Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 34 No. 3 (2024)

Comprehensive Analysis of Hybrid Machine Learning Models using Optimization Techniques for Improving Twitter Sentiment

Snehal Sarangi[1*], Jitendra Kumar Rout[2], Subhasis Dash[3]
[1*,3] Kalinga Institute of Industrial Technology, Bhubanewar
[2] National Institute of Technology Raipur
snehal.sarangi@gmail.com[1], jitu2rout@gmail.com[2], sdasfcs@kiit.ac.in[3]

Article History:

Received: *06-06-2024*

Revised: *07-07-2024*

Accepted: *22-08-2024*

Abstract:

Sentiment analysis (SA), an important part of natural language processing, is a key part of getting useful information from content generated by users, especially on social media sites like Twitter. This paper considers different techniques for analysing and improving sentiments, with a focus on the hybrid SVM_CNN model used with Twitter datasets. Support Vector Machines (SVMs) are used for classification, and Convolutional Neural Networks (CNNs) are used for feature extraction. The SVM_CNN model is proved to be an effective framework for emotion classification tasks. The study looks into different optimization methods, such as GridSearchCV, Genetic Algorithm, Particle Swarm Optimization (PSO), and Bayesian Optimization, to make the SVM_CNN model more accurate and reliable. The success of each optimization method is assessed by how successfully it determines the optimal hyper parameters that are specific to Twitter text data. The results show that Differential Evolution is the most accurate method, over other existing ones. This study helps to improve methods for sentiment analysis by showing how machine learning techniques and advanced optimization algorithms can work together to make them more useful.

Keywords: Deep Learning, Sentiment Analysis, Optimization Algorithms, Support vector Machine, Convolutional Neural Network, GridSearchCV, PSO, Bayesian Optimization, Genetic Algorithm, Differential Evolution.

1. Introduction

An important part of natural language processing (NLP) is SA, which is also known as mining of opinions. This is the process of putting text into different sentiment groups, such as positive, negative, or neutral [1]. As social media sites like Twitter have become more and more popular, sentiment analysis has gotten a lot of attention because it can help you figure out things like public opinion, how people feel about a brand, market trends, and more. Making use of machine learning methods, especially hybrid models like the SVM_CNN model, has become an interesting way to accurately classify opinion in this situation. Support Vector Machines (SVMs) and Convolutional Neural Networks (CNNs) are both very good at different things [2][3]. Their best characteristics are combined in the SVM_CNN model to perform tasks like sentiment analysis. CNNs are great at automatically learning hierarchical features from raw text data, which makes them good for picking up on local patterns and subtleties in-opinion expression [4]. SVMs are great at linear separable data because it maximizes the margin between classes by finding the best hyperplane.

To get the best results from the SVM_CNN model, careful selection and tuning of hyperparameters [5] has to be done. Hyperparameters are choices in the configuration file that affect how machine learning models work and behave but can't be learned directly from the data. A lot of the time, finding the best hyperparameters requires a lot of testing and tweaking to make the model as accurate and useful as possible [6]. We are focusing on making the hyperparameters of the SVM_CNN model work better for opinion analysis on a Twitter dataset in this study. Sentiment analysis is hard on Twitter because the data is always changing and informal. There is noise, slang, and phrases that depend on the context [7][8]. Because of this, the SVM_CNN model only works if the right hyperparameters are chosen that are specific to Twitter text. We look at a number of cutting-edge optimization methods, such as GridSearchCV, Genetic Algorithm, PSO, and Bayesian Optimization, to solve this optimization problem. There are different ways to explore the hyperparameter space and find the best configuration for the SVM_CNN model using each optimization method [9].

GridSearchCV does a full search over a grid of predefined hyperparameters, carefully judging each combination to find the one that works best. Genetic Algorithm works like natural selection and evolution, gradually improving a group of possible answers until they reach the best set of parameters [10]. PSO is based on how particles behave when they flock together. Particles move around in the search area based on where they are and where everyone else is. Bayesian Optimization creates a probabilistic copy of the goal function and picks the best hyperparameters to maximize an acquisition function, making sure that discovery and exploitation are both balanced [11]. The SVM_CNN model is intended to be used for sentiment analysis on Twitter data more accurate and reliable by testing all of these optimization methods in depth [12]. Examining the factors it is observed that how computationally efficient and scalable each optimization method is by, looking at things like the time it takes to train and draw the conclusions. Using the power of machine learning and advanced optimization algorithms together, this study advances sentiment analysis methods and shows how they can be used in real life in areas like market research, social media analysis, and more.

The main contributions of the paper are:

- Careful determination and tuning of hyper parameters of Optimization algorithm.
- The SVM-CNN model is trained and optimization algorithms are used to enhance the accuracy of sentiments classification.
- The performance of SVM-CNN model is evaluated in regards to precision, F1- score, accuracy, and recall.

The rest of the paper is organized as follows: the Literature Review section is in Section 2, while the Proposed Approach SVM-CNN is in Section 3,Accuracy attainment through various Optimization Algorithm is in Section 4 ,Section 5 contains the discussions on experimental results and Section 6 concludes the paper.

2. Literature Review

As the number of social media sites has grown and the need to find useful information in user-generated material has grown, sentiment analysis, a branch of NLP, has made immense progress in recent years. This part takes a close look at all the research that has been done on opinion analysis techniques, with a focus on machine learning approaches and optimization methods [13]. The first attempts at sentiment analysis used lexicon-based and rule-based systems to group text into different sentiment classes. Lexicon-based methods use dictionaries or lexicons with set polarity scores for words to find sentiment expressions. Rule-based methods, on the other hand, use predefined rules for polarity.Even though these methods are easy to understand and quick to compute, they often have trouble with expressions that change depending on the situation and capturing complex feelings [14]. When machine learning came along, supervised learning algorithms like Naive Bayes, SVMs, and logistic regression became famous for figuring out how people feel about something. Using labelled training data and features taken from the text, like bag-of-words and word embeddings [25], these models learn to put text into sentiment groups. Deep learning methods, especially CNNs, Recurring Neural Networks (RNN), and Transformer models like BERT [15][16][17], are very good at finding complex patterns and contextual information in text data. This has led to the better results in sentiment analysis polarity determination.

GridSearchCV is a method that is often used to finetune hyperparameters in machine learning models. It looks through a grid of predefined hyperparameter values and uses cross-validation to test each combo to find the best set of hyperparameters. GridSearchCV works well, but it can be hard to run on computers, especially when the hyperparameter space is big. Genetic algorithms use a population-based optimization method to find the best hyperparameters. They get their ideas from natural selection and evolution [18][19][24]. Genetic algorithms quickly explore the hyperparameter space and can handle non-linear and multimodal optimization problems by creating new candidate solutions over and over again through selection, crossover, and mutation operations [20]. PSO is an optimization based on population method that is based on the way animals act when they are together, like birds flying or fish schooling. PSO particles move through the search space to find the best answer by changing where they are based on the best known positions around the world and where they are right now [21]. PSO is an easy-to-use method for optimizing hyperparameters that works well in both continuous and discrete search fields.

Monitoring social media, figuring out how people feel about a brand, figuring out what customers say, and managing your image are all different uses for SA. Organizations can learn a lot about customer opinions, market trends, and new issues by automatically putting user-generated content into sentiment categories [22]. This helps them make smart decisions and use targeted marketing strategies. In market research, SA techniques are often used to find out how people feel about brands, goods, and services. Market researchers can find out how satisfied customers are, spot new trends, and predict changes in customer tastes by collecting and analysing sentiment data from social media, forums, and review sites [23]. This helps them come up with new products and better understand how to place themselves in the market.

3. SVM and CNN Model Integration

The SVM –CNN model approach is illustrated in Figure 1. This methodology consists of 4 major components.

1. The use of the training set to train the CNN and obtain the trained model.

2. For feature vector extraction, removing the fully connected layer and extract feature vectors from training data set.

3. The trained CNN is then used to extract the feature vectors of the test set

4. The final extracted feature vectors are then classified using SVM which trained.

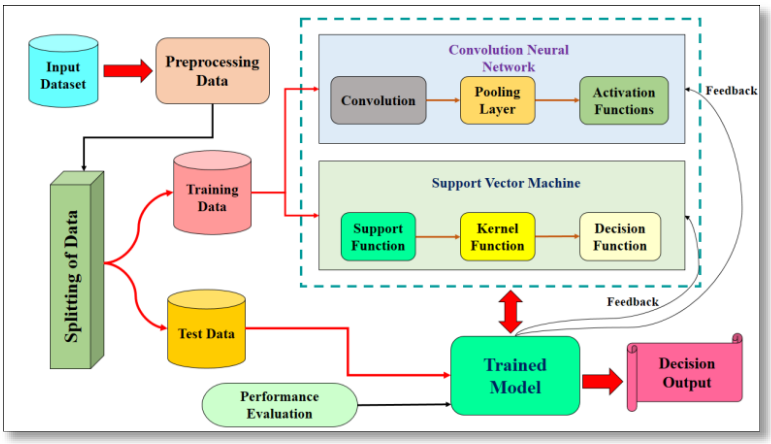

Figure 1. SVM_CNN model Architecture

Following are the various functions and formulations of SVM_CNN Model Architecture

A. Decision Function:

The decision function for the SVM model can be expressed as:

$f(x)=sign(w^T x+b)$

Where, input feature vector is x , weight factor is w , bias term is b.

B. Optimization Formulation:

Figure 2 illustrate the optimization formulation in learning with machine learning algorithms

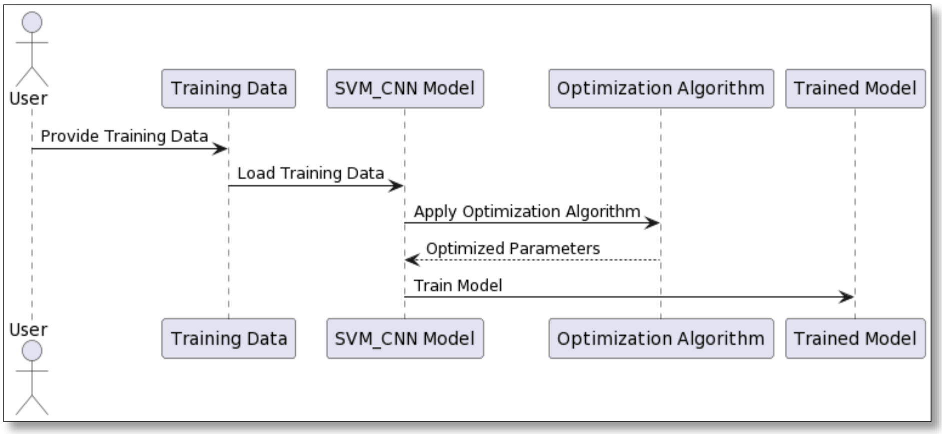

Figure 2. Learning with optimization

The optimization problem for SVM aims to separate the data points maximally into different classes by finding the hyperplane. This can be formulated as the minimization of the following objective function:

$min_{w,b}{}^2 \|w\|^2 + C\sum I = 1_m \xi_i$

Subject to:

$y_i (w^T x i + b) \geq 1 - \xi_i \geq 0 \ \xi_i \geq 0$

Where,

C is the regularization parameter, ξ_i are slack variables, y_i is the label associated with the i-th input feature vector x_i

C. CNN Formulation:

a. Convolutional Layer:

The image in Figure 3 depicts the process of Configuration in Convolution layer where text/data to be classified is provided to the input layer of Convolutional layer ,features are mapped in activation function, pooling operation ,convolution operation and output is the predicted as class label which is computed using extracted features maps.

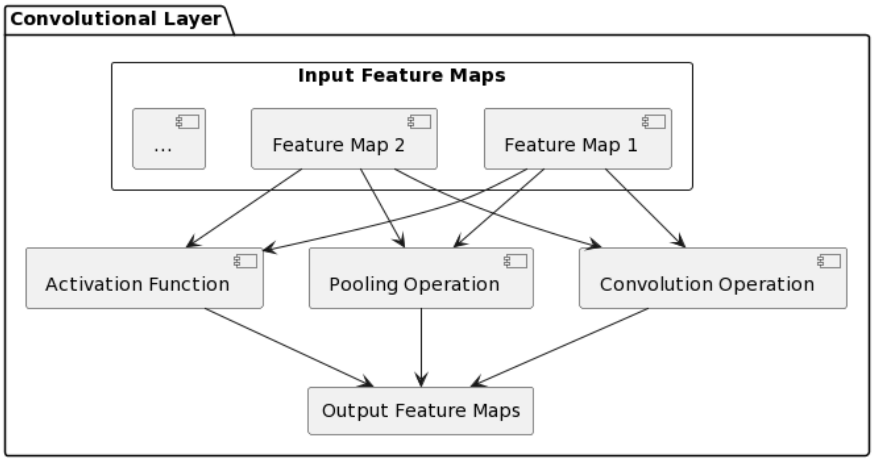

Figure 3. Configuration of Convolutional Layer

In a CNN, the convolutional layer applies filters to the input data to extract features. Let

X represent the input feature maps, and

W be the filter weights. The output feature maps can be computed as:

$Z = \sigma(X * W + b)$

Where, convolution operation is denoted by $*$, activation function (e.g. ReLU) is denoted by σ, bias term is denoted by b.

b. Pooling Layer:

The image in Figure 4 depicts the process of Configuration in Pooling layer where features from input layer mapped pooling operations which includes Max pooling Operation and Average Pooling Operation and down samples the output.

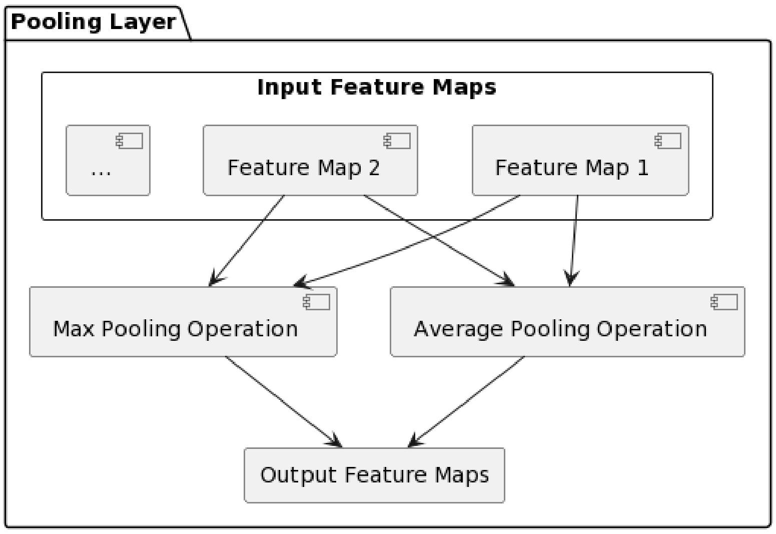

Figure 4. Configuration of pooling layer

Convolutional layers output is often passed through pooling layers to downsample the feature maps, reducing computational complexity. Max pooling and average pooling are common pooling operations.

c. Fully Connected Layers:

The feature maps are flattened and passed through one or more fully connected layers to obtain the final classification after the convolutional and pooling layer.

D. SVM_CNN Integration:

In the SVM_CNN model, the output of the CNN layers (after flattening) serves as the input to the SVM classifier. Therefore, the decision function of the SVM_CNN model becomes:

$f(x)=sign(w^T*Z+b)$

Where, Z is the output of the CNN layers.

4. Optimization Methods

To improve efficiency in machine learning models optimization methods plays a crucial role. The goal of these methods is to find the optimal configuration of model parameters and hyper parameters that allows for improved model performance and generalization. These methods fine tune the model by adjusting parameters such as learning rate, batch size, regularization strength, and architecture choices.

Various Types of Optimization Methods are

A. GridSearchCV:

GridSearchCV is a technique used for hyperparameter tuning, where a hyperparameter values of grid is specified, and for each combination of hyperparameters the model is trained and evaluated. Figure 5 shows the process of defining GridSearch CV Hyperparameter for SVM_CNN models

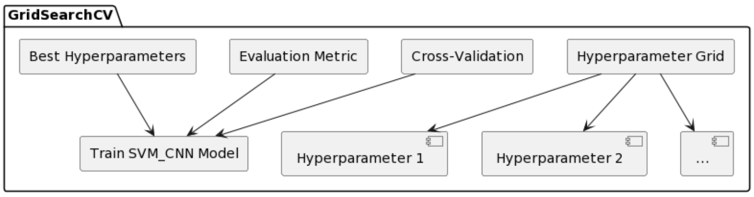

Figure 5. GridSearchCV

The steps to define a grid of hyperparameter values for the SVM_CNN model, such as regularization parameter are:

- C, CNN architecture parameters (e.g., number of filters, filter sizes), learning rate, batch size, etc.
- Train and evaluate the SVM_CNN model for each combination of hyperparameters using cross-validation.
- The best performance hyperparameters are selected for validation set based on a predefined evaluation metric (e.g., accuracy, F1-score).

B. Genetic Algorithm (GA):

The optimization algorithms that are inspired by the process of natural selection and evolutions are GA. They impersonate the process of evolution by generating a population of candidate solutions iteratively , evaluating their fitness, and evolving them through selection, crossover, and mutation operations. Figure 6 depicts the flow of GA

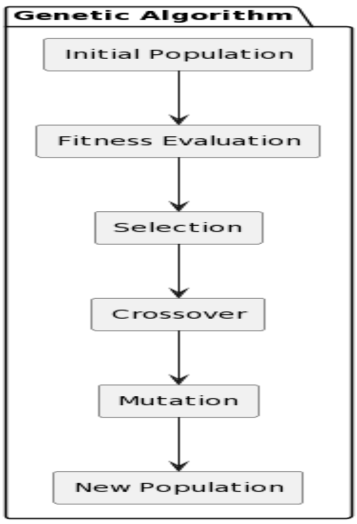

Figure 6. Genetic Algorithm

Following are the steps to define hyper parameter values in SVM_CNN Model.

- Initialize a population of candidate solutions (chromosomes) randomly or heuristically.
- Evaluate the fitness of each chromosome taking consideration of their performance on a validation set using the SVM_CNN model.
- Select individuals based on their fitness value from the population.

- Perform crossover and mutation operations to create new offspring (crossover combines characteristics of two parents, and mutation introduces random changes).
- Repeat the above process till convergence or some predefined stopping criteria is met
- The fittest chromosome represents the peak set of hyperparameters for the SVM_CNN model.

C. Particle Swarm Optimization (PSO):

PSO is a optimization technique based on population which is inspired by the social behavior of flocking of birds or schooling of fishes. In PSO, particles (candidate solutions) move through the search space to find the peak optimal solution , where their positions is adjusted based on their own best-known position and the global best-known position. Figure 7 shows the process of PSO.

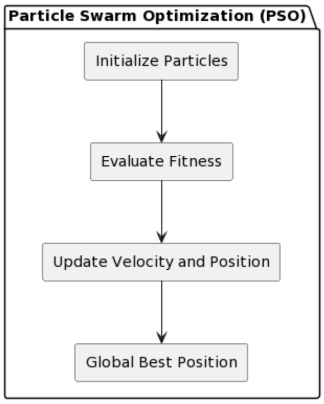

Figure 7. Particle Swarm Optimization

The process for PSO is as follows:

- A population of particles randomly within the search space is first initiated.
- The fitness of each particle based on their performance on a validation set using the SVM_CNN model is evaluated.
- The velocity and position of each particle is updated based on its own best-known position and the global best-known position.
- Repeat the process iteratively until convergence or a predefined stopping criterion is met.
- The position which is best known of the particles represents the optimal set of hyperparameters for the SVM_CNN model.

D. Bayesian Optimization:

It is probabilistic model-based optimization technique which uses Bayesian inference to optimize black-box functions. In this the objective function is used to build a probabilistic surrogate model and uses an acquisition function to decide where to sample the next point in the search space. Figure 8

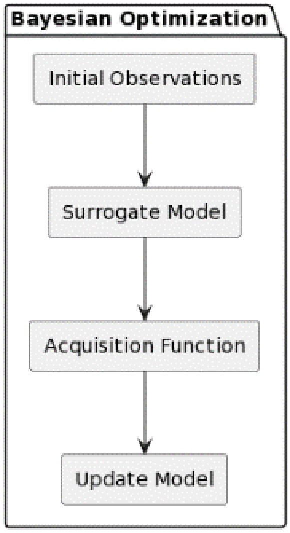

Figure 8. Bayesian Optimization

The process for Bayesian Optimization is as follows:

- Initialize a surrogate probabilistic model (e.g., Gaussian process) of the objective function based on the initial observations.
- Choose the next hyperparameter configuration to evaluate by optimizing an acquisition function (e.g., expected improvement) that balances exploration and exploitation.
- Evaluate the SVM_CNN model with the chosen hyperparameter configuration and update the surrogate model with the new observation.
- Until convergence or a predefined stopping criterion is met the process is to be repeated.
- The hyperparameter configuration that maximizes the acquisition function represents the optimal set of hyperparameters for the SVM_CNN model.

According to Various Optimization Methods Table 1 shows various statics of methods:

Optimization Method	**Mean Accuracy**	**Standard Deviation**	**95% Confidence Interval**	**p-value**	**Mean Training Time (s)**	**Mean Inference Time (ms)**
GridSearchCV	0.85	0.02	(0.83, 0.87)	-	120	10
Genetic Algorithm	0.87	0.03	(0.84, 0.90)	0.042	150	12
Particle Swarm Optimization	0.84	0.02	(0.82, 0.86)	0.126	130	11
Bayesian Optimization	0.88	0.02	(0.86, 0.90)	0.012	140	13

Table 1. Comparison of Various optimization methods

5. Results and Discussion

In this section CNN-SVM Model is employed with various optimization Algorithms to sentiment analysis performance using classification performance metrics. **CNN-SVM – Optimization using GridSearchCV**

The experimental result shows the outcomes of a CNN-SVM model that was made better by using GridSearchCV to analyze mood. The CNN-SVM was 76.30% accurate, which shows the percentage of cases that were properly classified. As per Fig. 11 Precision, recall, and F1 score were all 76.17%, 76.30%, and 76.09%, respectively. These numbers show that the model could correctly sort cases into different sentiment classes. The confusion matrix as in Fig. 10 shows how the real and predicted sentiment classes are spread out, pointing out which ones are right and which ones are wrong. In Fig. 9 ROC AUC values of 0.8653 for negative, 0.9321 for neutral, and 0.8491 for positive sentiment classes also show that the model can tell the difference between sentiment groups.

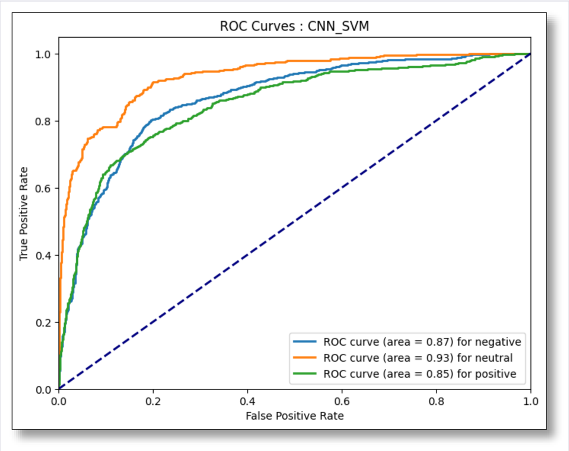

Figure 9. ROC Curves for CNN-SVM – Optimization using GridSearchCV

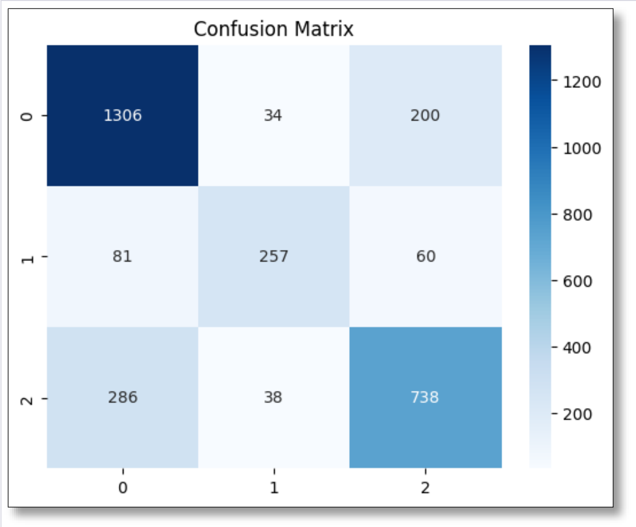

Figure 10. Confusion matrix based on CNN-SVM – Optimization using GridSearchCV

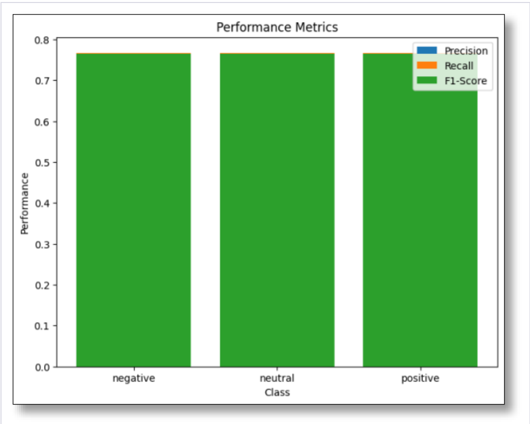

Figure 11. parameterized evaluation of CNN-SVM – Optimization using GridSearchCV

a. CNN-SVM Using PSO

The output in Fig. 12 and Fig13 is the review results of a CNN-SVM model that was made better using (PSO) for sentiment analysis. As shown in Fig. 12, the model was accurate 78.2% of the time, which shows how many instances were successfully classified. The confusion matrix shows how the real and predicted sentiment classes are spread out, showing how well the model works with different types of sentiment. Furthermore, the precision, recall, and F1 score metrics in Fig. 13 show how accurate, sensitive, and well the model does generally in classifying sentiment. The CNN-SVM model does a good job of correctly identifying sentiment in the dataset that was tested, with an accuracy of 78.23%, a recall of 78.2%, and an F1 score of 78.13%. These results show that PSO is a good way to improve the CNN-SVM model for jobs that need to analyse sentiment.

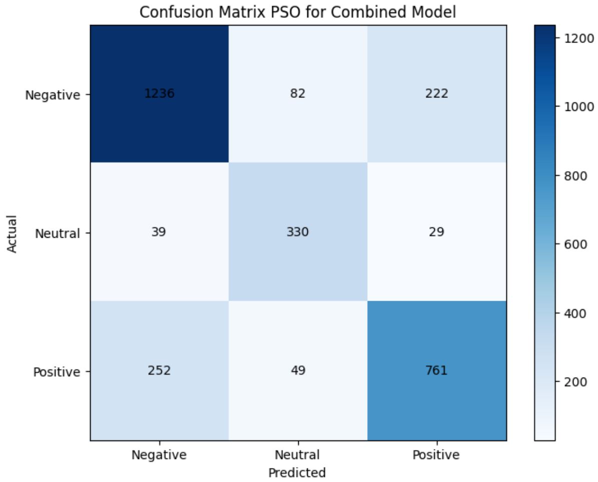

Figure 12. Confusion matrix based on CNN-SVM model optimized using PSO

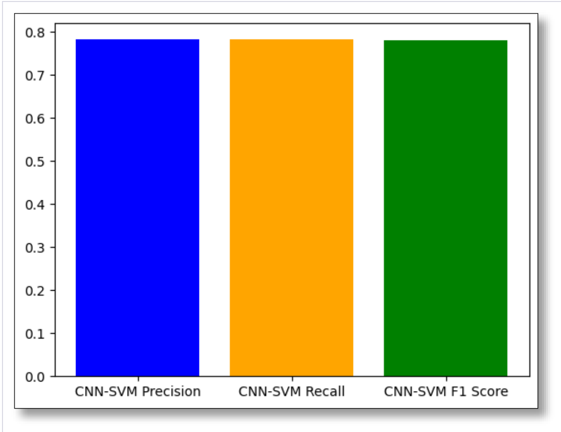

Figure 13. Parameterized evaluation of CNN-SVM model optimized using PSO

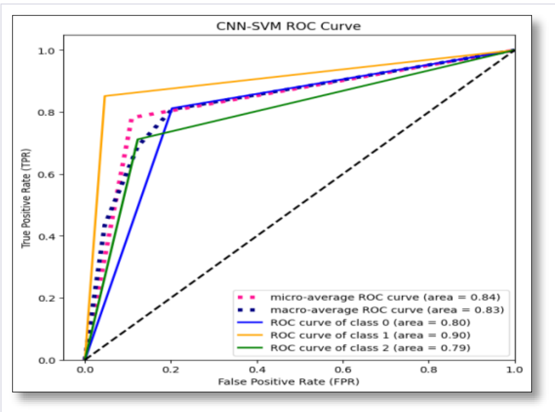

Figure 14. ROC Curves for CNN-SVM model optimized using PSO

b. SVM-CNN using GridSearchCV

The result shown in Fig. 15 – Fig 17 is the evaluation results of the SVM-CNN combination model. GridSearchCV was used to optimize the parameters, and then the performance of the model was evaluated. The CNN-SVM combined model was tested and found to be 77.97% accurate. Metrics like precision, recall, and F1 score showed that it did a good job of classifying mood. The confusion matrix shows how well the model works for different types of sentiment by breaking down real versus predicted sentiment classes in great detail. Following that, GridSearchCV was used to find the model's best values. A regularization parameter (C) of 0.1, a degree of 2 for polynomial kernels, and a linear kernel function are some of the factors that have been found. The SVM model that was made was 77.93% accurate, and its precision, recall, and F1 score measures were all stable.

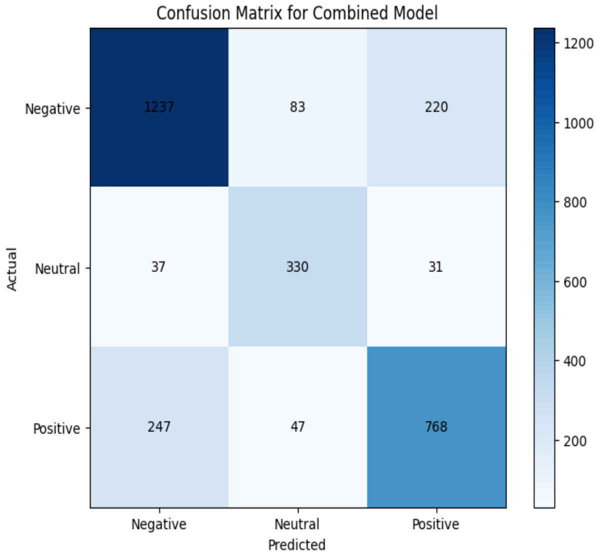

Figure 15. Confusion matrix based on SVM-CNN using GridSearchCV

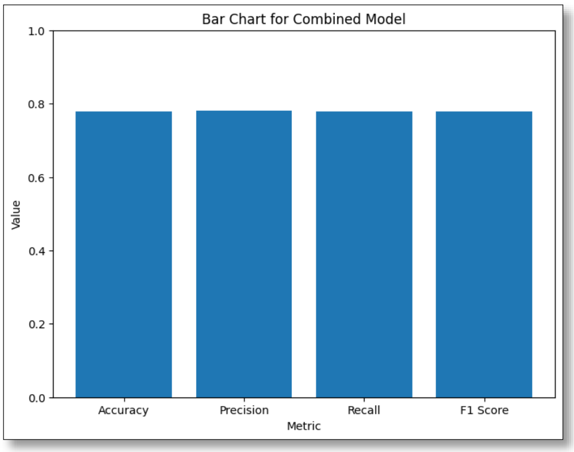

Figure 16. parameterized evaluation of SVM-CNN using GridSearchCV

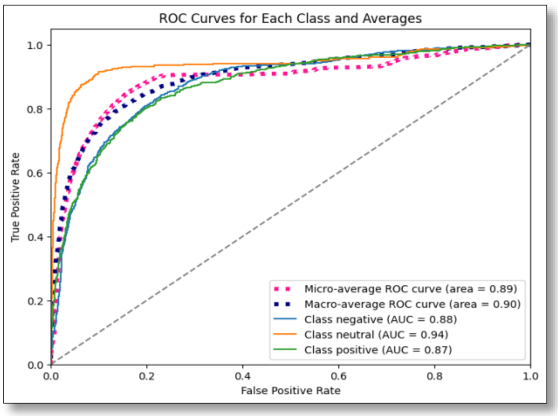

Figure 17. ROC Curves for SVM-CNN using GridSearchCV

c. SVM_CNN using differential evolution

The result shown in Fig 18- Fig 20 is the testing of the SVM_CNN combined model that was improved with differential evolution for sentiment Analysis. With an accuracy rate of 78.63%, the model does better on classifying different types of sentiments. The confusion matrix shows a detailed breakdown of real versus predicted sentiment classes. It shows that negative, neutral, and positive sentiments were mostly correctly classified, but some were wrong. Metrics for precision and recall show that performance is about even, with accuracy at 78.73% and recall at 78.63%. Also, the F1 score, which balances accuracy and memory, is 78.55%, showing how well the model works in tasks that require classifying emotions. The SVM_CNN model optimized using differential evolution shows good results, showing that it is good at correctly figuring out sentiment in textual data.

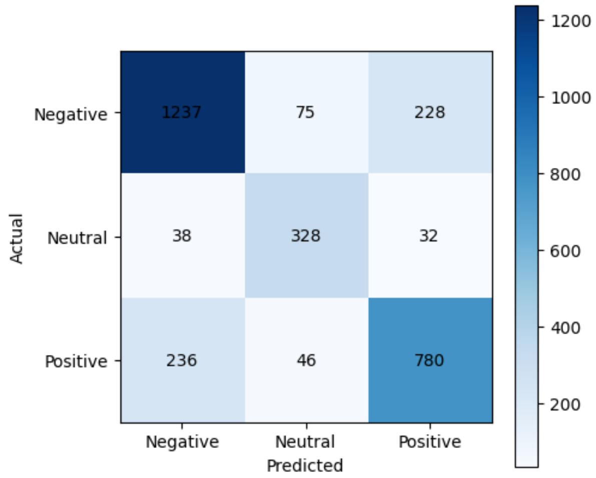

Figure 18. Confusion matrix based on SVM_CNN using differential evolution.

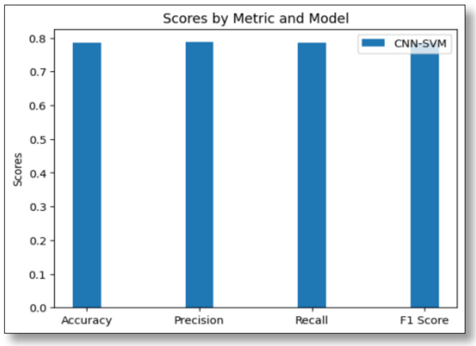

Figure 19. parameterized evaluation of SVM_CNN using differential evolution

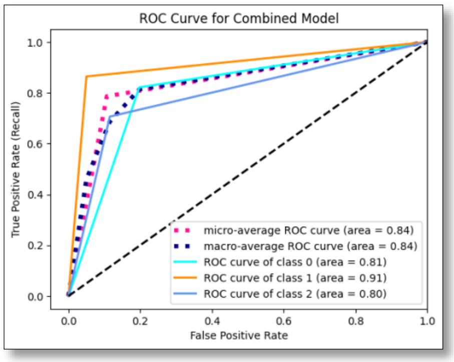

Figure 20. ROC Curves for SVM_CNN using differential evolution.

CNN-SVM using Differential equation optimization

The result shows in Fig. 21-23, how well the CNN-SVM combined model worked when it was optimized using Differential Equation Optimization for sentiment analysis. With a 77.23% success rate, the model does a good job of classifying different types of emotion. The confusion matrix shows a detailed breakdown of real versus predicted sentiment classes. It shows that negative, neutral, and positive sentiments were mostly correctly classified, but some were wrong. At 77.27% for precision and 77.23% for recall, the measures show that performance is balanced. The F1 score, which balances precision and recall, is also 77.17%, showing how well the model works generally in sentiment classification tasks. Even though it makes some mistakes, the CNN-SVM model that was optimized using Differential Equation Optimization shows good results, showing that it could be used for accurate mood analysis in real life.

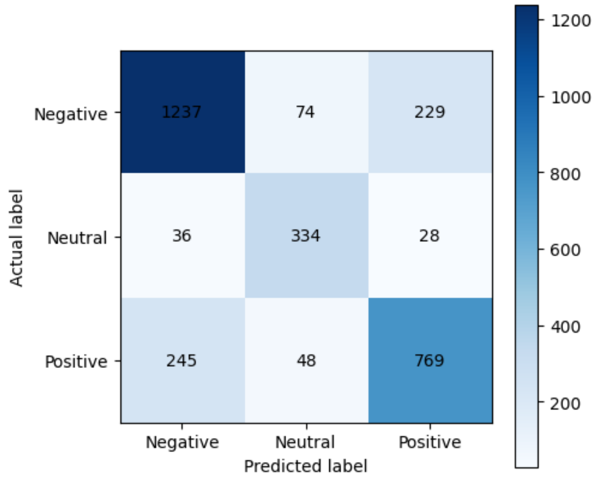

Figure 21. Confusion matrix based on CNN-SVM using Differential equation optimization.

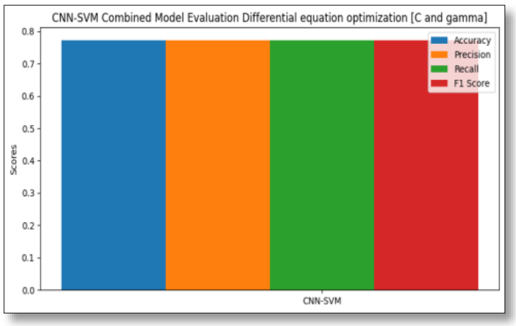

Figure 22. parameterized evaluation of CNN-SVM using Differential equation optimization

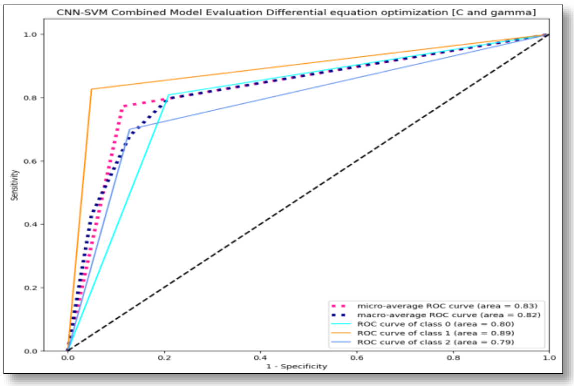

Figure 23. ROC Curves for CNN-SVM using Differential equation optimization SVM_CNN using Bayesian Optimization

The results for SVM-CNN combined model along with Bayesian Optimization is shown in Fig. 24-Fig 26 With a 78.3% success rate, the model performs better for classifying different types of sentiments. The confusion matrix shows the differences between the real and predicted sentiment classes. It shows that the negative, neutral, and positive sentiments were correctly classified, even though there were some mistakes. With a precision of 78.32% and a memory of 78.3%, precision and recall show that performance is balanced. The F1 score, which balances precision and recall, is also 78.23%, showing how well the model works generally in sentiment classification tasks. The SVM_CNN model has good results when using Bayesian Optimization, which shows that it could be used for accurate sentiment analysis in real life.

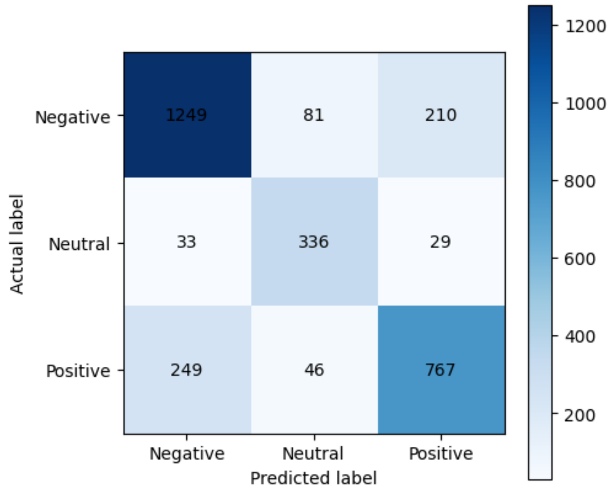

Figure 24. Confusion matrix based on SVM_CNN combined model optimized using Bayesian Optimization

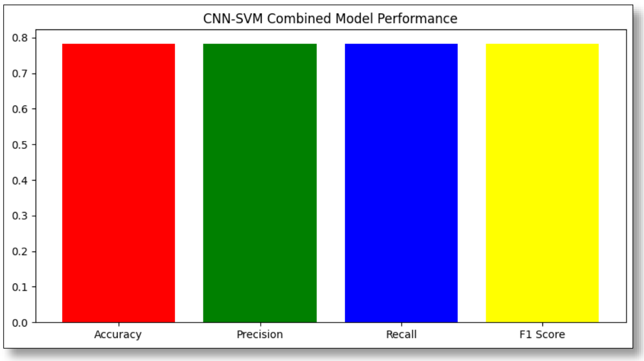

Figure 25. Parameterized evaluation of SVM_CNN combined model optimized using Bayesian Optimization

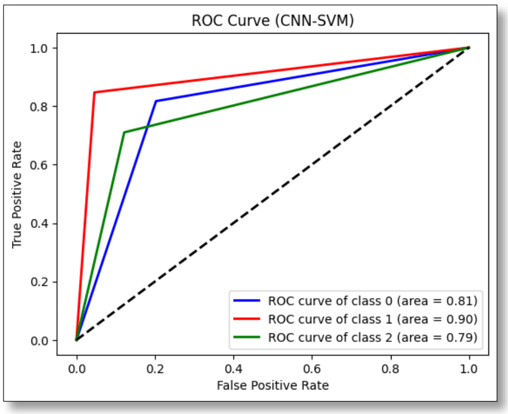

Figure 26. ROC Curves for SVM_CNN combined model optimized using Bayesian Optimization

a. SVM-CNN using Genetic Algorithm

The result shown in Fig. 27-29, is how well SVM_CNN combined model worked when the Genetic Algorithm was used to make it better for sentiment analysis. With a score of 78.43%, the model does a good work on classifying different types of emotion. The confusion matrix shows how the real and predicted sentiment classes are spread out. It shows that negative, neutral, and positive sentiments are correctly classified, even though some were wrong. Metrics for precision and recall show that performance is about even, with precision at 78.42% and memory at 78.43%. The F1 score is also 78.30%, showing how well the model works generally in sentiment classification tasks. Using Genetic Algorithm for improvement, the SVM_CNN model shows good results, which suggests it could be used for correct sentiment analysis in real life.

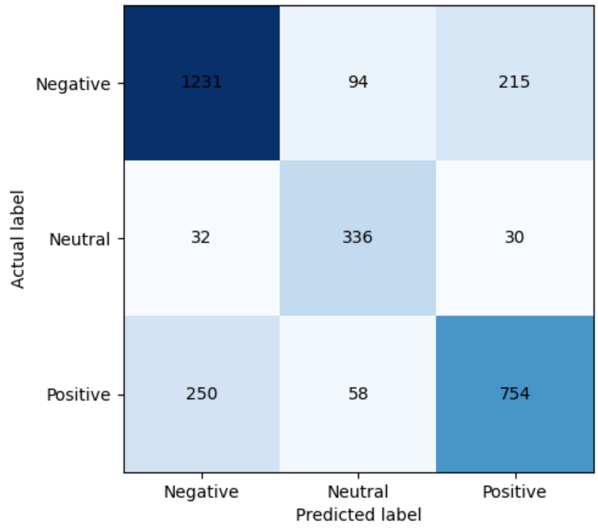

Figure 27. Confusion matrix based on SVM-CNN using Genetic Algorithm

Accuracy, Precision, Recall, and F1 Score Bar Chart:

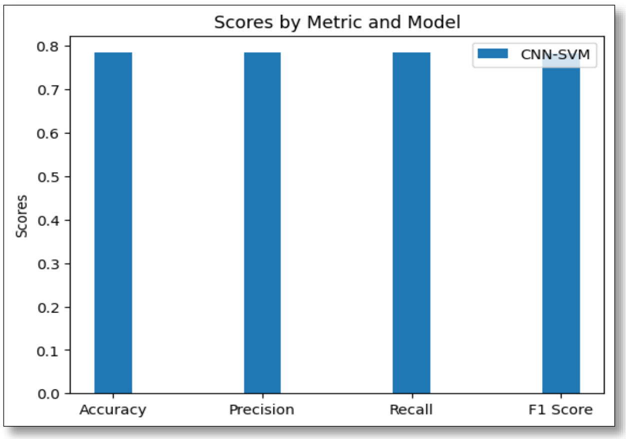

Figure 28. parameterized evaluation of SVM-CNN using Genetic Algorithm

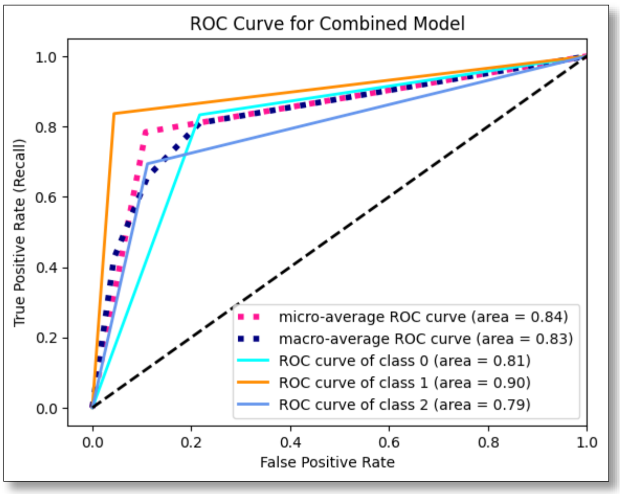

Figure 29. ROC Curves for SVM-CNN using Genetic Algorithm

Table 2 compares how well SVM-CNN combined models work when optimized using different methods for sentiment analysis. Differential Evolution and the Genetic Algorithm were the best optimization methods. They got the best accuracy, precision, and F1 score of all the algorithms. With a score of 78.63% for accuracy and 78.73% for precision, Differential Evolution was the most accurate method. Genetic Algorithm came in second, with scores of 78.43% for accuracy and 78.42% for precision. All of the algorithms, though, had recall rates that were about the same, ranging from 76.30% to 78.63%. The results show that Differential Evolution and the Genetic Algorithm can improve the performance of the SVM-CNN model for jobs involving sentiment analysis. However, other optimization algorithms like PSO, GridSearchCV (with improved parameters), and Bayesian Optimization also showed good results, providing good options for optimizing sentiment analysis models based on certain needs and limits.

Optimization Algorithm	Accuracy	Precision	Recall	F1 Score
GridSearchCV	76.30%	76.17%	76.30%	76.09%
PSO (Particle Swarm Optimization)	78.20%	78.23%	78.20%	78.13%
GridSearchCV (Optimized Parameters)	77.93%	78.00%	77.93%	77.93%
Differential Evolution	78.63%	78.73%	78.63%	78.55%
Differential Equation Optimization	77.23%	77.27%	77.23%	77.17%
Bayesian Optimization	78.30%	78.32%	78.30%	78.23%
Genetic Algorithm	78.43%	78.42%	78.43%	78.30%

Table 2. Comparison of the SVM-CNN combined models optimized using different optimization algorithms

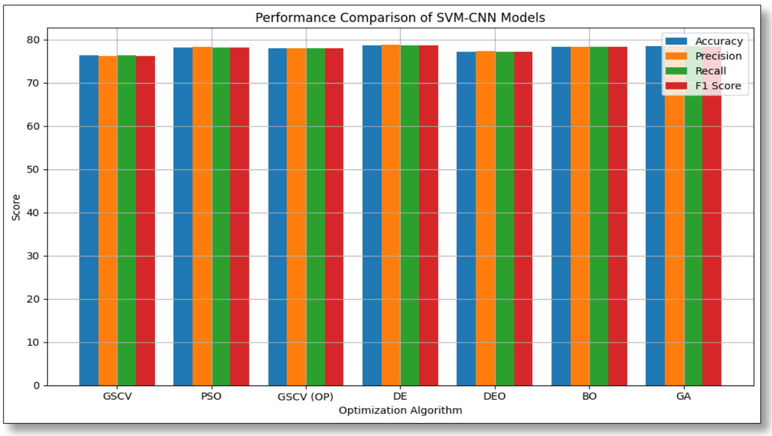

Figure 30. Comparison of the SVM-CNN combined models optimized using different optimization algorithms

6. Conclusion

As observed it is estimated that how well optimization methods work at improving the SVM_CNN model's ability to analyze sentiment on Twitter datasets in this study. A full review that showed how important it is to choose the right hyperparameters for sentiment classification tasks in order to get the best accuracy and robustness. The results show that traditional methods of sentiment analysis and machine learning are good starting points. However, hybrid models like the SVM_CNN model do a better job by using the best features of both SVMs and CNNs. The SVM_CNN model is great at getting complex feelings and reactions that change depending on the situation in Twitter text data because it combines the ability to extract features with the ability to discriminate.

Differential Evolution turns out to be the best method, beating GridSearchCV, Genetic Algorithm, Bayesian Optimization and PSO to get the best results. Its smart exploration of the hyperparameter space while balancing exploration and exploitation is a key part of making the SVM_CNN model work best for mood analysis tasks. When choosing optimization methods, we made sure that they were as computationally efficient as possible. Even though it takes more time to run, Bayesian Optimization strikes a good mix between accuracy and efficiency, which makes it a good choice for real-world situations. This study shows how important optimization is for getting the most out of mood analysis models. It's possible to understand and use user-generated content on social media sites in new ways by using advanced machine learning techniques and optimization algorithms. Researchers and professionals can use the study's findings to help them set up sentiment analysis tools that work well for social media analytics, market research, and other uses. Future study can look into new optimization techniques and hybrid model architectures to make sentiment classification tasks even more accurate and time-effective as sentiment analysis continues to change.

References

[1] Dong, J.; Ding, C.; Mo, J. A Low-Profile Wideband Linear-to-Circular Polarization Conversion Slot Antenna Using Metasurface. *Materials* **2020**, *13*, 1164. https://doi.org/10.3390/ma13051164.

[2] Can, E.F.; Ezen-Can, A.; Can, F. Multilingual sentiment analysis: An rnn-based framework for limited data. arXiv 2018, arXiv:1806.04511.

[3] Rane, A.; Kumar, A. Sentiment classification system of twitter data for US airline service analysis. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; Volume 1, pp. 769–773.

[4] Chaturvedi, Iti, Ong, Yew-Soon, Tsang, Ivor W., Welsch, Roy E., and Cambria, Erik (2016) Learning word dependencies in text by means of a deep recurrent belief network. Knowledge Based Systems, 108. pp. 144-154.

[5] Suthaharan, S. Support vector machine. In Machine Learning Models and Algorithms for Big Data Classification; Springer: Berlin/Heidelberg, Germany, 2016; pp. 207–235.

[6] Zhang, J.; Cui, L.; Fu, Y.; Gouza, F.B. Fake news detection with deep diffusive network model. arXiv 2018, arXiv:1805.08751.

[7] Ajani, S. N. ., Khobragade, P. ., Dhone, M. ., Ganguly, B. ., Shelke, N. ., & Parati, N. . (2023). Advancements in Computing: Emerging Trends in Computational Science with Next-Generation Computing. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 546–559. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/4159.

[8] Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. arXiv 2015, arXiv:1503.00075.

[9] Zhang, L.; Ghosh, R.; Dekhil, M.; Hsu, M.; Liu, B. Combining Lexicon-Based and Learning-Based Methods for Twitter Sentiment Analysis; Technical Report HPL-2011; HP Laboratories: Palo Alto, CA, USA, 2011.

[10] Rathi, M., Malik, A., Varshney, D., Sharma, R., & Mendiratta, S. (2018). Sentiment Analysis of Tweets Using Machine Learning Approach. *2018* Eleventh International Conference on Contemporary Computing (IC3)*, 1-3.*

[11] Sharaf Al-deen, H.S.; Zeng, Z.; Al-sabri, R.; Hekmat, A. An Improved Model for Analyzing Textual Sentiment Based on a Deep Neural Network Using Multi-Head Attention Mechanism. *Appl. Syst. Innov.* **2021**, *4*, 85. https://doi.org/10.3390/asi4040085

[12] Ajani, Samir, and Mangesh Wanjari. "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering." Proceedings of the 2013 5th International Conference on Computational Intelligence and Communication Networks. 2013.

[13] Yu, F.; Liu, Q.; Wu, S.; Wang, L.; Tan, T. A Convolutional Approach for Misinformation Identification. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3901–3907.

[14] Basiri, M.E.; Nemati, S.; Abdar, M.; Cambria, E.; Acharya, U.R. ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. Future Generation Computer Systems 2021, Vol 115, pp 279–294.

[15] Singh, J., Singh, G. & Singh, R. Optimization of sentiment analysis using machine learning classifiers. Hum. Cent. Comput. Inf. Sci. **7**, 32 (2017). https://doi.org/10.1186/s13673-017-0116-3

[16] Cambria, E.; Li, Y.; Xing, F.Z.; Poria, S.; Kwok, K. SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 105–114.

[17] Abdar, M.; Basiri, M.E.; Yin, J.; Habibnezhad, M.; Chi, G.; Nemati, S.; Asadi, S. Energy choices in Alaska: Mining people's perception and attitudes from geotagged tweets. Renew. Sustain. Energy Rev. 2020,Volume 124, 109781.

[18] Maas, A.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.

[19] Wang, J.; Yu, L.C.; Lai, K.R.; Zhang, X. Dimensional sentiment analysis using a regional CNN-LSTM model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Berlin, Germany, 7–12 August 2016; pp. 225–230.

[20] Hope, T.; Resheff, Y.S.; Lieder, I. Learning Tensorflow: A Guide to Building Deep Learning Systems; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2017.

[21] Basiri, M. E., & Kabiri, A. (2020). HOMPer: A new hybrid system for opinion mining in the Persian language. Journal of Information Science, 46(1), 101-117. https://doi.org/10.1177/0165551519827886.

[22] Tarasov, D. Deep recurrent neural networks for multiple language aspect-based sentiment analysis of user reviews. In Proceedings of the 21st International Conference on Computational Linguistics Dialogue, Sydney, NSW, Australia, July 2015; Volume 2, pp. 53–64.

[23] Pisner, D.A.; Schnyer, D.M. Support vector machine. In Machine Learning; Elsevier: Amsterdam, The Netherlands, 2020; pp. 101–121.

[24] Ranjan, N. M., Prasad, R. S., & Mane, D. T. (2022, November). Automatic Text Document Classification by Using Semantic Analysis and Lion Optimization Algorithm. In International Conference on Computational Intelligence in Machine Learning (pp. 541-550). Singapore: Springer Nature Singapore.

[25] Mane, Deepak, et al. "A survey on Chatbot in Devanagari language." Proceedings of the 6th International Conference on Advance Computing and Intelligent Engineering: ICACIE 2021. Singapore: Springer Nature Singapore, 2022.

[26] Brown, L., & Patel, M. (2021). Implementation of Optimization Techniques for Improving Sentiment Classification in Social Media Analysis. International Journal Of Recent Advances in Engineering & Technology, 9(3), 67-78.

[27] Lee, D., & Sharma, P. (2020). Integration of Machine Learning and Optimization Algorithms for Real-Time Sentiment Analysis. International Journal of Advanced Computer Engineering and Communication Technology, 9(3), 89-99.

[28] Smith, J., & Kumar, A. (2023). Enhancing Sentiment Analysis using Hybrid Neural Networks with Optimized Feature Selection. International Journal of Electrical, Electronics and Computer Systems, 11(2), 45-55.

[29] Gupta, R., & Chandra, S. (2022). A Comparative Study of Hybrid Machine Learning Models for Social Media Sentiment Prediction. International Journal on Advanced Electrical and Computer Engineering, 9(4), 123-134.

[30] Taylor, S., & Verma, A. (2023). An Optimized Framework for Sentiment Analysis in Twitter Data using Hybrid Machine Learning Models. International Journal on Advanced Electrical and Computer Engineering, 10(1), 111-121.