform and Semi-Supervised Feature Selection from Deep Features for Classification of BCI Signals. *Scientific Reports*, *12*(1), 11773. https://doi.org/10.1038/s41598-022-15813-3

Swapnarekha, H., Nayak, J., Behera, H. S., Dash, P. B., & Pelusi, D. (2023). An Optimistic Firefly Algorithm-Based Deep Learning Approach for Sentiment Analysis of COVID-19 Tweets. *Mathematical Biosciences and Engineering*, *20*(2), 2382–2407. https://doi.org/10.3934/mbe.2023112

Tanha, J., Mahmudyan, S., & Farahi, A. (2021). A hybrid semi-supervised boosting to sentiment analysis. *International Journal of Nonlinear Analysis and Applications*, *12*(2), 1769–1784. https://doi.org/10.22075/ijnaa.2021.23334.2522

Zhou, J., & Ye, J. (2023). Sentiment Analysis in Education Research: A Review of Journal Publications. *Interactive Learning Environments*, *31*(3), 1252–1264. https://doi.org/10.1080/10494820.2020.1826985

Zou, H., & Wang, Z. (2023). A Semi-Supervised Short Text Sentiment Classification Method Based on Improved Bert Model from Unlabeled Data. *Journal of Big Data*, *10*(1), 35. https://doi.org/10.1186/s40537-023-00710-x