VAWKUM Transactions on Computer Sciences
http://vfast.org/journals/index.php/VTCS@ 2026, ISSN(e): 2308-8168, ISSN(p): 2411-6335
Volume 14, Number 1, January-June 2026 pp:40-52

From Machine Learning to Transformers: A Comparative Study on Emotion Classification of Tweets

Beenish Zafar[1*], **Ali Saeed**[2], **Hafiza Maria Kiran**[2], **Mobashirah Nasir**[3], **Afifa Hameed**[2]

[1]Department of Computer Science (FOIT), University of Central Punjab, Lahore, Pakistan; [2]Department of Software Engineering (FOIT), University of Central Punjab, Lahore, Pakistan; [3]School of Sciences and Technology, University of Management and Technology, Lahore, Pakistan

Keywords: *sentiment analyses, comparative study, machine learning, large language models, text classification, natural language processing*

Journal Info:
Submitted:
September 17, 2025
Accepted:
February 15, 2026
Published:
March 01, 2026

ABSTRACT

In this era of social media, it is essential that policymakers, businessmen, and researchers focus on understanding the sentiments of the people through their expressions on social media. An analytical comparison of several well-known machine learning (ML) algorithms along with Transformer-based models has been exploited in this study to determine the emotions and sentiments of people through their tweets. ML algorithms included in the study consider classical models—Random Forest, SVM, ID3, Naive Bayes, KNN, MLP—and transformer models, namely BERT-Base and BERT-Large. These algorithms have been applied to a carefully curated dataset of tweets that have been labeled for their emotion, such as joy, fear, anger, and sadness. Evaluation metrics, including precision, accuracy, kappa score, F1 score, and recall, have been used to compare and analyze the performance of each algorithm. The findings of this paper reveal that both Random Forest and SVM showed the best accuracy, at 79.4% and 79.1% while Random Forest slightly outperformed SVM in accuracy. In Transformers, BERT-Base outperforms not only BERT-Large but also the performance of all ML algorithms with an accuracy of 86.3%. The proposed study offers important insights for future applications and studies in the area of emotion classification. It also highlights the importance of choosing a suitable algorithm for tasks related to text categorization.

***Correspondence author email address:** beenish.zafar@ucp.edu.pk
DOI: 0.21015/vtcs.v14i1.2236

1 Introduction

In this modern era, several well-known digital platforms such as Facebook and Twitter have drastically altered the way in which a naive person can express his/her feelings on a range of topics, be it politics or education or entertainment. As the amount of available data continues to expand, it is the need of the hour to use this data intelligently. Data without being handled intelligently is of no use to the researchers and policy makers [1]. The analytical capability to interpret and derive insightful information from the available data is considered crucial in the current period. Sentiment analysis is a technique that helps in figuring out the tone behind a comment or a tweet made on social media using Natural Language Processing (NLP) [2]. In this way, sentiment analysis leads to

the investigation of textual data, which results in the extraction of subjective information.

With the recent advances in the field of machine learning, several algorithmic techniques have been proposed that ultimately boost the performance and robustness of sentiment analysis. This study, likewise, explores various machine learning algorithms as well as transformer-based models, including Random Forest, KNN, ID3, SVM, Naive Bayes, MLP, BERT-Base, and BERT-Large. Each algorithm is a very efficient and well-known algorithm. Some of these algorithms are good for classification, while some are good for regression tasks, and some perform well for feature extraction [3]. In this study, a comparative analysis of these algorithms has been performed to evaluate the performance of these algorithms. The proposed study investigates the most effective methods for accurate sentiment analysis.

To perform sentiment analysis, it is of extreme importance that the algorithm chosen is the right choice as the performance of each algorithm will be different from the others. The performance of algorithms depends mainly on the dataset used and the task at hand, which, in this case is sentiment analysis [3]. For example, Random Forest, due to its nature of being an ensemble method, might be able to perform very well on the problems related to predictive nature, as it leverages the power of decision trees during its execution [4]. However, SVM is a better choice when the problem at hand is a high-dimensional space [5].

The evaluation of models has been performed based on different metrics which include accuracy, kappa score, F1 score, precision and recall. Preliminary results have indicated that both Random Forest and SVM achieve the highest accuracy in ML algorithms and BERT-Base outperforms all transformers and also has exceptional performance as compared to the other algorithms. These findings underscore the significance as well as the effectiveness of these algorithms in the sentiment analysis of social media posts. This research has contributed in the ongoing efforts to draw insightful information from social media posts and also provides a future direction aimed at improving the interpretation and understanding of public sentiments.

This paper is organized as follows: Section 2 covers related work, Section 3 describes the datasets, Sections 4 and 5 outline the methodology and experiments, Section 6 presents the results and discussion, Section 7 concludes, and Section 8 presents future work.

2 Related Work

Sentiment analysis on platforms of social media, particularly Twitter, has become a task of huge importance if we want to understand customer feedback, political discourse and public opinion. Several studies have been organized to explore the deployment of machine learning as well as deep learning models for performing sentiment analysis on tweets-like text-based data. Previous research primarily depended on machine learning approaches such as Random Forest, SVM, and Naive Bayes.

These machine learning models depended on handcrafted features like n-grams, Bag-of-Words (BoW), and sentiment lexicons. Authors in [6] showed that models like Naive Bayes are effective for tweet sentiment analysis. Their work showed that even basic machine learning models are capable of achieving competitive results for sentiment analysis. Likewise, in [7], authors achieved promising results in classifying twitter data into three categories using SVM.

More recently, deep learning models, including Gated Recurrent Units (GRU), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks, have been utilized to capture the sequential patterns in Twitter text. In [8], the authors investigated the use of deep learning methods for sentiment analysis, showing that LSTM and GRU outperform traditional methods on text data. However, as observed in this study, these models often struggle with noisy and sparse social media text.

The introduction of transformer-based architectures, notably BERT, has enhanced sentiment analysis performance. BERT, proposed by Devlin et al. [9], is a pre-trained language model that performs exceptionally well across multiple NLP tasks. Wang et al. [11] showed that BERT outperforms traditional models and even deep learning approaches when fine-tuned on domain-specific tasks. In the context of Twitter sentiment analysis, In [12] the authors found that BERT-based

models, especially BERT-Base, significantly improve accuracy and robustness over other deep learning and machine learning approaches.

Some research has explored hybrid approaches as well. They deploy hybrid models by combining traditional machine learning with deep learning to attain improved results.

In [13], Zhang et al. introduced hybrid models that incorporate both feature-based and representation-based learning to improve sentiment classification, particularly on noisy datasets like Twitter. Several studies highlight the common challenges faced while working with Twitter data. These challenges include informal language, abbreviations, slang, and short text. In [14] the authors demonstrated that Twitter sentiment analysis requires domain adaptation and preprocessing strategies like tokenization, stopword removal, and handling emoticons and hashtags.

3 Dataset

The dataset that has been used for this study is named as Emotion Classification NLP, available at kaggle. The link to the dataset is available in the footnote [1]. It is composed of brief text snippets representing tweets. These tweets are annotated with a unique emotion out of fear, joy, anger and sadness. The dataset contains over 7000 entries which have been partitioned into subsets representing train, validation and test. The dataset is quite suitable for performing NLP related tasks as each entry in the dataset contains a single text field and its corresponding label. Thus, making it compatible with tasks related to supervised learning. This dataset has been chosen as it is quite relevant for analyzing emotions expressed through tweets in informal language or have been expressed implicitly.

The class distribution of all samples between fear, joy, anger and sadness has been visualized in Figure 1. It is evident from the visualization that the dataset is slightly imbalanced as fear is the only emotion that occurs most frequently while sadness is the least sampled emotion. The imbalance reflected in the dataset is a reflection of natural variance found in real-world emotional states of users on social media. Furthermore, a histogram, representing the lengths of tweets in dataset has also been illustrated in Figure 2. It shows that most of the tweets are of short length containng under 30 tokens. This further reinforces the characteristic of brevity of tweets.

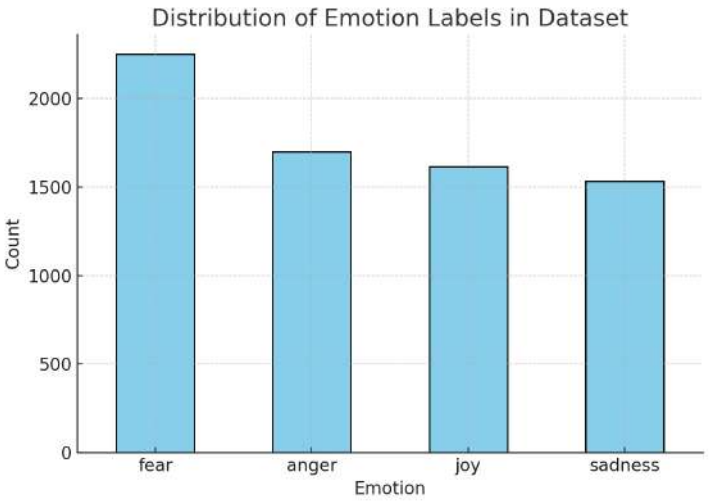

Figure 1. Class distribution of the dataset showing imbalance across four emotions.

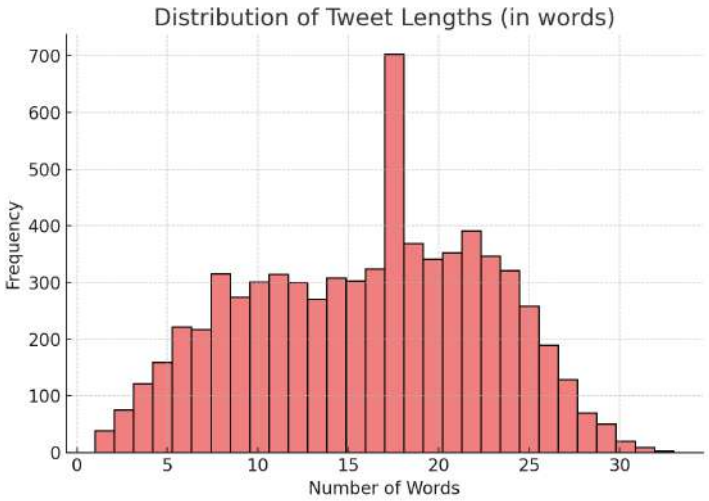

Figure 2. Text length distribution of samples, indicating that most texts are short (under 30 words).

Overall, the dataset offers a challenging yet practical testbed for evaluating emotion classification approaches. Its real-world nature, coupled with class imbalance and short text characteristics, makes it appropriate for benchmarking both machine learning algorithms as well as ad-

[1] Dataset available at https://www.kaggle.com/datasets/anjaneyatripathi/emotion-classification-nlp

vanced transformer-based architectures.

4 Proposed Methodology

In the following section, machine learning algorithms that are part of this study are discussed in part A. In part B, evaluation metrics, on the basis of which algorithms have been compared, are discussed. Whole flow of adopted methodology has been depicted in Figure 3.

4.1 Machine Learning Techniques

4.1.1 Random Forest

RF is considered one of the most widely used supervised learning algorithms. It is mainly exploited for classification and regression tasks. While performing classification tasks, it is used to handle categorical data, while for regression tasks, it handles continuous data. RF is based on an ensemble of decision trees, where each tree is used to generate a class prediction. After having all class predictions, the decision of final prediction is made by taking the majority vote (classification) or averaging (regression) all the trees.

RF uses bagging (Bootstrap Aggregating) to train multiple decision trees on several different random subsets of the data, helping reduce variance and overfitting. It also makes use of feature randomness which works by selecting random feature subsets for each split. This phenomena promotes diversity and reduces association among trees.

Random Forest calculates feature randomness as follows [15] :

$$RF_f^{ij} = \frac{1}{T} \sum_{j \in \text{all trees}} \text{norm} f_{ij} \tag{1}$$

Where:

RF_f^{ij} = Importance of feature i

$\text{norm} f_{ij}$ = Normalized value of feature i in tree j

T = Number of trees in the forest

4.1.2 Support Vector Machine

Support Vector Machine (SVM) comes under the umbrella of supervised ML algorithms. It can be used for classification as well as regression tasks. SVM works by plotting the data points in feature space that is n-dimensional. The task of classification is carried out by finding a hyperplane that splits the data points into unique classes in the best possible way. The main aim of SVM is to recognize an optimum hyperplane which will maximize the gap between the closest data points of opposite classes. This margin or distance is crucial for attaining a generalized model that can perform well on unseen data.

SVM divides the margins into two types: hard and soft margin. When data is linearly separable, hard margin approach may work well but it can also lead to over-fitting and miss-classification if we are dealing with noisy data or data that is not perfectly separable. In such cases, soft margin approach works well with noisy and overlapping data to provide a generalized model.

The decision boundary in SVM is represented by the following equation [16]:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \geq 0 \tag{2}$$

The inequality guarantees a clear division across the hyperplane, with points on one side labeled positive and classified into one class, and points on the other side labeled negative and classified into the other class.

4.1.3 Iterative Dichotomiser 3

A decision tree, also known as Iterative dichotomiser of ID3, is a robust algorithm that is used for prediction and classification. The name decision tree is given due to its structure which resembles a tree. In this tree each node or branch is representation of a feature that belongs to the dataset. By navigating these branches, outcome of a test data point can be determined. In this decision tree, each one of the leaf nodes represent a class label which will be predicted. In order to build a decision tree, information gain and entropy of each feature are calculated against the target variable. This helps in determining the order in which the featue nodes will be entered inside the decision tree. Once the tree has been constructed, the prediction of test data points is done by performing the traversal of the complete tree up to the leaf nodes.

Entropy and Information Gain are given using the following formulas. [17]:

$$E(S) = -\sum_{i=1}^{c} p_i \log_2 p_i \tag{3}$$

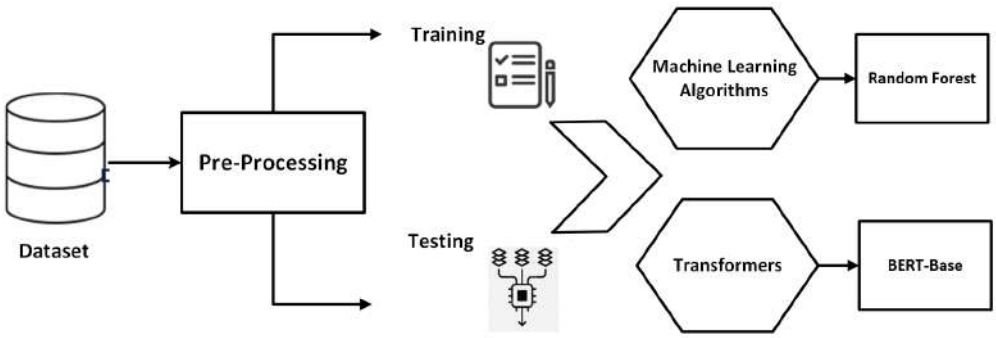

Figure 3. Methodology Flow Chart

$$IG(E) = 1 - \sum_{i=1}^{c} p_i^2 \tag{4}$$

4.1.4 Multi-Layer Perception

The multi-layer perceptron, known as MLP, belongs to a class of artificial neural networks (ANNs) that are composed of multiple layers of neurons and have the ability to transform input data in a structured and feed-forward manner. MLP typically has three types of layers. A layer for input that receives data, a few hidden layers that work on the data to transform it and a layer for output that generates predictions.

Every neuron that resides in a layer is completely connected to all neurons that belong to the next layer and a weighted sum of inputs is applied across it, which is then processed using a non-linear activation function. MLP is mostly used for supervised learning, as it proves to be very powerful for learning complex patterns in data. Provided a considerable amount of data and enough computational power, they can be particularly effective in capturing nonlinear relationships.

The calculation is given in the following equations [18]:

$$z_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \tag{5}$$

$$a_j = f(z_j) \tag{6}$$

4.1.5 Naive Bayes

Naive Bayes is a probability based algorithm which is simple yet powerful. It is mostly used in classification tasks. Naive Bayes works on the basis of Bayes' Theorem, which computes the likelihood of a hypothesis on the basis of given data. The algorithm assumes that the features are conditionally unrelated. But it is often not the case in real-life data, but due to this naive assumption, the calculations are simplified and results are surprisingly well in practice. Naive Bayes is really effetive in text classification and often used in practices like sentiment analysis and spam detection. Although it is a simple algorithm, yet it performs well with large datasets and can prove to be robust in noise handling.

The calculation of Naive Bayes is represented in the following equations [19]:

$$P(C \mid X) = \frac{P(X \mid C) \cdot P(C)}{P(X)} \tag{7}$$

$$P(X \mid C) = \prod_{i=1}^{n} P(x_i \mid C) \tag{8}$$

$$P(C \mid X) \propto P(C) \prod_{i=1}^{n} P(x_i \mid C) \tag{9}$$

4.1.6 K-Nearest Neighbor

K-Nearest Neighbors (KNN) is also a type of supervised machine learning algorithm and can be used for classification as well as regression tasks. In KNN, the predictions are computed by calculating the distance between the test point and all training data points. This distance is calculated on the basis of the Euclidean distance formula. However, depending upon the scenario, Manhattan distance or Minkowski distance can also be used.

After this distance has been computed, selection of the best K data points is conducted, and the then for classification, it predicts based on the most frequent class. One key characteristic of KNN is that it has no explicit training phase, making it fast to train, but it is slow during prediction due to the runtime computations required to calculate the distances between the test and training points.

The mathematical formula to calculate the Euclidean distance is given in [20]:

$$\text{EuclideanDistance} = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{10}$$

4.1.7 Logistic Regression

Logistic regression also comes under supervised machine learning. It is mostly used for two-fold classification, although it can be used for multi-class problems by employing some solution like softmax regression. The algorithm checks the probability of a test input belonging to a certain class, mostly through the sigmoid function. Sigmoid matches a given input to a value between 0 and 1. The model then tries to learn the ideal weights by minimizing the log probability or loss by applying optimization methods such as gradient descent. After the training phase, class labels are assigned to the test data according to a threshold on the predicted probability.

Logistic regression is an efficient as well as interpretable model that provides probabilities. But one of its drawbacks can be the assumption of a linear connection among features and the target. It can cause issues while working with complex and non-linear problems. Moreover, it can be reactive to anomalies. Although it has its limitations, still it is often used in applications such as medical diagnosis and credit scoring. [21].

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n)}} \tag{11}$$

4.2 Transformers

4.2.1 BERT-Base

BERT, a pre-trained transformer-based language model, was introduced by Google in 2018 [22]. By leveraging bidirectional context, the model significantly advanced NLP, taking into account words both before and after a target word in a sentence. Unlike previous models, which only read text in a single direction, BERT's bidirectional nature allows it to better understand the meaning of words based on their context. BERT-base is a compact variant of the BERT architecture, consisting of 12 layers, 768 hidden units, and approximately 110 million parameters. Pre-training is performed on vast text corpora using MLM and NSP objectives. The model is adapted for NLP tasks—question answering, text classification, and named entity recognition—attaining leading performance across numerous benchmarks.

4.2.2 BERT-Large

BERT-large is an improved version of the BERT model, designed to provide higher capacity and more powerful representations for natural language understanding tasks [22]. It extends the original BERT-base model by increasing the number of layers and parameters. With 24 layers, BERT-large consists of 1,024 hidden units and roughly 340 million parameters and is pre-trained using MLM and NSP tasks, similar to BERT-base, but its larger architecture enables it to capture more complex language patterns and nuances. The increased model capacity allows BERT-Large to achieve better performance than BERT-Base on downstream tasks, including question answering, sentiment analysis, and text classification. Although computationally more demanding, its strong performance makes it suitable for state-of-the-art NLP systems.

4.3 Evaluation Metrics

Several evaluation metrics are used to judge the performance of algorithms by researchers. These metrics enable a comprehensive evaluation and comparison of the algorithms' effectiveness. The presented methodology is assessed on the basis of the following metrics:

4.3.1 Accuracy

Accuracy is one of the metrics that is commonly used for the evaluation of classification algorithms in ML and DL. It is computed as the proportion of correctly predicted instances out of all input data points. The formula for

accuracy is given by Equation 12:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Where:

- TP denotes True Positive instances
- TN denotes True Negative instances
- FP denotes False Positive instances
- FN denotes False Negative instances

4.3.2 Recall

Recall is another performance metric which is used for performance evaluation of the classification algorithms. It determines the true positive rate as the proportion of correctly predicted positive instances. Recall is formulated as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

4.3.3 Precision

Precision is another important performance metric which is used for classification algorithms. It evaluates the proportion of correctly predicted positive instances among all positive predictions in the test set. Precision can be formulated as the ratio of TPs to the sum of TPs and FPs. A higher precision is an indication that the algorithm makes few false positive predictions. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$

4.3.4 F1 Score

F1 score is another important metric for evaluating classification algorithms in ML. It is extremely useful in cases where an issue of class imbalance exists, as it integrates both precision and recall together in its calculation. The F1 score eventually combines the two prominent metrics, precision and recall, and later calculates their harmonic mean by providing a single value that is used to balance both metrics. When both precision and recall are high, the F1 score will also be high. Conversely, if both metrics are low, the F1 score will be low as well. If one of the metrics is high but the other one is low, the F1 score will yield a moderate value, reflecting the balance between precision and recall.

$$\text{F1 Score} = \frac{2 \times \text{P} \times \text{R}}{\text{P} + \text{R}} \tag{15}$$

5 Experiments

5.1 Preprocessing for Transformer Models

For transformer models, the preprocessing has been mainly done by utilizing the capabilities of transformer-based models, especially BERT. The text columns served as the source of features while label columns were used to extract the target labels. BERT tokenizer was responsible for dealing with the preprocessing of textual data of tweets. This tokenizer has many features, including the ability to perform wordpiece tokenization, conversion of tokens into IDs, conversion to lowercase and generation of attention masks responsible for differentiating between meaningful and important tokens from padded locations. Sequences were padded or truncated to a length of maximum 128 tokens. This padding was done to ensure uniform batches. All this preprocessing resulted into tokenized data and encoded labels, which was then converted to TensorFlow datasets. For efficient and optimized model training, the dataset was shuffled and batched. BERT can effectively and directly capture all the syntactic and semantic relations from raw data, thus reducing the need of traditional preprocessing steps like stemming or lemmatization.

5.2 Preprocessing for Machine Learning Models

In this paper, a uniform preprocessing pipeline was created for the traditional machine learning algorithms, which included logistic regression, SVM, NB, KNN, RF, and MLP. The raw text data was first encoded into numerical feature vectors using the TF-IDF approach. This method assigns greater importance to terms that are more distinctive across documents while reducing the weight of common terms, thus creating a sparse and high-dimensional feature space ideal for text classification tasks. The TF-IDF vectorizer was learned from the training data and applied to the test data to preserve feature consistency.

Unlike transformer models, machine learning algorithms do not necessitate special steps like tokenization, truncation or padding. Additionally, no further preprocessing methods such as stopword removal, stemming, or lemmatization were used, as the TF-IDF representation naturally handles word frequency distributions.

The resulting feature vectors served as input for each machine learning model, with the target emotion labels kept in their categorical form. To mitigate potential convergence issues, especially in logistic regression and MLP, the maximum number of iterations was increased. Overall, this pipeline offered a consistent framework for evaluating the performance of various machine learning algorithms in comparison to transformer-based models.

All machine learning algorithms were trained and tested on the dataset for effective evaluation and comparison of sentiment analysis. The algorithms included Random Forest, SVM, ID3, MLP, Naive Bayes and KNN. Each model was trained and evaluated on the same processed dataset to ensure consistent comparison. The same dataset was also trained leveraging transformer models, specifically BERT-Base and BERT-Large. We assessed their performance based on accuracy, Kappa statistic, precision, recall, and F1-score. This approach allowed us to identify the strengths and limitations of each algorithm in accurately classifying emotions, with the goal of determining the most effective model for sentiment analysis on Twitter data. The link of complete code is given in the footnote [2]. Following algorithm defines the experimentation conducted for this study.

Table 1. Comparison of Accuracy and Kappa Score for Various Algorithms

Algorithm	Accuracy	Kappa Score
Machine Learning Methods		
Random Forest (RF)	0.794	0.721
Support Vector Machine (SVM)	0.791	0.718
Iterative Dichotomiser 3 (ID3)	0.784	0.710
Multi-Layer Perceptron (MLP)	0.732	0.640
Naive Bayes (NB)	0.621	0.472
k-Nearest Neighbors (KNN)	0.553	0.395
Logistic Regression	0.783	0.706
Transformer Models		
BERT-Base	0.863	0.82
BERT-Large	0.702	0.60

6 Results and Discussion

Table 1 compares accuracy and Kappa score for a variety of algorithms across two categories: machine learning methods and transformer models. This comparison provides insights into the strengths and limitations of the different models when applied to the given dataset. The comparison of various models reveals that BERT-Base outperforms all others with an accuracy of 86.3% and a Kappa score of 0.82, highlighting its superiority in handling text-based tasks due to its ability to capture complex contextual relationships. Among all implemented ML techniques, RF as well as SVM delivered strong performances, with accuracies of 79.4% and 79.1%, respectively, while traditional models like Naive Bayes and KNN underperformed. Overall, transformer-based models like BERT outperform other approaches.

Interestingly, after fine-tuning the model BERT-Large, it still performed a lot worse than BERT-Base with only 70.2% accuracy as compared to 86.3% accuracy of BERT-Base.This difference is largely influenced by dataset size and model capacity, as BERT-Large comprises about 335M parameters—nearly three times that of BERT-Base (110M). This reason can increase the chance of overfitting if a dataset of moderate size, such as Kaggle Emotion Classification dataset, which is used in this study, is used to train BERT-Large. Moreover, during the process of fine-tuning, BERT-Large proves to be much more sensitive to hyperparameters such as number of epochs, learning rate, and batch size. Lack of

Algorithm 1. Generalized Emotion Classification Model Training and Evaluation

Input: Training dataset D_{train}, validation dataset D_{val} (optional), test dataset D_{test}, set of models $\mathcal{M}$
Output: Trained models and evaluation metrics
Steps:

1. For each model $m \in \mathcal{M}$:
   (a) Initialize model m
   (b) Train m on D_{train}
   (c) If D_{val} is available, validate m on D_{val}
   (d) Test m on D_{test}
   (e) Compute evaluation metrics: Accuracy, Precision, Recall, F1-score, Confusion Matrix, Cohen's Kappa
2. Compare all models based on evaluation metrics
3. Return trained models and evaluation results

[2] Link of the code: https://colab.research.google.com/drive/1dyfCvjtx1O-j1da4SvhoG95mNqN3p9uT?usp=sharing

Table 2. Performance metrics for various models on different emotions classes.

Model and Class	Precision	Recall	F_1 Measure
ML Methods Results			
SVM (Joy)	0.88	0.82	0.85
SVM (Anger)	0.82	0.75	0.78
SVM (Fear)	0.75	0.82	0.78
SVM (Sadness)	0.74	0.76	0.75
NB (Joy)	0.90	0.52	0.66
NB (Anger)	0.84	0.51	0.63
NB (Fear)	0.48	0.94	0.64
NB (Sadness)	0.80	0.39	0.52
KNN (Joy)	0.61	0.49	0.54
KNN (Anger)	0.50	0.61	0.55
KNN (Fear)	0.55	0.63	0.59
KNN (Sadness)	0.58	0.45	0.51
ID3 (Joy)	0.85	0.83	0.84
ID3 (Anger)	0.79	0.76	0.78
ID3 (Fear)	0.76	0.78	0.77
ID3 (Sadness)	0.75	0.77	0.76
RF (Joy)	0.89	0.82	0.85
RF (Anger)	0.85	0.73	0.79
RF (Fear)	0.71	0.86	0.78
RF (Sadness)	0.80	0.73	0.76
LR (Joy)	0.87	0.79	0.83
LR (Anger)	0.83	0.74	0.78
LR (Fear)	0.72	0.85	0.78
LR (Sadness)	0.77	0.73	0.75
MLP (Joy)	0.82	0.74	0.78
MLP (Anger)	0.75	0.73	0.74
MLP (Fear)	0.73	0.75	0.74
MLP (Sadness)	0.63	0.71	0.67
Transformer Models Results			
BERT-Base (Joy)	0.91	0.88	0.89
BERT-Base (Anger)	0.87	0.84	0.85
BERT-Base (Fear)	0.85	0.85	0.85
BERT-Base (Sadness)	0.82	0.90	0.86
BERT-Large (Joy)	1.00	0.22	0.33
BERT-Large (Anger)	0.63	0.93	0.75
BERT-Large (Fear)	0.89	0.78	0.83
BERT-Large (Sadness)	0.57	0.86	0.69

optimization may lead towards a less generalized model. Due to resource limitations, large language models are often trained using small batch sizes to maintain memory limitations, which can cause unstable dynamics during the training process. All these factors point towards the justification of BERT-Base as proving itself as the most effective and reliable model for multi-class emotion recognition. This finding is consistent with earlier studies that suggest overly large transformer models do not always enhance performance on small-sized to medium-sized NLP datasets [9] [10].

The detailed results of each algorithm are given in table 2. The table presents the performance metrics—precision, recall, and F1 scores for six ML models (SVM, Naive Bayes, KNN, ID3, RF, and MLP) and BERT variants (BERT-Base, BERT-Large) were compared across four emotions: Joy, Anger, Fear, and Sadness. SVM showed particularly strong performance for Joy (P: 0.88, R: 0.82, F1: 0.85) and Anger (P: 0.82, R: 0.75, F1: 0.78).

In contrast, Naive Bayes struggles with precision and recall, especially for Fear (precision: 0.48, recall: 0.94), resulting in a low F1 score of 0.64. KNN generally underperforms across all classes, with its highest F1 score at 0.59 for Fear. ID3 demonstrates balanced performance, closely matching SVM and Random Forest for joy and anger. Random Forest excels in both Joy (precision: 0.89, recall: 0.82, F1: 0.85) and Fear (precision: 0.71, recall: 0.86, F1: 0.78), showcasing its effectiveness in emotion classification. MLP has lower scores, particularly in sadness (F1: 0.67), indicating variability in model perfor-

mance based on the emotion being classified. Overall, SVM and RF demonstrate the highest reliability for this task, whereas NB and KNN exhibit significant limitations.

Among BERT models, two variants—BERT-Base and BERT-Large—were evaluated. BERT-Base shows high effectiveness, especially for Joy (P: 0.91, R: 0.88, F1: 0.89), Anger (P: 0.87, R: 0.84, F1: 0.85), and Fear (P: 0.85, R: 0.85, F1: 0.85). The F1 score for all classes indicates that BERT-Base strikes a balance between precision and recall. BERT-Large, while having perfect precision for Joy (1.00), exhibits a major weakness in recall for Joy (0.22). Upon examining BERT-Large's performance across different classes, a significant imbalance is evident in the "Joy" category. The model achieved a flawless precision score of 1.00, yet its recall was merely 0.22. This suggests that while the model accurately identifies "Joy" when it predicts it, it misses most actual instances of "Joy."

This issue likely stems from a mix of data sparsity, with relatively few "Joy" samples in the dataset, class imbalance, and overfitting due to BERT-Large's large number of parameters compared to the dataset size. Furthermore, the model might be overly confident in a few high-probability predictions, which boosts precision but negatively impacts recall. These findings underscore the necessity of tracking class-specific metrics, rather than just overall accuracy. Future research could address this problem by employing class rebalancing strategies, such as using weighted loss functions, augmenting data for minority classes, or oversampling "Joy" examples, to enhance recall without sacrificing precision.

It performs better for Anger (precision: 0.63, recall: 0.93, F1: 0.75) and Fear (precision: 0.89, recall: 0.78, F1: 0.83), but struggles with Sadness (precision: 0.57, recall: 0.86, F1: 0.69). This discrepancy between precision and recall suggests that the model performs well at identifying certain classes (especially Anger), but it is not as good at capturing all the instances of other classes.

To contextualize our achievement of attaining 86.3% accuracy of BERT-Base on the Emotion Classification dataset, we have compared our results with several recent peer-reviewed papers. These papers have also worked on emotion classification tasks on Twitter-based data and achieved notable results. In [45], the authors fine-tuned BERT to achieve 94% macro-accuracy in emotion classification of Twitter data. Transformer-based models were shown to perform strongly in emotion classification of social media posts. In [44], hybrid transformer models BERT-BiLSTM and distilled BERT achieved a very high accuracy rate lying between 92.45% and 97.35%.

The authors in [43] used a joint classifier combining RoBERTa and BERT to achieve accuracy of 87.22% in sentiment analysis of short, tweet-like textual data. Conversely, in [42], BERT performed poorly for a language-specific study, reporting an accuracy of just 61%. This study illustrates that the characteristics, quality, and preprocessing of the dataset can influence the results of emotion classification. These findings from literature indicate that our results of BERT-Base, with 86.3% accuracy, are consistent with the benchmarks. However, adopting more advanced hybrid approaches of complex transformer models may lead to higher and improved accuracy in emotion classification tasks of tweet-like data.

6.1 Comparison with Existing Studies

A fair comparison of emotion classification models requires evaluation on the same dataset. The Emotion Classification NLP (Kaggle) dataset used in this study (7,000 English tweets labeled as joy, anger, fear, and sadness) has not been previously benchmarked in peer-reviewed literature using both classical machine learning and transformer-based models.

A detailed review indicates that no published study has reported comprehensive multi-model evaluation, transformer comparison (BERT-Base vs. BERT-Large), Cohen's Kappa, and per-class analysis on this specific dataset. Therefore, direct numerical comparison with prior work is not possible.

This establishes the novelty of the present study. To the best of our knowledge, this is the first peer-reviewed work to provide a systematic benchmark on this dataset across seven classical machine learning algorithms and two transformer architectures under a unified experimental setup. The results serve as baseline references for future research using the same dataset.

7 Conclusion

This analysis compared the performance of ML and BERT-based models for emotion classification on Twitter data. Random Forest (RF) and SVM emerged as the best-performing machine learning models, achieving high accuracy and kappa scores. Among BERT-based models, BERT-Base outperformed others with an accuracy of 0.863, show-casing its strength in capturing complex sentiment. Overall, BERT-based models showed the most promise, while traditional machine learning models also remain reliable for emotion classification tasks.

8 Future Work

In this study the issue of class imbalance has not been addressed which may have been a cause of weaker performance on under-represented classes like sadness and joy. Future work could explore several techniques like data augmentation, resampling or cost sensitive learning to have a better representation of minority classes.

Author Contributions

Beenish Zafar: Conceptualization, Supervision, Methodology, Experimentation, Formal Analysis, Writing - Original Draft, Corresponding Author **Ali Saeed**: Data curation, Project Administration, Model Validation, Writing - Reviewing and Editing **Hafiza Maria Kiran**: Visualization, Investigation. **Mobashirah Nasir**: Critical Review, Research Guidance **Afifa Hameed**: Writing-Reviewing and Editing, Final Approval of Manuscript

Compliance with Ethical Standards

It is declared that none of the authors has any conflict of interest. It is also declared that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

References

[1] O. Adwan, M. Al-Tawil, A. Huneiti, R. Shahin, A. A. Zayed, and R. Al-Dibsi, "Twitter sentiment analysis approaches: A survey," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 15, pp. 79–93, 2020.

[2] A. D. Dubey, "Twitter sentiment analysis during covid-19 outbreak," *Available at SSRN 3572023*, 2020.

[3] H. Huang, A. A. Zavareh, and M. B. Mustafa, "Sentiment analysis in e-commerce platforms: A review of current techniques and future directions," *IEEE Access*, vol. 11, pp. 90 367–90 382, 2023.

[4] L. Mandloi and R. Patel, "Twitter sentiments analysis using machine learninig methods," in *2020 International Conference for Emerging Technology (INCET)*. IEEE, 2020, pp. 1–5.

[5] K. H. Manguri, R. N. Ramadhan, and P. R. M. Amin, "Twitter sentiment analysis on worldwide covid-19 outbreaks," *Kurdistan Journal of Applied Research*, pp. 54–65, 2020.

[6] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[7] A. Pak, P. Paroubek *et al.*, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREc*, vol. 10, no. 2010, 2010, pp. 1320–1326.

[8] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[9] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint arXiv:1907.11692*, 2019.

[11] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, "Pre-trained language models and their applications," *Engineering*, vol. 25, pp. 51–65, 2023.

[12] A. A. Chowdhury, A. Das, S. K. Saha, M. Rahman, and K. T. Hasan, "Sentiment analysis of covid-19 vaccination from survey responses in bangladesh." 2021.

[13] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[14] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" in *Proceedings of the international AAAI conference on web and social media*, vol. 5, no. 1, 2011, pp. 538–541.

[15] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[16] V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[17] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp. 81–106, 1986.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[19] H. Zhang, "The optimality of naive bayes," *Aa*, vol. 1, no. 2, p. 3, 2004.

[20] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning*, vol. 6, pp. 37–66, 1991.

[21] M. P. LaValley, "Logistic regression," *Circulation*, vol. 117, no. 18, pp. 2395–2399, 2008.

[22] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, no. 2. Minneapolis, Minnesota, 2019.

[23] A. Mitra and S. Mohanty, "Sentiment analysis using machine learning approaches," *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS*, vol. 2, pp. 63–68, 2020.

[24] Y. Wang, J. Guo, C. Yuan, and B. Li, "Sentiment analysis of twitter data," *Applied Sciences*, vol. 12, no. 22, p. 11775, 2022.

[25] N. Braig, A. Benz, S. Voth, J. Breitenbach, and R. Buettner, "Machine learning techniques for sentiment analysis of covid-19-related twitter data," *IEEE Access*, vol. 11, pp. 14 778–14 803, 2023.

[26] D. Antonakaki, P. Fragopoulou, and S. Ioannidis, "A survey of twitter research: Data model, graph structure, sentiment analysis and attacks," *Expert systems with applications*, vol. 164, p. 114006, 2021.

[27] F. Barbieri, L. E. Anke, and J. Camacho-Collados, "Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond," *arXiv preprint arXiv:2104.12250*, 2021.

[28] Z. B. Nezhad and M. A. Deihimi, "Twitter sentiment analysis from iran about covid 19 vaccine," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, vol. 16, no. 1, p. 102367, 2022.

[29] A. S. Neogi, K. A. Garg, R. K. Mishra, and Y. K. Dwivedi, "Sentiment analysis and classification of indian farmers' protest using twitter data," *International Journal of Information Management Data Insights*, vol. 1, no. 2, p. 100019, 2021.

[30] Y. Qi and Z. Shabrina, "Sentiment analysis using twitter data: a comparative application of lexicon-and machine-learning-based approach," *Social Network Analysis and Mining*, vol. 13, no. 1, p. 31, 2023.

[31] X. Liu, T. Shi, G. Zhou, M. Liu, Z. Yin, L. Yin, and W. Zheng, "Emotion classification for short texts: an improved multi-label method," *Humanities and Social Sciences Communications*, vol. 10, no. 1, pp. 1–9, 2023.

[32] I. Ameer, N. Bölücü, M. H. F. Siddiqui, B. Can, G. Sidorov, and A. Gelbukh, "Multi-label emotion classification in texts using transfer learning," *Expert Systems with Applications*, vol. 213, p. 118534, 2023.

[33] Z. Li, H. Xie, G. Cheng, and Q. Li, "Word-level emotion distribution with two schemas for short text emotion classification," *Knowledge-Based Systems*, vol. 227, p. 107163, 2021.

[34] C. Singla, F. N. Al-Wesabi, Y. S. Pathania, B. S. Alfurhood, A. M. Hilal, M. Rizwanullah, M. A. Hamza, and M. Mahzari, "An optimized deep learning model for emotion classification in tweets." *Computers, Materials & Continua*, vol. 70, no. 3, 2022.

[35] N. Jamal, C. Xianqiao, F. Al-Turjman, and F. Ullah, "A deep learning–based approach for emotions classification in big corpus of imbalanced tweets," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 3, pp. 1–16, 2021.

[36] A. Chiorrini, C. Diamantini, A. Mircoli, D. Potena *et al.*, "Emotion and sentiment analysis of tweets using bert." in *Edbt/icdt workshops*, vol. 3, 2021, pp. 1–7.

[37] A. Glenn, P. LaCasse, and B. Cox, "Emotion classification of indonesian tweets using bidirectional lstm," *Neural Computing and Applications*, vol. 35, no. 13, pp. 9567–9578, 2023.

[38] J. F. Raisa, M. Ulfat, A. Al Mueed, and S. S. Reza, "A review on twitter sentiment analysis approaches," in *2021 international conference on information and communication technology for sustainable development (ICICT4SD)*. IEEE, 2021, pp. 375–379.

[39] F. Rustam, M. Khalid, W. Aslam, V. Rupapara, A. Mehmood, and G. S. Choi, "A performance comparison of supervised machine learning models for covid-19 tweets sentiment analysis," *Plos one*, vol. 16, no. 2, p. e0245909, 2021.

[40] A. Poornima and K. S. Priya, "A comparative sentiment analysis of sentence embedding using machine learning techniques," in *2020 6th international conference on advanced computing and communication systems (ICACCS)*. IEEE, 2020, pp. 493–496.

[41] J. Xue, J. Chen, R. Hu, C. Chen, C. Zheng, Y. Su, and T. Zhu, "Twitter discussions and emotions about the covid-19 pandemic: Machine learning approach," *Journal of medical Internet research*, vol. 22, no. 11, p. e20550, 2020.

[42] M. H. Algifari and E. D. Nugroho, "Emotion Classification of Indonesian Tweets using BERT Embedding," *Journal of Applied Informatics and Computing*, vol. 7, no. 2, pp. 172–176, 2023.

[43] L. He, "Enhanced Twitter sentiment analysis with dual joint classifier integrating RoBERTa and BERT architectures," *Frontiers in Physics*, vol. 12, p. 1477714, 2024, Frontiers Media SA.

[44] A. Sathish *et al.*, "Intelligent emotion sensing using BERT BiLSTM and generative AI for proactive customer care," *Scientific Reports*, vol. 15, no. 1, pp. 1–22, 2025, Nature Publishing Group.

[45] T. Nijhawan, G. Attigeri, and T. Ananthakrishna, "Stress detection using natural language processing and machine learning over social interactions," *Journal of Big Data*, vol. 9, no. 1, p. 33, 2022, Springer.