Dynamic weight adaptation in soft voting for emotion detection using neural networks

Nisha, Rakesh Kumar
Department of Computer Science and Applications (DCSA), Kurukshetra University, Kurukshetra, India

Article Info

Article history:

Received May 5, 2025
Revised Nov 7, 2025
Accepted Dec 6, 2025

Keywords:

Ensemble learning
Machine learning
Multi-label emotion
Real time predictions
Soft voting

ABSTRACT

Confirming elevated accuracy and speed in multi-label automatic emotion classification endures to pose extensive challenges. Old-style machine learning (ML) models are broadly used for this. However, large-scale fast embryonic textual information often obstructs their performance. Deep learning (DL) models resolve the former problem efficiently, but fine-tuning the hyperparameter entails a lot of work and experience. Ensemble learning practices offer enhanced accuracy, but classical soft voting classifiers with static weights fall short to adapt effectively to diverse data traits. To tackle this limitation, this study proposes a novel ensemble framework that employs a neural network (NN) based dynamic weight adaptation within a soft voting classifier. The model dynamically adjusts the weights of core ML classifiers based on their real-time predictive likelihood and performance statistics. This adaptive weighting suggestively enhances the model's ability in detecting nuanced emotional expressions in text, improving responsiveness and generalization. Comprehensive experiments conducted on yardstick emotion dataset demonstrate that proposed integration of NN driven adaptive weighting within an ensemble framework outpaces traditional approaches, capturing an overall classification accuracy of approximately 98% thus offering a scalable and robust solution for real-world sentiment analysis applications.

Corresponding Author:

Nisha
Department of Computer Science and Applications (DCSA), Kurukshetra University
Kurukshetra, Haryana, India
Email: nisha.jrf.dcsa@kuk.ac.in

1. INTRODUCTION

Textual data available on digital media, blogs, forums, and customer assessments has risen exponentially. These modes are common for expressing emotions, intentions, and thoughts [1]. Mining emotional signals from text is crucial topic of affective computing and natural language processing (NLP). Text based emotion detection (TBED) is widely used in consumer behavior investigation, mental health monitoring, personalized suggestions, and social media moderation [2]. Originally sentiment analysis classifies positive, negative, or neutral emotions. However, crude classification typically fails to convey the complexity of human emotions [3]. Emotions are conscious psychological reactions (like love, hate, or terror) that are subjectively experienced as powerful feelings focused on a specific entity and typically accompanied by physiological and behavioral changes in the human organism (American Psychological Association – APA 2023) [4]. This multidimensional emotion classification provides psychological insights and benefits real-world decision-making systems [5]. TBED has progressed so much all because of machine learning (ML) algorithms. Due to their robustness and capacity to learn from labelled data models like support vector machine (SVM), Naïve Bayes (NB), and random forest (RF) [6]. These models beat traditional technique of mining such as rule and lexicon approach, which struggle to keep up with language's rapid growth and dependency on context. However, typical ML methods sometimes struggle with real time voluminous data such as Twitter, where over 500 million tweets are generated daily [7]. Human-computer interactions (HCI) require emotion recognition as computers don't feel emotions, but researches show that user engagement is just as central as technological [8]. Emotion-aware computing improves personalization and user experience by tailoring responses to emotional inputs [9]. By analyzing online input, organizations may enhance system effectiveness, HCI tactic, and decision-making. Data collection and analysis technologies pervasiveness in people's daily lives produces gigantic amounts of data about users and their behaviors, resulting in the Internet of Behaviors and exponential data growth [10]. TBED can help in developing adaptive systems, response architecture like user comments on Mobile eCommerce apps [11], evaluate interactive systems, and comprehend people's thinking to check mental wellness can reveal user true challenges [12]. Brevity and casual language make short-text sorting difficult, worsening inflexibility. Recent ML techniques like ensemble learning have been useful in malware detection, music identification, image and text classification [13]-[17]. The effectiveness of the ensemble system is due to “the wisdom of crowds” i.e., diversifying baseline classifier.

Recently, ensemble of deep learning (DL) models also outperformed in TBED than others because of reducing variance and leverages complementary strengths of DL models [18]-[21]. Furthermore, ensemble learning methods mitigate baseline model drawbacks. There are several ensemble approaches in the literature mainly voting, boosting, bagging, and stacking [22]. Table 1 lists pros and cons of each.

Table 1. Ensemble technique summary

Ensemble technique	Pros	Cons
Voting	- Simple. - Reduce risk of poor performance of base classifier.	- All models treated equally (until weight is assigned).
Bagging	- Reduce variance (prevents overfitting). - Excellent handles high-dimensional data. - Allow weak learners outperform strong ones. - Strong against noise or outliers. - Easy implementation and adaptability	- Bias is high. - Compute intensive. - Loss of model interpretability.
Boosting	- Lowers variance. - Ease of model interpretability.	- Slower training. - Compute intensive. - Every classifier must fix the mistakes of its predecessors. - Scalability issues associated with sequential training.
Stacking	- Deeper data comprehension. - More reliable. - Used to group diverse strong learners.	- Final model interpretation difficulty. - Computational complexity (in time). - Higher tendency to overfit.

Hard and soft voting are common subtype of voting ensemble approach for prediction. Hard voting or majority voting, uses discrete class labels predicted by separate models to select the most popular class [23]. Whereas soft voting employs each model's predicted probability for more nuanced and accurate results, especially when confidence levels vary. These voting done on static weight ignoring model performance across contexts. Based on this foundation, the proposed model in this study uses variation of soft voting, where a neural network (NN) dynamically adjusts the contributions of each base classifier instead of equal weighting or majority voting.

TBED remains a challenging task due to overlapping emotional categories, imbalanced datasets, and the limited generalization ability of individual classifiers. This study addresses the challenge of improving multi-class emotion detection from textual data, a task where traditional classifiers often struggle with overlapping semantic cues. While prior research has employed ensemble learning or DL for sentiment and emotion recognition, but most works are limited either by small-scale datasets, fixed-weight voting strategies, or lack of adaptability across emotion categories [24]-[27]. These limitations highlight the need for more flexible ensemble frameworks that can dynamically exploit the strengths of different base classifiers. To address this gap, our contribution lies in proposing a NN based ensemble with dynamic weight adaptation, enabling baseline models such as SVM, NB, RF, and eXtreme gradient boosting (XGBoost) to emphasize their relative strengths making ensemble prediction more resilient and flexible. This integration improves classification and system reaction to complex, nuanced emotional content in brief texts. The suggested model improves TBED, HCI, and emotion-aware AI systems using benchmark dataset. The rest of the article is organised as follows: in section 2, we discuss methods utilised and proposed model to address the problem.

Section 3 covers experiments, results, and comparisons of several models for the given topic. Section 4 presents the conclusion and future scope.

2. METHOD

The meta-learning architecture in this study adjusts base classifiers contributions to input data attributes, unlike static voting classifiers with fixed weights. Figure 1 shows the NN based soft voting workflow from data preprocessing and base classifier predictions to adaptive weight assignment. Here, firstly data is rigorously preprocessed to improve and standardize textual inputs. SVM, RF, NB, and XGBoost served as base classifiers which identify emotions independently. Further NN that assigns dynamic weights to base classifiers acts as a meta-learner instead of directly predicting the emotion, it learns how much trust (weight) to put on each base classifier depending on their strengths for different classes. This approach optimizes model strengths and refines decision-making through learnt weight assignments.

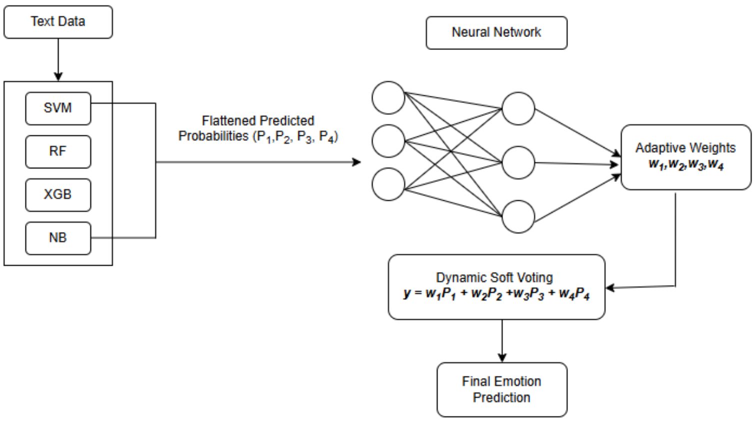

Figure 1. Ensemble framework with adaptive dynamic weighting

2.1. Feature engineering

Text classification uses feature engineering to keep only the most significant traits from text. It includes feature extraction which create numerical feature vectors from preprocessed text, term frequency – inverse document frequency (TF-IDF) vectorization used in this study. Identifying the most effective textual features capable of accurately estimating emotions as by selecting the relevant features cuts overfitting, computational complexity, and thus improve model performance [28]. Our study used the filter-based statistical approach chi-square (χ^2) for feature selection. The χ^2 approach examines the statistical association between features and labels to assess their impact on classification. The top 500 traits are chosen using the χ^2 test to prioritize the most important qualities for predictions.

2.2. Classification models

Ensemble learning highly depends upon diversity of combined base model and should complementarity in nature to accomplish robust categorization. In this study heterogeneous base classifier those with good accuracy like SVM for high-dimensional text characteristics, RF due to its robustness to overfitting, NB for probabilistic interpretability, and XGBoost for non-linear decision boundaries were chosen. The suggested solution is uncommon since these classifiers assign adaptive weights. Traditional ensemble learning assigns fixed weights to classifiers, assuming their contributions remain consistent across all instances. However, real-world online textual data is often highly variable: new slang, evolving sentiments, and domain shifts can significantly impact classifier reliability. Thus, a dynamic weight adaptation mechanism is introduced to improve performance in such unpredictable environments.

2.3. Meta-learner architecture

The adaptive weight allocation for the ensemble classifiers is attained over a multi-layer perceptron trained on predicted probabilities of base classifiers thus help in determine optimal weight distributions for them. These learned weights are subsequently incorporated into a soft voting NN based ensemble classifier. The NN weighting mechanism is suitable for learning complex inter-classifier relationships because it approximates non-linear functions. This adaptive approach ensures that classifiers exhibiting higher confidence in specific instances contribute more significantly to the final classification decision, thereby optimizing overall predictive performance. Algorithm 1 states the meta learner architecture working flow.

Algorithm 1. Dynamic weight assignment for soft voting using neural network
Input: $P_1, P_2, \dots. . P_n$ Probability distributions from n base classifiers (size: $n * m$ where m is the number of emotion classes). Additional contextual features (optional) like Confidence scores or entropy of predictions.
Output: W: Dynamic weights for each base classifier.
P_{final}: Final predicted emotion class label.
Steps for Dynamic Weight Assignment for Soft Voting NN
Step 1: Prepare Input for NN
1. **Combine Probability Distributions:** Flatten the probability distribution of each base classifier. Input to NN: $\left[P_{1_{c1}}, P_{1_{c2}}, \dots, P_{n_{cm}}\right]$ where $P_{i_{c_j}}$ denotes the probability predicted by the *i-th* base classifier for the *j-th* class. If n classifiers and m emotion classes, the input vector will have a size of $n * m$.

Step 2: Neural Network Architecture
1. **Input Layer:** Accept the flattened probability distribution vector (confidence scores) from the base classifiers.
2. **Hidden Layers:** Fully connected layers with ReLU to learn the non-linear relationships between the outputs of the base classifiers.
3. **Output Layer:** Outputs dynamic weights for each base classifier. Activation function: Softmax to normalize the weights so their sum is 1.
   - Output: $W = [w_1, w_2, \dots, w_n]$,where w_i represents the dynamic weight for the i^{th} classifier.

Step 3: Train the Neural Network
1. **Objective:** Minimize error (e.g., cross-entropy loss) to assign optimal weights that maximize classification accuracy.
2. **Training Dataset:** Train the network to learn the optimal weights that combine the classifier predictions effectively.
3. **Loss Function:** Loss function such as cross-entropy to compare the combined prediction (weighted by W) with the true label.

Step 4: Dynamic Weight Adaptation During Inference
1. **Generate Dynamic Weights:** For each test sample, pass the predicted probabilities from each base classifier through the trained NN to generate dynamic weights W_{test}.
2. **Final Prediction:** Apply the weights W_{test} to the probability distributions: $P_{\text{final}} = \sum_{i=1}^{n} w_i \times P_i$ where P_i is the predicted probability from the *i-th* classifier, and w_i is its corresponding weight.
   - **Prediction:** Assign the final predicted class label by taking the argmax over the weighted probabilities: $\hat{y} = \arg\max(P_{\text{final}})$.

3. RESULTS AND DISCUSSION

To ensure that our suggested model is as accurate as possible, we have carried out different kinds of tests on the multitext dataset that consists of phrases, tweets, and dialogues. Base models like SVM, NB, RF, and XGBoost were implemented individually and in combination using hard and soft voting, followed by a NN-based weighted ensemble for multi-class emotion classification.

3.1. Experimental results

All experiments were conducted on a desktop processor running with Windows 10 operating system. The models discussed and compared in the above sections were implemented in Python. The "Emotions" dataset which contains English-language tweets meticulously labelled with 6 emotion classes: sadness (0), joy (1), love (2), wrath (3), fear (4), and surprise (5) used for this study, was obtained from Kaggle [29]. Approximately 417K Twitter comments were originally collected. We extracted balanced subsets for each emotion for our experiment as data is intrinsic in original nature. Stratified sampling is used to create three balanced datasets of 30K, 60K, and 90K samples. This method distributes emotions evenly, preventing class imbalance, and providing enough cases to train classifiers.

3.2. Performance evaluation

For quantifying the performance of the proposed framework, it is necessary to have model's assessment metrics. A complete description of outcomes of the base models, shown by confusion matrix (CM) in Figures 2(a)-(f) for the six different classes of emotions (0-5). The operation flow employs pipelining of data preparation, feature extraction, and subsequent classification via dynamically weighted ensemble learning. The efficacy of classifier's is encapsulated by emphasizing the share of valid predictions along the diagonal and the dispersion of false predictions across six emotion categories. CM presents an evident demonstration of how well the model differentiates across multiple emotions and where sporadic confusions occur. Such an evaluation is crucial in multi-class emotion detection tasks, as it not only reflects the robustness of the approach but also offers insights into model behavior for improving classification strategies. 80-20 cross validation is used out for model's evaluation. From Figure 2(g), it quite evident that the proposed dynamic NN based method uniformly surpassed static baselines, including equal-weight soft voting and majority voting, with improvements reflected in both accuracy and class-wise error reduction. This validates that this dynamic weight adaptation facilitates direct to performance boost over fixed heuristic techniques.

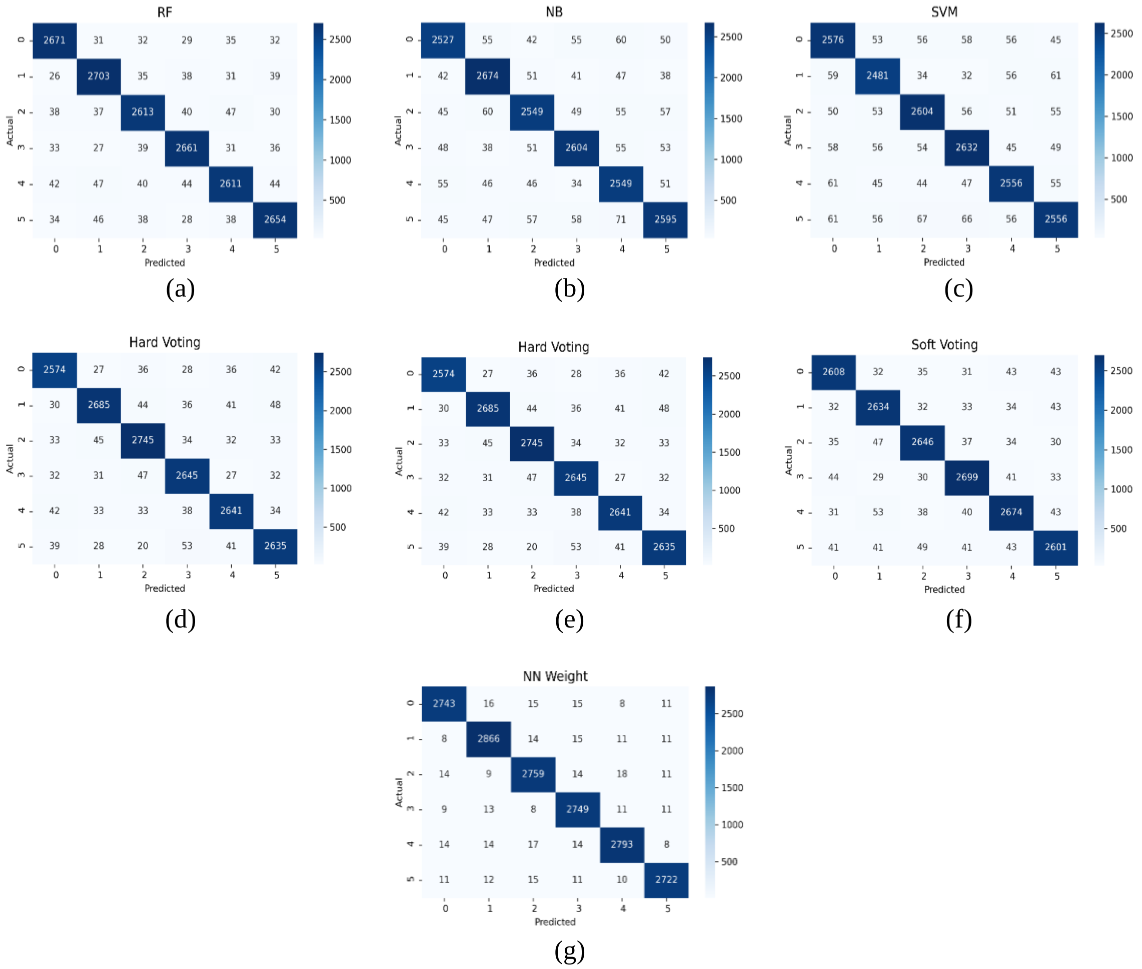

Figure 2. CM of six emotion classes using different classifiers; (a) RF, (b) NB, (c) SVM, (d) XGBoost, (e) hard voting, (f) soft voting, and (g) proposed NN weight

Evaluation is also performed on the outcomes of each model, but per-class accuracy is not reported here as it is typically inflated by the large number of TN in multi-class settings. Instead, we follow the scikit-learn convention of reporting precision, recall, and F1-score per class, which provide a more reliable measure of class-level performance. Since our dataset is nearly balanced, these metrics fairly represent the model's effectiveness as shown in Tables 2-8. which indicates that NN based dynamic voting classifier performed equally well in classifying the all emotions separately.

A concise summary of the quantitative results obtained by the different classifiers that were mentioned before is depicted in Figure 3, which clearly shows that the proposed model with NN weight attain highest overall classification accuracy of 97.84% and guarantees that the base classifiers who contribute more to correct predictions get increased influence in participation.

Table 2. Classification result of different classes using RF

Class	Precision	Recall	F1-score
Sadness (0)	93.91	94.28	93.95
Joy (1)	93.49	94.11	93.81
Love (2)	93.42	93.11	93.29
Anger (3)	93.69	94.12	93.91
Fear (4)	93.47	92.32	92.90
Surprise (5)	93.62	93.51	93.37

Table 3. Classification result of different classes using NB

Class	Precision	Recall	F1-score
Sadness (0)	93.91	94.28	93.95
Joy (1)	93.49	94.11	93.81
Love (2)	93.42	93.11	93.29
Anger (3)	93.69	94.12	93.91
Fear (4)	93.47	92.32	92.90
Surprise (5)	93.62	93.51	93.37

Table 4. Classification result of different classes using SVM

Class	Precision	Recall	F1-score
Sadness (0)	89.91	90.57	90.24
Joy (1)	90.41	91.11	90.76
Love (2)	91.08	90.76	90.92
Anger (3)	91.04	90.94	90.99
Fear (4)	90.64	91.02	90.83
Surprise (5)	90.61	89.21	89.95

Table 5. Classification result of different classes using XGBoost

Class	Precision	Recall	F1-score
Sadness (0)	92.96	91.77	92.36
Joy (1)	92.93	92.40	92.66
Love (2)	92.74	93.13	92.94
Anger (3)	92.30	92.73	92.51
Fear (4)	92.95	92.82	92.89
Surprise (5)	91.50	92.54	92.02

Table 6. Classification result of different classes using hard voting

Class	Precision	Recall	F1-score
Sadness (0)	93.60	93.84	93.72
Joy (1)	94.24	93.10	93.67
Love (2)	93.85	93.94	93.89
Anger (3)	93.33	93.99	93.66
Fear (4)	93.72	93.62	93.67
Surprise (5)	93.31	93.57	93.44

Classification result of different classes using soft voting

Class	Precision	Recall	F1-score
Sadness (0)	93.44	93.41	93.43
Joy (1)	92.88	93.80	93.34
Love (2)	93.50	93.53	93.51
Anger (3)	93.68	93.85	93.76
Fear (4)	93.20	92.88	93.04
Surprise (5)	93.13	92.37	92.74

Table 8. Classification result of different classes using NN voting

Class	Precision	Recall	F1-score
Sadness (0)	98.00	97.69	97.84
Joy (1)	97.82	97.65	97.90
Love (2)	97.56	97.66	97.61
Anger (3)	97.55	98.14	97.85
Fear (4)	97.97	97.66	97.81
Surprise (5)	98.13	97.88	98.00

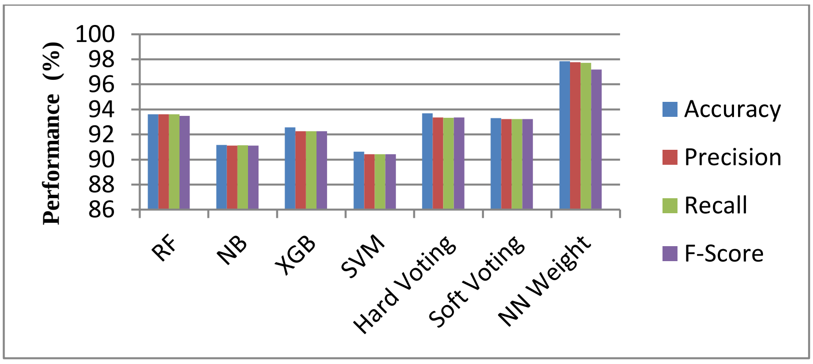

Figure 3. Performance evaluation of different techniques

3.3. Ablation study

We tested different optimizers, learning rate (LR), and batch size varied between 64, 128, and 256, and epoch to find stable hyperparameter settings concurrently while keeping the overall architecture fixed, which is summarized in Table 9. Early stopping (patience=5) and dropout (rate=0.3) were applied to mitigate overfitting, while validation monitoring ensured stable convergence.

Table 9. Architecture details of meta-learner

Layers	Configuration
Input	Predictive probabilities from base classifiers
Hidden	128 (ReLU), 64 (ReLU)
Output	Softmax, 6 classes
LR	0.001, 0.0001
Optimizer	ADAM and SGD
Batch size	64, 128, and 256
Epoch	10, 20, and 30

Further the ablation study that was carried out using the proposed adaptive strategy with soft voting technique is summarized in Table 10, which offers a succinct description of the study. However, these experiments may have blended learning paradigms without entirely isolating the influence of the NN-based weighting method. It is possible to demonstrate, from the Table 10, that the optimizer Adam produced the most optimum output as compared to SGD with different hyperparameters combination.

We separated the effect of χ^2 feature selection by running the NN-based ensemble with and without feature filtering. According to Table 11 of this study, the accuracy dropped from 97.84% (with χ^2) to 89.67% (without χ^2). This shows that feature selection with χ^2 cuts down on noise and boosts the model's ability.

Table 10. Ablution study of the proposed model

Optimizer	(LR)	Epochs	Batch size	Accuracy (%)	Loss
Adam	0.001	10	64	92.56	0.7438
Adam	0.001	10	128	93.98	0.7875
Adam	0.001	10	256	93.63	0.6653
Adam	0.0001	10	64	92.85	0.1670
Adam	0.0001	10	128	93.61	0.1593
Adam	0.0001	10	256	93.97	0.1431
Adam	0.001	20	64	95.86	0.2396
Adam	0.001	20	128	94.77	0.2854
Adam	0.001	20	256	96.09	0.2432
Adam	0.0001	20	64	94.33	0.0174
Adam	0.0001	20	128	97.49	0.0256
Adam	0.0001	20	256	97.05	0.0234
Adam	0.001	30	64	96.11	0.0146
Adam	0.001	30	128	97.84	0.0094
Adam	0.001	30	256	96.94	0.0178
Adam	0.0001	30	64	94.94	0.0335
Adam	0.0001	30	128	95.03	0.0143
Adam	0.0001	30	256	96.77	0.0151
SGD	0.001	10	64	89.61	0.7347
SGD	0.001	10	128	90.85	0.7452
SGD	0.001	10	256	89.97	0.6419
SGD	0.0001	10	64	84.60	0.4162
SGD	0.0001	10	128	86.89	0.3776
SGD	0.0001	10	256	87.06	0.3786
SGD	0.001	20	64	88.33	0.0334
SGD	0.001	20	128	92.68	0.0286
SGD	0.001	20	256	93.84	0.0290
SGD	0.0001	20	64	87.98	0.0330
SGD	0.0001	20	128	93.73	0.0284
SGD	0.0001	20	256	94.48	0.0206
SGD	0.001	30	64	91.83	0.0247
SGD	0.001	30	128	92.73	0.0191
SGD	0.001	30	256	93.97	0.0201
SGD	0.0001	30	64	94.88	0.0114
SGD	0.0001	30	128	95.64	0.0189
SGD	0.0001	30	256	96.84	0.0113

Table 11. Proposed model's performance over feature selection technique

Case	Feature selection	Accuracy (%)	Loss	Precision (%)	Recall (%)	F1-score (%)
1	---	89.67	0.697	82	79	74
2	Chi-Square (χ^2)	**97.84**	**0.0094**	**97**	**97**	**97**

Figure 4 presents the model's accuracy-loss graph, which confirms that the model boosts classification efficacy with rising accuracy and declining loss metrics across every epoch. This visualization aids in assessing the robustness and consistency of the models.

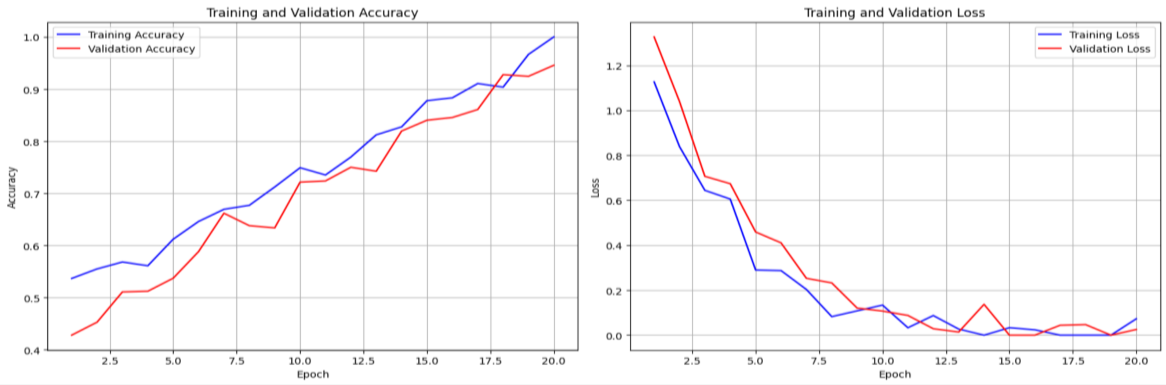

Figure 4. Training and validation performance of proposed NN weight respectively

The results clearly indicate that, in addition to accuracy, other factors such as dataset size and computational complexity should be considered when determining the best-performing model. For this we compared our proposed model effectiveness against existing approaches [18], [24]-[27] as shown in Table 12. It can be concluded that proposed model outperformed than that of prior studies employed on relatively smaller datasets and DL base models. In contrast, our framework was trained on a much larger dataset with a greater number of emotions classes consistently achieve higher accuracy and balanced performance, thereby demonstrating robustness and scalability compared to earlier static or task-specific methods which often fail to exploit the class-specific strengths of individual models. Earlier literature on adaptive ensembles, we identify that this approach introduces additional computational cost and effectiveness of output heavily depends on maintaining sufficient diversity among base classifiers, which may limit generalization if the ensemble composition is too homogeneous. Due to potential limitation of a baseline models, the ensemble's progress may be confined, and more testing on various real-world datasets is needed to confirm scalability.

Table 12. Accuracy comparison of our proposed study with similar literature studies

Source and publication year	Dataset type	Emotion classes/total count	Base model	Ensemble accuracy
Mohammed and Kora (2023) [18]	Arabic-Egyptian Corpus 2	10K	GRU, LSTM, and CNN	SVM=93.2% (soft voting)
Al-Laith and Shahbaz (2021) [24]	Dialects Tweets	5,5 K	GRU, LSTM, and CNN	LG=83.2% (soft voting)
Xu *et al.* (2020) [25]	Sina Weibo/English	2	CNN, SVM, LSTM, and RNN	CNN=97%
Rane and Kumar (2018) [26]	US Airlines	3,14 K	GRU and LSTM	GB=85.3% (soft voting)
Thiab *et al.* (2024) [27]	SemEval 2019	4,40 K	CNN and BERT RoBERTa and XLNeT	94.4% (soft) 93.3%(hard)
Proposed	Emotion (Kaggle)	6,90 K	SVM, NB, RF, and XGBoost	NN=98% (soft voting)

4. CONCLUSION

Through the incorporation of a novel adaptive NN based dynamic weight assignment into ensemble framework, this research presents a fresh approach to the multi-class emotion detection. When it comes to processing extremely changing online short textual data, the adaptability of the model is improved by the utilization of NN to dynamically calculate classifier weights. The findings of the empirical research verify the usefulness of this approach, revealing that it performs exceedingly well in comparison to the conventional ensemble classifiers. This methodology provides a solid framework for the classification of sentiments and emotions, and it has the potential to be utilized in real-time opinion mining as well as intelligent HCI systems. Building on this foundation, future research may focus on extending the model to cross-lingual and code-mixed datasets, integrating transformer-based embeddings such as BERT and RoBERTa, and finding more critical emotions like sarcasm and offensive language. Such generalization would enhance the model's applicability to diverse adapting the framework for real-time applications.

FUNDING INFORMATION

The author acknowledges financial support from the University Grants Commission (UGC), Government of India, through the UGC-NET Junior Research Fellowship (JRF) scheme.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Nisha	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Rakesh Kumar		✓		✓	✓	✓				✓	✓	✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state there is no conflict of interest.

DATA AVAILABILITY

The dataset that supports the findings of this study is openly available on Kaggle at https://www.kaggle.com/datasets/nelgiriyewithana/emotions, reference number [29].

REFERENCES

[1] P. Nandwani and R. Verma, “A review on sentiment analysis and emotion detection from text,” *Social Network Analysis and Mining*, vol. 11, no. 1, p. 81, Dec. 2021, doi: 10.1007/s13278-021-00776-6.
[2] K. Alahmadi, S. Alharbi, J. Chen, and X. Wang, “Generalizing sentiment analysis: a review of progress, challenges, and emerging directions,” *Social Network Analysis and Mining*, vol. 15, no. 1, pp. 1–28, Apr. 2025, doi: 10.1007/s13278-025-01461-8.
[3] H. Kaur, V. Mangat, and Nidhi, “A survey of sentiment analysis techniques,” in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India: IEEE, Feb. 2017, pp. 921–925, doi: 10.1109/I-SMAC.2017.8058315.
[4] L. Feldman-Barrett, “Wilhelm Wundt’s Conception of Affect,” how-emotions-are-made.com, [Online]. Available: https://how-emotions-are-made.com/notes/Wilhelm_Wundt%27s_conception_of_affect. (Accessed: Apr. 15, 2025).
[5] N. Alswaidan and M. E. B. Menai, “A survey of state-of-the-art approaches for emotion recognition in text,” *Knowledge and Information Systems*, vol. 62, no. 8, pp. 2937–2987, Aug. 2020, doi: 10.1007/s10115-020-01449-0.
[6] J. Deng and F. Ren, “A Survey of Textual Emotion Recognition and Its Challenges,” *IEEE Transactions on Affective Computing*, vol. 14, no. 1, pp. 49–67, Jan. 2023, doi: 10.1109/TAFFC.2021.3053275.
[7] A. Al Maruf, F. Khanam, M. M. Haque, Z. M. Jiyad, M. F. Mridha, and Z. Aung, “Challenges and Opportunities of Text-Based Emotion Detection: A Survey,” *IEEE Access*, vol. 12, pp. 18416–18450, 2024, doi: 10.1109/ACCESS.2024.3356357.
[8] T. A. B. Tretow-Fish and M. S. Khalid, “Methods for Evaluating Learning Analytics and Learning Analytics Dashboards in Adaptive Learning Platforms: A Systematic Review,” *Electronic Journal of e-Learning*, vol. 21, no. 5, pp. 430–449, Dec. 2023, doi: 10.34190/ejel.21.5.3088.
[9] T. Yu, I. Benbasat, and R. T. Cenfetelli, “The Influence of Electronic Word of Mouth on Beliefs and Purchase Intentions: The Mediating Role of Trust,” *Decision Support Systems*, vol. 50, no. 2, pp. 175–187, 2011.
[10] M. Javaid, A. Haleem, R. P. Singh, S. Rab, and R. Suman, “Internet of Behaviours (IoB) and its role in customer services,” *Sensors International*, vol. 2, pp. 1–8, 2021, doi: 10.1016/j.sintl.2021.100122.
[11] T. Olagunju, O. Oyebode, and R. Orji, “Exploring Key Issues Affecting African Mobile eCommerce Applications Using Sentiment and Thematic Analysis,” *IEEE Access*, vol. 8, pp. 114475–114486, 2020, doi: 10.1109/ACCESS.2020.3000093.
[12] O. Oyebode, F. Alqahtani, and R. Orji, “Using Machine Learning and Thematic Analysis Methods to Evaluate Mental Health Apps Based on User Reviews,” *IEEE Access*, vol. 8, pp. 111141–111158, 2020, doi: 10.1109/ACCESS.2020.3002176.
[13] V. Kumar, P. S. S. Aydav, and S. Minz, “Multi-view ensemble learning using multi-objective particle swarm optimization for high dimensional data classification,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 8523–8537, Nov. 2022, doi: 10.1016/j.jksuci.2021.08.029.
[14] R. K. Shahzad and N. Lavesson, “Comparative analysis of voting schemes for ensemble-based malware detection,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 4, no. 1, pp. 98–117, 2013, doi: 10.22667/JOWUA.2013.03.31.098.
[15] L. J. M. Raboy and A. Taparugssanagorn, “Verse1-Chorus-Verse2 Structure: A Stacked Ensemble Approach for Enhanced Music Emotion Recognition,” *Applied Sciences*, vol. 14, no. 13, pp. 1–26, Jul. 2024, doi: 10.3390/app14135761.
[16] T. Iqball and M. A. Wani, “Weighted ensemble model for image classification,” *International Journal of Information Technology (Singapore)*, vol. 15, no. 2, pp. 557–564, Feb. 2023, doi: 10.1007/s41870-022-01149-8.

[17] N. H. Kulkarni, G. N. Srinivasan, B. M. Sagar, and N. K. Cauvery, "Improving Crop Productivity Through A Crop Recommendation System Using Ensembling Technique," in *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)*, Bengaluru, India: IEEE, Dec. 2018, pp. 114–119, doi: 10.1109/CSITSS.2018.8768790.

[18] A. Mohammed and R. Kora, "An effective ensemble deep learning framework for text classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 8825–8837, Nov. 2022, doi: 10.1016/j.jksuci.2021.11.001.

[19] A. Mohammadi and A. Shaverizade, "Ensemble deep learning for aspect-based sentiment analysis," *International Journal of Nonlinear Analysis and Applications*, vol. 12, no. Special Issue, pp. 29–38, 2021, doi: 10.22075/IJNAA.2021.4769.

[20] M. Heikal, M. Torki, and N. El-Makky, "Sentiment Analysis of Arabic Tweets using Deep Learning," *Procedia Computer Science*, vol. 142, pp. 114–122, 2018, doi: 10.1016/j.procs.2018.10.466.

[21] A. Mansy, S. Rady, and T. Gharib, "An Ensemble Deep Learning Approach for Emotion Detection in Arabic Tweets," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, pp. 980–990, 2022, doi: 10.14569/IJACSA.2022.01304112.

[22] M. Alojail and S. Bhatia, "A Novel Technique for Behavioral Analytics Using Ensemble Learning Algorithms in E-Commerce," *IEEE Access*, vol. 8, pp. 150072–150080, 2020, doi: 10.1109/ACCESS.2020.3016419.

[23] Ankit and N. Saleena, "An Ensemble Classification System for Twitter Sentiment Analysis," *Procedia Computer Science*, vol. 132, pp. 937–946, 2018, doi: 10.1016/j.procs.2018.05.109.

[24] A. Al-Laith and M. Shahbaz, "Tracking sentiment towards news entities from Arabic news on social media," *Future Generation Computer Systems*, vol. 118, pp. 467–484, May 2021, doi: 10.1016/j.future.2021.01.015.

[25] D. Xu, Z. Tian, R. Lai, X. Kong, Z. Tan, and W. Shi, "Deep learning based emotion analysis of microblog texts," *Information Fusion*, vol. 64, pp. 1–11, Dec. 2020, doi: 10.1016/j.inffus.2020.06.002.

[26] A. Rane and A. Kumar, "Sentiment Classification System of Twitter Data for US Airline Service Analysis," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Tokyo, Japan: IEEE, Jul. 2018, pp. 769–773, doi: 10.1109/COMPSAC.2018.00114.

[27] A. Thiab, L. Alawneh, and M. AL-Smadi, "Contextual emotion detection using ensemble deep learning," *Computer Speech and Language*, vol. 86, pp. 101604, Jun. 2024, doi: 10.1016/j.csl.2023.101604.

[28] L. Singh, S. Singh, and N. Aggarwal, "Two-Stage Text Feature Selection Method for Human Emotion Recognition," in *Lecture Notes in Networks and Systems*, vol. 46, pp. 531–538, 2019, doi: 10.1007/978-981-13-1217-5_51.

[29] N. Elgiriyewithana, "Emotion Dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/nelgiriyewithana/emotions. (Accessed: Dec. 2025).

BIOGRAPHIES OF AUTHORS

Nisha has received her master degree from UIET, Kurukshetra, and now she is pursuing Ph.D. in Computer Science & Applications from Department of Computer Science & Applications, Kurukshetra University, Kurukshetra (Haryana). Her current research work is in machine learning and NLP. She can be contacted at email: nisha.jrf.dcsa@kuk.ac.in.

Rakesh Kumar is a professor and chairman at the Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, Haryana, India. He obtained his B.Sc. Degree, Master's degree – Gold Medalist (Master of Computer Applications) and Ph.D. (Computer Science & Applications) from Kurukshetra University, Kurukshetra. His research interests are in genetic algorithms, software testing, artificial intelligence, and networking. He has published research papers in many esteemed international journals and participated in national and international conferences. He has been a member of technical program committees and organizing committees of various international and national conferences. He can be contacted at email: rakeshkumar@kuk.ac.in.