Towards Explainable Sentiment Analysis: A Hybrid BiLSTM-SVM Approach for the 2025 Nepal Protests

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTERS OF SCIENCE
IN
APPLIED MATHEMATICS

Submitted by

Madhusree Purkayastha (24/MSCMAT/40)

Krishna MalliK (24/MSCMAT/17)

Under the supervision of

Dr. Sumedha Seniaray

DEPARTMENT OF APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY, 2026

DEPARTMENT OF MECHANICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

We, **Madhusree Purkayastha (24/MSCMAT/40)** and **Krishna Mallik (24/MSCMAT/17)**, students pursuing an M.Sc. in Applied Mathematics at Delhi Technological University, hereby affirm that the project dissertation entitled "**Towards Explainable Sentiment Analysis: A Hybrid BiLSTM–SVM Approach for the 2025 Nepal Protests**" submitted for partial fulfilment of our Master of Science degree, is an authentic documentation of our independent research conducted throughout the 2024–25 academic term.

This material has not been submitted previously to obtain any degree, diploma, fellowship, or equivalent academic recognition at this or any alternative institution. Furthermore, all external data sources, secondary literature, and published references utilized in this document have been appropriately cited following recognized academic conventions.

Madhusree Purkayastha (24/MSCMAT/40)

Krishna Mallik (24/MSCMAT/17)

Place: Delhi

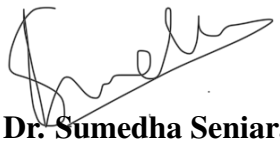
Date: 23.05.2026

DEPARTMENT OF MECHANICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

This serves as certification that the dissertation titled "**Towards Explainable Sentiment Analysis: A Hybrid BiLSTM–SVM Approach for the 2025 Nepal Protests**" presented by Madhusree Purkayastha (24/MSCMAT/40) and Krishna Mallik (24/MSCMAT/17) to fulfill partial requirements for the Master of Science degree in Applied Mathematics at Delhi Technological University, represents a genuine record of research executed under my academic supervision during the 2025–26 session.

To the best of my awareness, no part of this manuscript has been submitted for a degree or diploma at this or any other educational establishment.



Place: Delhi

Date: 23.05.2026

Dr. Sumedha Seniaray
(SUPERVISOR)
Assistant Professor
Department of Applied Mathematics
Delhi Technological University

DEPARTMENT OF MECHANICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to **Dr. (Prof.) R. Srivastava**, Head of the Department of Applied Mathematics at Delhi Technological University, along with all the faculty members of the department, for providing the academic environment, facilities, and encouragement necessary for carrying out this work.

We are especially thankful to our project supervisor, **Dr. Sumedha Seniaray**, Assistant Professor in the Department of Applied Mathematics, for her continuous guidance and support throughout the course of this research. Her valuable suggestions, constructive feedback, and technical advice greatly contributed to the successful completion of this dissertation. Her encouragement and mentorship helped us approach both the theoretical and practical aspects of the project with greater clarity and confidence.

We also extend our appreciation to our classmates and friends for their useful discussions, suggestions, and support during different stages of the project. Their cooperation and shared learning experience were genuinely helpful throughout this journey.

Finally, we are deeply grateful to our families for their constant encouragement, patience, and emotional support. Their trust and motivation played an important role in helping us complete our postgraduate studies successfully.

Place: Delhi — Madhusree Purkayastha (24/MSCMAT/40)

Date: 23.05.26 — Krishna Mallik (24/MSCMAT/17)

Abstract

The Nepal Protest of 2025 generated a massive discourse among users, creating a huge corpus of comments with varying perspectives, sentiments, and concerns regarding the political climate at that time. It is imperative to analyze this data in order to gain insight into how society perceives this political scenario and its impact on their lives. This paper explores a hybrid explainable sentiment analysis framework utilizing YouTube commentaries made in the context of 2025 Nepal Protests.

A total of 10,000 comments from YouTube related to the Nepalese protests of 2025 were collected using the YouTube Data API, followed by preprocessing in the form of language filtering, removing noise, converting to lowercase, lemmatizing, negation handling, and tokenization. Baseline models such as Multinomial Naive Bayes, Logistic Regression, SVM, CNN, and BiLSTM were used in this study with TF-IDF and padded sequences.

Based on these baselines, a novel BiLSTM-SVM framework is proposed wherein the BiLSTM framework acts as a deep feature extractor and its output features are fed to an SVM classifier for sentiment classification. The performance of this hybrid system was recorded at an average accuracy of 88% along with a precision, recall, and F1-score of 0.87 in a macro-average. To resolve the issue of interpretability in this model, SHAP (SHapley Additive exPlanations) framework was applied, highlighting the most significant words that led to particular sentiment classes. Words like ”corrupt,” ”violent,” and ”destroy” had a strong impact on negative sentiment classes, whereas ”good,” ”love,” and ”peaceful” contributed to positive predictions. Sentiment distribution analysis across the data corpus showed that there was a positive trend in sentiment owing to reformist and civic sentiment classes, with a large number of negative sentiments due to dissatisfaction with political institutions amid the protests.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

The rapid increase in digital communication has resulted in an unprecedented volume of publicly available user-generated content across social media platforms, online forums and news comment sections. Such data often reflects public opinion, emotional responses and social behaviour toward political and societal events. One such event that generated widespread global attention was the Nepal Protests of 2025, during which thousands of individuals expressed their views online in real time.

Sentiment analysis aims to determine whether a piece of text expresses a positive, negative or neutral emotion. The ability to computationally analyze public sentiment during major socio-political movements provides deep insights into public dissatisfaction and changing opinion trends. This project focuses on building an end-to-end sentiment analysis system to evaluate the overall sentiment surrounding the Nepal Protests of 2025 using both machine learning and deep learning methods.

1.2 Problem Formulation

User-generated protest-related content puts forward several challenges for sentiment classification. The raw dataset contains noise such as URLs, negation inconsistencies, non-English text, emojis and special characters. Moreover, protest-related discourse often includes negations, sarcasm and emotionally expressions that makes sentiment identification more difficult.

Formally, the sentiment classification problem can be expressed as positive,negative and neutral. The objective is to develop a robust model capable of accurately classifying public sentiment during the Nepal Protests of 2025.

1.3 Objectives

The key objectives of this research are as follows:

- To collect and analyze a large dataset of user comments related to the Nepal Protests of 2025.
- To clean and preprocess the text data through normalization, noise removal, lemmatization and negation handling.
- To extract meaningful textual features using TF–IDF and sequence-based embeddings.
- To use traditional machine learning classifiers such as Multinomial Naive Bayes, Logistic Regression, and SVM.
- To build and train deep learning models such as Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (LSTM).
- To evaluate the performance of all models based on accuracy, precision, recall, and F1 score.
- To interpret the final sentiment distribution to understand the public mood surrounding the Nepal Protests of 2025.

1.4 Motivation

Public sentiment plays a crucial role in shaping the outcomes of social and political movements, making it important to understand the emotion of the population during the 2025 Nepal Protests. It is motivated by the rising relevance of social media analytics for capturing real-time public opinion, the growing need for automated tools capable of processing large-scale political discourse, and the academic value of comparing machine learning and deep learning for sentiment analysis. By addressing these factors, the work aims to bridge the gap between raw, unstructured online conversations and meaningful, quantifiable insights that can support policymakers, researchers, journalists and civil society organizations in understanding societal sentiment during a major socio-political event.

Chapter 2

LITERATURE REVIEW

Sentiment analysis has grown into a dominant sub-field of NLP, driven by the massive influx of user-generated data across the web [1, 2]. Methodologies have shifted significantly, moving away from basic dictionary-matching techniques toward highly complex statistical and neural network designs [3,4]. This chapter reviews the fundamental academic literature shaping the current study, touching upon classical machine learning, recurrent architectures, hybrid models, and AI interpretability.

Contemporary literature highlights the necessity of monitoring shifting narratives during periods of crisis. For example, Prama *et al.* [5] mapped emotional volatility and shifting public attitudes during the July 2024 student-led uprising in Bangladesh. Their research provides a foundational blueprint for tracking discourse, emotion, and sentiment during severe political conflicts, which directly applies to the challenges of analyzing the 2025 Nepal protests. Moreover, Singh and Thomas [6] executed a comprehensive systematic review focused entirely on YouTube sentiment analysis. They emphasized the distinct difficulties of processing vast amounts of noisy, slang-filled data, reiterating the demand for specialized, robust analytical workflows.

2.1 Traditional Machine Learning for Sentiment Classification

Initial automated sentiment frameworks leaned heavily on statistical learning utilizing hand-crafted feature vectors. Ranjan *et al.* [1] introduced a tiered text processing methodology that analyzed syntactic and pragmatic structures to decode colloquial internet language. They showed that meticulous preprocessing such as lemmatization and part-of-speech tagging is essential for cleaning social media noise, and that the granularity of the text being analyzed heavily influences model accuracy.

Taking a broader approach, Ozdemir [7] explored whether digital sentiment extracted

from global event databases could predict real-world civil unrest. By utilizing ensemble tree models, the research proved a definitive statistical correlation between online public mood and physical socio-political escalations, confirming that NLP tools can serve as predictive behavioral indicators.

Traditional classifiers have also demonstrated remarkable durability when analyzing geopolitically tense events. Carina *et al.* [8] established that leveraging Support Vector Machines combined with TF-IDF vectorization creates an incredibly efficient framework for managing highly polarized online discourse, as seen during the Israel-Palestine conflict. Similarly, Neogi *et al.* [9] affirmed the reliability of classical boundaries by examining Twitter reactions to the Indian farmers' protests. Their work confirmed that TF-IDF and Bag-of-Words representations are highly effective baselines for modeling South Asian political sentiment.

To train these supervised classifiers, generating accurate ground-truth labels for raw datasets is a critical preliminary step. Anam and Kusnawi [10] evaluated the efficacy of Flair, VADER, and TextBlob for automated sentiment tagging during post-election analysis. In a parallel scenario, Al Maruf *et al.* [11] successfully utilized the VADER lexicon to assign initial sentiment polarities to unlabeled texts related to the Ukraine-Russia conflict, proving the utility of unsupervised lexicon tools when manual human annotation is impossible.

2.2 Deep and Recurrent Neural Architectures

Because sparse matrices cannot understand contextual nuance, researchers increasingly turned to deep learning algorithms that inherently discover structural text features. Mahadevaswamy and Shashirekha [12] noted that bidirectional recurrent models vastly outperform simple frequency-counting techniques because they read text sequences both forward and backward, capturing vital contextual clues that unidirectional systems miss. Minaee *et al.* [13] expanded on this by designing a dual architecture that paired Convolutional Neural Networks (CNNs) with BiLSTMs, allowing the system to detect both short, localized phrases and long, sequential contexts simultaneously.

When comparing deep learning directly to classical models, Hamdi *et al.* [14] concluded that while SVMs remain competitive, Bidirectional LSTM networks especially those utilizing attention layers excel at interpreting deeper semantic meanings in reviews. This conclusion was supported by Rahman *et al.* [15], who demonstrated via a RoBERTa-BiLSTM framework that bidirectional tracking is vital for comprehending texts with complex, long-range dependencies.

2.3 Hybrid Architectures

To leverage the specific advantages of both mathematical and neural methods, recent studies have championed multi-layered hybrid models. Metu *et al.* [16] designed a pipeline that fed dense recurrent embeddings directly into an SVM, proving that utilizing a maximum-margin classifier on neural data creates a far more stable decision boundary than using either model in isolation. Wafa and Saadi [17] verified the strength of this concept on YouTube data by combining topic modeling with deep sentiment scoring to improve classification accuracy on user-generated videos.

Subsequent research continues to endorse this architecture. Jonnala *et al.* [18] utilized a Bi-LSTM layer to generate hidden sequence maps, which were subsequently categorized by a Logistic Regression model. Their findings demonstrated an optimal balance between processing speed and predictive accuracy. Furthermore, Babu *et al.* [19] established that linking Bi-LSTM contextual networks with classical feature extractors like Bag-of-Words yields superior results specifically for chaotic YouTube comment sections, heavily validating the architectural decisions of this dissertation.

Looking closely at political uprising contexts, Hossen *et al.* [20] implemented a hybrid transformer approach to process social media reactions during the 2024 Bangladesh revolution, achieving excellent metrics on multilingual protest data. Das *et al.* [21] provided additional empirical support by demonstrating the strong compatibility and high performance of merging Bi-LSTM features with SVM classifiers in varied linguistic environments.

2.4 Explainability in NLP Systems

With the rise of opaque ”black box” deep learning systems, the necessity for interpretation frameworks has skyrocketed. Bidve *et al.* [22] used LIME (Local Interpretable Model-agnostic Explanations) on complex neural and ensemble networks to reveal hidden biases and validate how text features were being selected before putting the models into production. Thogesan *et al.* [23] later customized SHAP for transformer-based NLP systems, visualizing the exact attribution weights assigned across different network layers, drastically enhancing the transparency of deep models.

In an exhaustive review of the field, Diwali *et al.* [2] mapped the trajectory of explainable AI in sentiment analysis, highlighting the persistent trade-off between a model’s raw predictive power and its innate transparency. Their research cemented the consensus that applying post-hoc interpretation tools is currently the most viable strategy for making advanced deep learning models accountable. Islam *et al.* [3] confirmed that

while hybrid architectures deliver unmatched accuracy, their structural opacity remains a massive hurdle, necessitating external verification tools. Garg *et al.* [4] successfully tackled this by combining deep transformer networks with SHAP and LIME pipelines, proving that interpretability layers directly reduce false-positive rates by identifying flawed linguistic reasoning within the model.

Finally, Mosca *et al.* [24] offered a thorough foundational guide to utilizing SHAP in natural language tasks, explaining exactly how game-theoretic attribution acts as the current industry standard for tracking token-level importance in complex texts.

2.5 Gaps Addressed by This Work

This study aims to explore the impact of automatic sentiment classifiers on the accuracy and generalization ability of classification systems. Current existing research has two unaddressed gaps to fill:
(i) The existing hybrid techniques have been scarcely adapted or experimented upon for use with small, politically motivated, multi-lingual comment corpora commonly seen in South Asian protest movements;
(ii) There is little information about how choosing between automatic annotators (TextBlob or VADER) affects the effectiveness of hybrid neural pipelines; and
(iii) SHAP explainability of BiLSTM-SVM models on socio-political corpora remains unexplored thus far. The present dissertation addresses all three questions at once.

Chapter 3

Machine Learning Models

We used machine learning algorithms in this study as baseline models to compare performance. These machine learning algorithms are still used a lot in sentiment analysis because they do not need a lot of computer power are easy to set up and can handle text that does not have a lot of information. Even though new deep learning models can understand the context of text better the old classifiers are still really good at what they do especially when we are working with datasets or datasets that are specific, to one area like the machine learning algorithms we are talking about.

In this chapter, we will talk about some of the machine learning algorithms that were used during the course of our experiments. Some of the algorithms used were Multinomial Naïve Bayes, Logistic Regression, and Support Vector Machines. We will see the mathematical aspects associated with Multinomial Naïve Bayes, Logistic Regression, and Support Vector Machines. We will also look at the process involved in training of Multinomial Naïve Bayes, Logistic Regression, and Support Vector Machines.

3.1 Multinomial Naive Bayes

Multinomial Naive Bayes is a classification algorithm that uses probability. It is based on Bayes theorem.

The model assumes that features in text are independent. This means that one word does not directly affect another in the document.This assumption is often not true, for natural language tasks. Multinomial Naive Bayes still works well for many text classification problems. It is simple and fast.

If a document is represented by a term-frequency vector $\mathbf{x} = (x_1, x_2, \ldots, x_V)$, where V represents the vocabulary size, the posterior probability of the document belonging to class c is calculated as:

$$P(c \mid \mathbf{x}) = \frac{P(c)\, P(\mathbf{x} \mid c)}{P(\mathbf{x})} \propto P(c) \prod_{i=1}^{V} P(x_i \mid c)^{x_i}, \tag{3.1}$$

One common issue in probabilistic models occurs when previously unseen words appear during testing, causing probability values to become zero. To reduce this problem, Laplace smoothing is applied to the conditional probability term:

$$\hat{P}(x_i \mid c) = \frac{N_{ic} + \alpha}{\sum_{j=1}^{V} (N_{jc} + \alpha)}, \tag{3.2}$$

where N_{ic} denotes the frequency of term i in class c, and α represents the smoothing parameter, typically assigned a value of 1.

In practice the MNB model works with sparse text data and does not need a lot of computer power. However it does not consider the order of words and their context so it is not good at understanding sentiment expressions. In this study the model performed average in the begining. It did not perform well as the other classifiers. For simple text classifiers MNB is still a choice since it is easy to use and it does not require a lot of resources. For more complex tasks other models might be more suitable.

3.2 Logistic Regression

Logistic Regression is a way to figure out what group something belongs to. It is often used for tasks where we need to understand how people feel about something like if they like it or not. Logistic Regression is different from Bayes because it looks at how all the information is connected to the result. It does not assume that each piece of information is separate from the others. The Logistic Regression classifier tries to guess the chances that something belongs to a group using a special math function called the softmax function. Logistic Regression is really good at understanding the relationship, between the information we have and the groups we are trying to predict. For a three-class sentiment classification problem, the probability of class c for input vector $\mathbf{x}$ is defined as:

$$P(y = c \mid \mathbf{x};\, \Theta) = \frac{\exp(\mathbf{w}_c^\top \mathbf{x} + b_c)}{\sum_{k=1}^{K} \exp(\mathbf{w}_k^\top \mathbf{x} + b_k)}, \tag{3.3}$$

where $K = 3$, $\mathbf{w}_c$ represents the weight vector associated with class c, and b_c denotes the bias term. The parameter set $\Theta = \{\mathbf{w}_c, b_c\}_{c=1}^{K}$ is optimized by minimizing the regularized cross-entropy loss:

$$\mathcal{L}(\Theta) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbf{1}[y_n = c]\ \log P(y = c \mid \mathbf{x}_n;\, \Theta) + \frac{\lambda}{2}\|\Theta\|^2, \tag{3.4}$$

where λ controls the ℓ_2 regularization strength and N represents the number of training samples.

Logistic Regression does a job with TF-IDF representations because it can easily tell apart data that has a lot of dimensions but not a lot of information. However Logistic Regression still has a time understanding how words are related to each other in a sequence and in context. In the experiments we did for this research Logistic Regression did better than MNB in some ways. It was not as good as the approach that used SVM. Logistic Regression had some problems so it was not as effective, as the SVM approach. That is why Logistic Regression did not do as well as we had hoped.

3.3 Support Vector Machine

One method of text classification using machine learning is through the use of Support Vector Machine. Separating the text into various classes is the primary function of the Support Vector Machine. The separator here is termed the hyperplane. This separator is used in the Support Vector Machine to ensure that the text is separated as much as possible by maximizing the margin between them. Through such means, the Support Vector Machine attempts to improve its performance when it encounters new texts.

For a binary classification setting with training samples $\{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$, where $y_n \in \{-1, +1\}$, the optimization objective is expressed as:

$$\min_{\mathbf{w},\, b,\, \boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \tag{3.5}$$

$$\text{subject to} \quad y_n(\mathbf{w}^\top \mathbf{x}_n + b) \geq 1 - \xi_n, \quad \xi_n \geq 0, \quad \forall\, n,$$

where $\mathbf{w}$ represents the normal vector to the hyperplane, b represents the bias parameter, and ξ_n are slack variables that allow limited classification errors. The trade off between maximizing and minimizing classification mistakes are controlled by parameter C.

One-Versus-Rest (OVR) strategy was used since three sentiment classification was involved. In this approach, multiple binary classifiers are trained independently, and the class with the highest confidence score is selected during prediction.

Why SVM Performs Well on Text Data

The SVM algorithm proves to be more efficient in dealing with text classification due to the sparsity of feature vectors in text data sets. This approach will ensure that the important terms are considered while the highly frequent but insignificant terms are considered less important. The previous literature suggests that the TF-IDF algorithm works very well in conjunction with an SVM classifier [25].

A further benefit offered by the SVM is that the decision boundary produced remains stable even in a noisy environment. The fact that social media commentary is usually marked by erratic use of language means that stability is very important.

Performance in This Study

The SVM model outperformed the other machine learning models we considered. It performed very well indeed. Using the sentiment labels created by TextBlob, the SVM model alone achieved an accuracy rate of 87%. Most importantly, it performed better than many other deep learning models, which were created from scratch. This proves that the traditional approach to text classification can still perform excellently when solving some specific text classification problems. The SVM model is a maximum margin classifier, and it can still compete with other models. The SVM model performs well in text classification in a domain.

3.4 Model Training Setup

SVM machine learning model turned out to be the best of all considered here. It performed outstandingly well. Using sentiment labels provided by TextBlob the model achieved accuracy of 87 percent. What is even more important is the fact that it exceeded accuracy of some other deep learning approaches. A consistent approach to preparing the machine learning models was followed for fair comparison of results.

The text corpus was preprocessed into numeric features with the help of the TF-IDF vectorization technique.

In order to find an optimal compromise between the computational complexity and predictive accuracy the vocabulary size was restricted to 5,000-10,000 terms.

The data was partitioned into 80% training set. And 20% testing subset through stratified random sampling.

It helped maintain proportionality of number of cases in both subsets.

Tuning of classifiers' parameters included finding the most appropriate combination and cross validation.

Some classifiers required tuning of such parameters as Laplace smoothing for MNB penalty on weights for LR and penalty parameter C for SVM.

LinearSVC classifier implementation was used for better computational efficiency.

3.5 Evaluation Metrics

All model performances were evaluated with typical classification criteria used for sentiment analysis studies.

Accuracy

Accuracy can be defined as the percentage of correct classifications out of total predicted data:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3.6}$$

Precision, Recall, and F1-Score

Precision and recall were computed separately for each sentiment class:

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}, \qquad \text{Recall}_c = \frac{TP_c}{TP_c + FN_c}, \tag{3.7}$$

$$F_1^{(c)} = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \tag{3.8}$$

Precision indicates the number of true predictions made, but recall refers to the number of true samples which were correctly detected. F1 score is obtained from the precision and recall.

Macro and Weighted Averages

The macro-average involves the computation of arithmetic mean among all categories with equal weight assigned to each category, whereas the weighted average takes into account the frequencies of different classes.

Chapter 4

Deep Learning Models

The significance of deep learning in natural language processing can not be stressed too much. Deep learning is different from machine learning, as it involves learning directly from data. For instance, neural networks do not require any kind of assistance during the learning process. Deep learning involves the use of training in order to establish the meaning of words and how the same are connected within the sentences. The learning process is what informs the creation of deep learning models used for determining the sentiment in social media comments. Deep learning is extremely powerful in dealing with problems related to natural language processing.

In this particular scenario, the research is conducted by employing two kinds of deep learning models. These two models include convolutional neural networks and bidirectional long short-term memory networks. This is attributed to the nature in which both of these models conduct themselves with regards to natural language processing. This study is designed in such a way that the two models are compared as far as their efficiency in natural language processing goes. The main purpose of this study is to compare the performance of both convolutional neural networks and bidirectional long short-term memory networks when it comes to sentiment classification.

4.1 Convolutional Neural Network (CNN)

Convolutional neural networks were originally developed to analyze images. However, they are highly effective when used to solve issues relating to text categorization. The use of convolutional neural networks to language-based problems is achieved through linear processing of words. This is done in order to identify any patterns within the texts. Convolutional Neural Networks look at words one, after the other to see what is important.

The model works well in finding phrases or keyword combinations that usually show

if someone is happy or sad. That is the reason why CNN-based models can successfully capture features from social media commentaries that show a lot of emotions. In particular, such a model can detect features like frequently used words or phrases that convey certain emotions. Social media commentaries can be considered a suitable source for this type of task. Such a model is successful in detecting those phrases or keywords that convey some emotional context.

Formal Architecture

Let us say that the tokens in a text sequence are represented using a d-dimensional embedding vector. In that case, an embedding matrix would be formed as $\mathbf{E} \in \mathbb{R}^{T \times d}$. The convolution operation is performed on the embedding matrix using a filter of size h, which can be expressed as $\mathbf{f} \in \mathbb{R}^{h \times d}$ to produce features:

$$c_i = \sigma(\mathbf{f} \cdot \mathbf{E}_{i:i+h-1} + b), \quad i = 1, \ldots, T - h + 1, \tag{4.1}$$

where $\sigma(\cdot)$ denotes the ReLU activation function and b represents the bias term.

After convolution, a global max-pooling operation is applied to the generated feature map in order to retain the most informative signal:

$$\hat{c} = \max_i \ c_i. \tag{4.2}$$

A combination of outputs of all filters is then fed into a dense layer with a softmax function for sentiment classification.

CNN architectures are also limited to considering local neighbourhoods of words. This might be okay since the method will be able to recognize strongly polarized statements. However, it might not fully comprehend the relationship between words in sentences due to lack of consideration for long-term dependencies.

Training Parameters

In order to give stable gradient calculations during the training process, a CNN architecture was trained using batches of size 32. Early stopping was applied in case of validation loss plateaus, and about 6 to 10 epochs were conducted during the training phase.

However, there was a slight drawback to the CNN architecture utilized in the study, in that random initialization of the embedding vector was used instead of huge pre-trained language models, while it is known that CNN architectures usually train

quickly and yield good results. In this regard, comprehension of deep semantic relations in complex political language was hindered by the model.

4.2 Bidirectional Long Short-Term Memory Network (Bi-LSTM)

The Recurrent Neural Networks (RNNs) were invented specifically to analyze sequential data because the RNN remembers previously acquired information as it processes data in each time step. However, the issue that the information gained by the neural network vanishes soon due to vanishing gradients does not allow them to deal with long-term dependency.

To overcome this problem, the Long Short-Term Memory (LSTM) networks were invented. They possess memory cells as well as gates to control their information. Therefore, they can retain important information.

LSTM Cell Equations

The LSTM unit operates on the input vector $\mathbf{x}_t$ at each time step t, incorporating the hidden state $\mathbf{h}_{t-1}$ carried forward from the previous step through the computations outlined below:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \tag{4.3}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \tag{4.4}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \tag{4.5}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \tag{4.6}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \tag{4.7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{4.8}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, and $\mathbf{o}_t$ denote the input, forget, and output gates in that order. The term $\tilde{\mathbf{c}}_t$ represents the candidate cell state produced at the current step, whereas $\mathbf{c}_t$ is the resulting updated memory cell. The symbol $\odot$ denotes the Hadamard (element-wise) product, and $\sigma(\cdot)$ is the sigmoid function applied component-wise.

Bidirectional Extension

By processing the sequence in both forward and backward directions at the same time, a bidirectional LSTM expands upon the conventional LSTM architecture. This helps the model to extract contextual information from the sentence's preceding and succeeding words.

The forward and backward hidden states are computed as:

$$\overrightarrow{\mathbf{h}}_t = \overrightarrow{\text{LSTM}}(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}), \qquad \overleftarrow{\mathbf{h}}_t = \overleftarrow{\text{LSTM}}(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}). \tag{4.9}$$

The final contextual representation for each token is obtained by concatenating both hidden states:

$$\mathbf{h}_t^{\text{bi}} = \left[\overrightarrow{\mathbf{h}}_t \parallel \overleftarrow{\mathbf{h}}_t\right]. \tag{4.10}$$

The network is better able to understand contextual dependencies, such as negations, sentence reversals, and more intricate linguistic patterns that are frequently seen in social media conversations, thanks to this bidirectional structure.

Training Method

An embedding layer at the start of the BiLSTM model transforms input tokens into dense numerical vectors. In order to capture contextual interactions from both directions of the sequence, these embeddings are then processed by bidirectional recurrent layers.

To lessen overfitting and enhance generalization performance, dropout regularization was used during training. Lastly, probabilities for the three emotion categories were produced using a dense layer with softmax activation.

The BiLSTM outperformed the CNN model on longer and more context-dependent remarks. When digesting politically complex or emotionally complex statements, its capacity to store sequential knowledge was helpful.

4.3 Comparison of Deep Learning Approaches

In their application, both DL networks exhibited some advantages and disadvantages. CNN networks managed to detect short term local sentiment trends and were resource efficient. On the other hand, they lacked the capacity to understand longer-term dependencies and more complicated phrases due to their use of mainly localized kernels.

As opposed to CNNs, the BiLSTM model performed better at understanding more

complicated sentence structures and opinions involving several clauses as it maintained context throughout the whole text sequence. This enabled an improvement in the classification accuracy of politically charged statements where meanings were indirectly altered via changes to the context.

The training phase for the BiLSTM network took longer than for the CNN network due to its more extensive context-learning ability. The experiments also showed that among all individual non-hybrid models used, traditional linear SVM still maintained the highest accuracy.At the end it is suggested that conventional machine learning approaches are highly effective for relatively small data sets.

Chapter 5

Methodology

The entire methodology of the research including collecting of the data, preprocessing of the data, training the models, and finally analyzing the explainability aspects is discussed in detail in this chapter. In fact, this process included several stages, such as collecting comments on YouTube, standardizing and cleansing text, labeling sentiments, training baseline models, developing the proposed hybrid approach, and conducting SHAP analysis of model predictions.

This aim of this process was to create a reliable sentiment analysis tool capable of coping with noisy social media discussions about protests in Nepal during 2025.

5.1 Data Collection

YouTube Data API version 3 was used to obtain the dataset. In order to search for relevant videos, a list of protest keywords related to 2025 Nepal protests was used. Following the identification of appropriate videos, the relevant video IDs were extracted and utilized to automatically retrieve user comments.

Video identifiers, anonymous user identifiers, and comment text were all included in the gathered dataset. For additional preparation and analysis, all extracted data was saved in CSV format. Personally identifying information other than generic platform IDs was removed from the dataset in order to uphold ethical norms and user privacy.

The final dataset had 7,754 unique comments after duplicate and incorrect entries were removed.

5.2 Data Cleaning

Comments on social media are typically very unstructured and include multilingual content, symbols, abbreviations, and uneven writing styles. As a result, a number of

preprocessing steps were taken prior to model training.

Lowercasing

To ensure consistency across the dataset, all text was changed to lowercase. As a result, the models were unable to treat different capitalizations of the same word as distinct tokens.

For example: “Good”, “GOOD”, and “good” were all converted to “good”.

Removal of URLs and Special Characters

Hyperlinks, hashtags, punctuation, and other symbols are common in online comments, however they don’t really help with sentiment assessment. URLs, superfluous punctuation, and non-alphanumeric characters were eliminated using regular expression (Regex) operations.

Emojis were not completely discarded since they frequently convey emotional significance. A specialized Python package was used to translate emoji symbols into textual descriptions while maintaining their sentiment information.

Handling Non-English Text

Language identification was used as part of the process to verify that only submissions written in the English language were selected since some comments were made in other languages. Limiting the dataset to English only submissions made it easier to maintain compatibility with the chosen NLP packages and embedding methods.

Negation Handling

Negation words can drastically change the meaning of a phrase. Even if the sentence has positive words negation words can make the sentiment negative.

To remove this problem ,during preprocessing, negation words were combined with nearby words to solve this problem:

“not good” $\rightarrow$ “not_good”

This strategy allowed the models to better preserve the intended sentiment polarity without requiring deeper grammatical parsing.

Lemmatization

Lemmatization was done in order to get to the root form of the words and avoid unnecessary vocabulary inflation.

For instance: “running”, “runs”, and “ran” were stemmed into “run”.

Tokenization

After the data was cleaned, each comment was then tokenized into separate tokens in order to process the text using machine learning and deep learning algorithms.

“The protest was not peaceful.” $\rightarrow$ [the, protest, was, not, peaceful]

Depending on the architecture used for the model, the tokenized text can either be converted to TF-IDF vectors or to integer sequences.

Sentiment Labelling

Since there were no pre-assigned labels to the sentiment classes for the gathered YouTube comments, the automated methods were used to create labels. The two popular sentiment lexicons that have been tested independently are VADER and TextBlob, which assign labels to the comments in terms of positive, negative, or neutral classes.

The first sentiment lexicon VADER was especially created for texts in social media and understands the strength of punctuation, capital letters usage, and slang terms. On the contrary, the TextBlob uses a linguistically based technique to determine the polarity and subjectivity of a text.

Table 5.1: Label Distribution Under Each Lexicon (7754 comments)

Lexicon	**Negative**	**Neutral**	**Positive**
Textblob	1906	3394	2454
Vader	2721	2523	2510

It was noted that TextBlob allocated a higher percentage of comments to the neutral class than did VADER. Since neutral comments can be easy to categorize, models trained using TextBlob labels tend to attain better accuracy.

5.3 Feature Engineering

Upon preprocessing, the textual data was converted to numeric data for use in machine learning and deep learning models.

TF–IDF Representation

Machine learning models traditionally require numerical vectors as features and not actual text data. This is why TF-IDF vectorization was used for representing comments in a numerical way.

TF-IDF boosts relevant words' significance while decreasing insignificant common words' contribution.

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D), \tag{5.1}$$

where $\text{TF}(t, d) = f_{t,d} / \sum_{t' \in d} f_{t',d}$ and $\text{IDF}(t, D) = \log(|D|/(1 + |\{d \in D : t \in d\}|))$.

The generated TF-IDF vectors were used as input for MNB, Logistic Regression, and SVM classifiers.

Word Embeddings

As deep learning models rely on sequential numeric data instead of sparse vector representations, the tokenized words were encoded as integer values via a vocabulary mapper implemented through a tokenizer.

In order to make sure that the dimensionality of each sequence was equal, we decided to pad each sequence to a fixed length $L = 200$. The resulting sequences were then embedded into dense vectors representing semantics of the words..

5.4 Preprocessing Pipeline

The preprocessing process was done in a sequence where, first, incomplete and erroneous records were excluded from the data set. Next, non-English comments were also eliminated in order to make sure that the data was linguistically consistent.

Afterward, text normalization procedures such as lowercasing, symbol removal, negation handling, and lemmatization were applied. Finally, the cleaned comments were converted into either TF-IDF vectors or padded token sequences depending on the requirements of the target model.

Consistency has been improved across all experiments using this standerdised preprocessing.

5.5 Models and Experimental Procedure

Traing and evaluation of the models were done using 80/20 split. To maintain the distribution of sentiment classes across both subsets stratified sampling was used.

5.5.1 Baseline Machine Learning Models

The following traditional machine learning algorithms were evaluated:

- **Multinomial Naive Bayes (MNB):** The probabilistic classifier that relies on the distribution of words to determine sentiment categories. It does not take much computing time, but has no notion of context.
- **Logistic Regression (LR):** The linear classifier that can provide probability values by using the TF-IDF feature vector.
- **Support Vector Machine (SVM):** A maximum-margin classifier that showed excellent results with sparse text representation. LinearSVC was used for scalable computing.

5.5.2 Baseline Deep Learning Models

The following neural network architectures were tested:

- **CNN:** Uses convolutional filters to identify localized sentiment patterns and important phrases within comments.
- **Bi-LSTM:** Analyses textual sequences in both forward and reverse directions, allowing better contextual analysis. The model was trained with 64 hidden units, a dropout rate of 0.5, and for 10 epochs.

5.5.3 Proposed Hybrid BiLSTM-SVM Architecture

Feature Extraction using BiLSTM

The suggested architecture is based on a combination of feature extraction from the context through a BiLSTM network followed by classification using an SVM algorithm. In this work, the BiLSTM network has not been used for making any prediction directly but to extract meaningful features.

The workflow of the proposed architecture is summarized below:

- **Input preparation:** The preprocessed comments were padded up to 200 tokens long. The preprocessed data was tokenized by an algorithm into integers depending on the order of their occurrence within the vocabulary.
- **Embedding layer:** The input tokens are represented in 256 dimensions and have a maximum vocabulary of 20,000 words.
- **Contextual feature extraction:** A bidirectional LSTM layer with 512 units processed the embedding sequences. Global Average Pooling and Global Max Pooling layers were applied to capture complementary contextual information.
- **Dense feature generation:** The pooled outputs were concatenated and passed through a dense layer containing 256 neurons with ReLU activation.
- **Feature normalization:** The resulting vectors were standardized using Z-score normalization before being forwarded to the SVM classifier.

Previous studies have shown that combining recurrent neural networks with optimized feature representations can improve sentiment classification performance and enhance evaluation metrics such as recall and F1-score.

SVM Classification

Finally, the vectors from the BiLSTM network served as input vectors to the final SVM classifier, which ensured maximization of the separation margin among sentiments along with minimization of classification errors.

Due to the fact that there was an element of class imbalance in the sentiment labels provided by VADER and TextBlob, weighted class penalty was applied during the training process of SVM classifier.

Hyperparameter tuning was performed via the grid search method coupled with 5-fold cross validation. The C hyperparameter varied over the following values $\{0.01, 0.1, 1, 10, 100\}$ for both linear and RBF kernel classifiers. For optimization purposes, the macro-averaged F1-score measure was chosen as the evaluation criterion.

5.5.4 Model Explainability Using SHAP

SHAP was added to the proposed approach in order to increase its interpretability. SHAP utilizes game theory and computes the importance of individual variables in making predictions for a particular model.

The SHAP value for feature i is calculated as:

$$\phi_i(f, \mathbf{x}) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \big[f(S \cup \{i\}) - f(S) \big], \tag{5.2}$$

where F represents the complete feature set and $f(S)$ denotes the model prediction obtained using the subset of features S.

It was through the use of SHAP that the most significant contributing words for positive, negative, or neutral sentiment were identified. The transparency in the model's decision-making process was enhanced as a result of this approach.

Chapter 6

Results and Discussion

The results that were acquired through different types of learning approaches such as machine learning, deep learning, and hybrid learning approaches have been discussed in this chapter. The performance of these models was evaluated based on certain standards such as accuracy, precision, recall, and F1 score. In addition to comparing the scores, the performance of these models was compared so as to determine the pros and cons of each model when it comes to performing the comment analysis task.

Further discussion will focus on the impact of different sentiment labeling methods on the models.

6.1 Machine Learning Results

The conventional ML models used for the experiment were trained with the help of the TF-IDF feature vectors derived from the dataset. Three baselines were used to train the ML models namely; MNB, LR, and SVM.

Naive Bayes Performance

Multinomial Naive Bayes was used as the baseline algorithm in the experiment as this algorithm is considered one of the most simple models. The model's probabilistic nature enabled it to handle words with evident sentiments quite well. However, dealing with sarcastic and negative comments became problematic for this model.

These results show that model worked somewhat better using the TextBlob labeled training data. Using the VADER labeled set of data, the accuracy attained by the model was roughly 68%. The recall rate was relatively higher compared to the precision rate since there were some instances of overfitting for the negative class.

Using the TextBlob labeled set of data, accuracy improved to about 72%. The model exhibited better performance overall, but it could not interpret certain complex

	precision	recall	f1-score	support
negative	0.61	0.80	0.69	250
neutral	0.79	0.49	0.60	239
positive	0.69	0.74	0.72	245
accuracy			0.68	734
macro avg	0.70	0.67	0.67	734
weighted avg	0.70	0.68	0.67	734

Figure 6.1: VADER sentiment

	precision	recall	f1-score	support
negative	0.94	0.28	0.43	164
neutral	0.70	0.90	0.79	330
positive	0.70	0.76	0.73	241
accuracy			0.72	735
macro avg	0.78	0.65	0.65	735
weighted avg	0.75	0.72	0.69	735

Figure 6.2: TextBlob sentiment

sentences due to its lack of contextual knowledge. Nonetheless, model was an important base line model due to its efficiency and cost effectiveness.

Logistic Regression Performance

More stability and balance in the results was found in Logistic Regression than in Naive Bayes. This is due to the efficiency of the model in dealing with high-dimensional and sparse vectors in TF-IDF representation.

	precision	recall	f1-score	support
negative	0.69	0.74	0.72	250
neutral	0.70	0.76	0.73	239
positive	0.82	0.69	0.75	245
accuracy			0.73	734
macro avg	0.74	0.73	0.73	734
weighted avg	0.74	0.73	0.73	734

Figure 6.3: VADER sentiment

	precision	recall	f1-score	support
negative	0.87	0.48	0.61	164
neutral	0.71	0.97	0.82	329
positive	0.83	0.69	0.75	241
accuracy			0.77	734
macro avg	0.81	0.71	0.73	734
weighted avg	0.79	0.77	0.75	734

Figure 6.4: TextBlob sentiment

In VADER sentiments, Logistic Regression yielded around 73% accuracy. There was consistency in the performance between all three categories, which implies that the generalization was better compared to MNB.

On TextBlob-labelled sentences, the model attained an approximate accuracy of 77%. Significant improvements were observed in precision for positive and negative sentiments, while the ability to identify neutral statements helped in achieving good macro average results.

Even though Logistic Regression performed well, it was not able to handle sequential relationships within text.

SVM Performance

Out of all the classic machine learning algorithms, Support Vector Machine turned out to be the most efficient one. This is because of the capability of the algorithm that it could effectively manage sparse TF-IDF by maximizing the margin between sentiments while being impervious to noise in texts.

	precision	recall	f1-score	support
negative	0.77	0.76	0.77	250
neutral	0.73	0.80	0.76	239
positive	0.84	0.76	0.80	245
accuracy			0.78	734
macro avg	0.78	0.78	0.78	734
weighted avg	0.78	0.78	0.78	734

Figure 6.5: VADER sentiment

	precision	recall	f1-score	support
negative	0.88	0.69	0.77	164
neutral	0.86	0.96	0.91	330
positive	0.87	0.85	0.86	241
accuracy			0.87	735
macro avg	0.87	0.83	0.85	735
weighted avg	0.87	0.87	0.86	735

Figure 6.6: TextBlob sentiment

Accuracy scores close to 78% for VADER labeling and close to 87% for TextBlob labeling, achieved by SVM classifier, became the best-performing standalone baseline method in this experiment.

High precision and recall scores as well as high F1-scores were found in each of the three sentiment classes. In particular, SVM performed exceptionally well for identifying positive and neutral sentiment samples.

The ratio of the macro and weighted measures indicates that SVM performed quite adequately considering class imbalance and domain specific vocabulary.

This suggests that traditional linear methods still remain competitive when solving sentiment analysis tasks on medium size data sets.

6.2 Deep Learning Results

In this case, unlike the traditional algorithms, the deep learning algorithms applied to the padded tokens rather than using the vectors of TF-IDF values.

There were two models of neural networks that were used in this analysis: CNN and BiLSTM.

CNN Performance

The Convolutional Neural Network provided adequate to high accuracy when analyzing both labeled datasets. The algorithm was efficient in identifying sentiment clusters and emotionally loaded phrases.

	precision	recall	f1-score	support
negative	0.76	0.71	0.74	250
neutral	0.78	0.85	0.82	239
positive	0.76	0.75	0.75	245
accuracy			0.77	734
macro avg	0.77	0.77	0.77	734
weighted avg	0.77	0.77	0.77	734

Figure 6.7: VADER sentiment

	precision	recall	f1-score	support
negative	0.81	0.71	0.75	164
neutral	0.89	0.90	0.89	330
positive	0.78	0.82	0.80	241
accuracy			0.83	735
macro avg	0.82	0.81	0.82	735
weighted avg	0.83	0.83	0.83	735

Figure 6.8: TextBlob sentiment

Accuracy rates were observed to range between roughly 77% and 83%, based on how the data was annotated by the annotator.

On the other hand, when there were longer statements that needed interpretation in a much wider context, the CNN sometimes failed due to its reliance on convolutional filters that concentrated primarily on smaller groups of words.

Nonetheless, despite the limitations of the CNN, it still performed better than several traditional baselines.

Bi-LSTM Performance

The performance of the BiLSTM model was marginally better than that of the CNN model in terms of contextual understanding. This was attributed to the fact that the neural network was able to process language in two directions.

	precision	recall	f1-score	support
negative	0.84	0.65	0.73	164
neutral	0.89	0.88	0.88	329
positive	0.74	0.87	0.80	241
accuracy			0.82	734
macro avg	0.82	0.80	0.80	734
weighted avg	0.83	0.82	0.82	734

Figure 6.9: VADER sentiment

	precision	recall	f1-score	support
negative	0.85	0.66	0.75	164
neutral	0.90	0.86	0.88	329
positive	0.75	0.90	0.82	241
accuracy			0.83	734
macro avg	0.83	0.81	0.81	734
weighted avg	0.84	0.83	0.83	734

Figure 6.10: TextBlob sentiment

BiLSTM model had an accuracy score of almost 82% on data labeled using VADER and around 83% on TextBlob labelled data.

Performance was exceptionally good when it came to detecting neutral or context based comments. There were no major differences among the F1-scores for the different categories of sentiments, which indicates that the model has consistent sequential learning abilities.

Even though BiLSTM took more time to train than the CNN model, it performed better because of its strong capacity to capture contextual information.

6.3 Baseline Model Performance

It can be seen from the comparison that classification accuracy performed by TextBlob labelings was relatively higher compared to others. This might be due to the larger number of neutral texts identified by TextBlob, thus eliminating ambiguities during classification.

For single classification methods, the highest accuracy was obtained by SVM. On the other hand, BiLSTM exhibited improved context-awareness and developed meaningful

Model	Accuracy (VADER)	Accuracy (TextBlob)
Multinomial Naive Bayes	68%	72%
Logistic Regression	73%	77%
Support Vector Machine (SVM)	78%	87%
CNN	77%	83%
Bi-LSTM	82%	83%

Table 6.1: Performance comparison of ML and DL models

semantic representations. Considering this, BiLSTM was employed as the feature extraction model in the hybrid approach.

6.4 Hybrid Model Performance

The proposed BiLSTM-SVM hybrid architecture was evaluated using the same experimental setup as the baseline models.

```
Best Params: {'C': 10, 'gamma': 'scale', 'kernel': 'linear'}
Accuracy: 0.8774983881366866
              precision    recall  f1-score   support

    negative       0.78      0.88      0.83       381
     neutral       0.94      0.92      0.93       679
    positive       0.88      0.82      0.85       491

    accuracy                           0.88      1551
   macro avg       0.87      0.87      0.87      1551
weighted avg       0.88      0.88      0.88      1551
```

Figure 6.11: Hybrid model performance

The best performance was exhibited by the hybrid model when compared to other techniques. The hybridization of features extraction through the BiLSTM and the decision boundary optimization technique of the SVM enabled the model to outperform other individual systems.

The hybrid model was found to be more stable in terms of sentiments and could better handle social media noises.

In order to gain a better understanding of the predictions, the following confusion matrix was created for the hybrid architecture.

As per the results obtained, excellent classification was achieved for all three types of sentiments. For neutral sentiment, the recall score was recorded as about 0.92, implying that the proposed hybrid model detected neutral emotions quite efficiently.

For negative sentiment, recall was recorded as about 0.88, where softer versions of negative sentiment were sometimes classified as neutral. On the other hand, positive

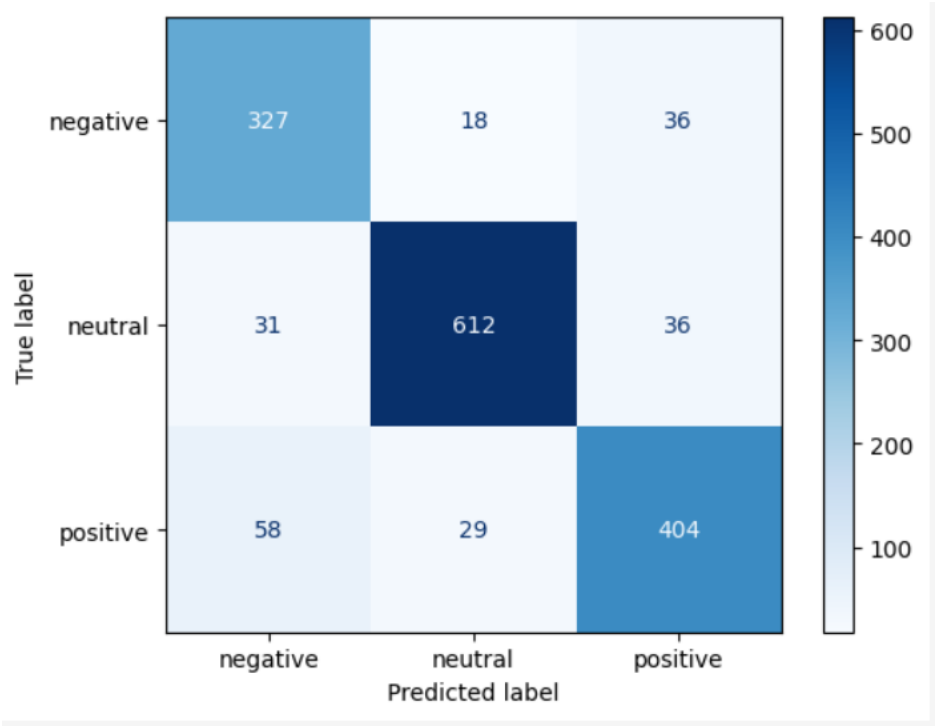


Figure 6.12: Confusion matrix of the hybrid model

recall was recorded as about 0.82, and at times, some of the moderately positive sentiment was sometimes classified as neutral.

Hence, the above confusion matrix clearly depicts that the proposed hybrid approach was capable of predicting efficiently, even in politically polarized settings.

6.4.1 SHAP Analysis

For better interpretation and understanding of the behaviour of the models, SHAP analysis was used in the hybrid approach. As a result of computation constraints, SHAP values were calculated using samples of 200 comments.

From the results obtained, it was clear that many words had an impact on sentiments. Negative sentiments were greatly affected by words such as “corrupt”, “wrong”, “destroy”, and “violent”. This is in response to public dissatisfaction with institutional and political concerns.

On the other hand, words such as “good”, “love”, “strong”, and “peaceful” had positive impacts on the sentiment prediction of supportive or optimistic sentiments. Words like “political” entities, “location”, and discussion terms generally formed neutral sentiments.

Moreover, the results obtained from SHAP analysis further add value to the understanding

Table 6.2: Top SHAP-Identified Words Per Sentiment Class

Rank	Negative (score)	Neutral (score)	Positive (score)
1	corrupt (13.78)	oli (0.46)	social (8.19)
2	wrong (2.83)	another (0.39)	good (5.89)
3	destroy (2.07)	like (0.37)	first (3.82)
4	bad (1.94)	watch (0.36)	love (3.12)
5	military (1.75)	not_do (0.28)	right (3.08)
6	mean (1.28)	party (0.28)	many (3.00)
7	dead (1.25)	violence (0.26)	lol (2.53)
8	violent (1.21)	not_them (0.24)	old (2.50)
9	population (1.19)	well (0.18)	free (2.13)
10	crazy (1.18)	intellectual (0.17)	strong (2.01)
11	hard (1.12)	belong (0.16)	kind (1.97)
12	foreign (1.09)	manipur (0.16)	ready (1.86)
13	evil (0.96)	befor (0.16)	peaceful (1.57)
14	brutal (0.96)	north (0.14)	wild (1.53)
15	fuck (0.96)	ittrade_marks (0.14)	top (1.17)
16	leftist (0.94)	want (0.14)	young (1.08)
17	arrest (0.94)	chinese (0.14)	cool (1.04)
18	not_wrong (0.92)	burn (0.13)	elect (1.02)
19	stupid (0.92)	korea (0.12)	proud (0.98)
20	serious (0.88)	india (0.12)	exactly (0.97)

of the nature of political discourse in the period under investigation. Whereas positive sentiments referred to issues such as reform, unity, and civic activism, negative sentiments included corruption, violence, and political instability.

```
=== OVERALL DOMINANT SENTIMENT: 'POSITIVE' ===

Impact Breakdown by Sentiment:
  - negative: 67.6779
  - neutral: 6.8342
 -> POSITIVE: 79.8321 (DOMINANT)
```

Figure 6.13: Dominant sentiment distribution

In general, the findings indicate that positive and reformist sentiments were prevalent in the online debate. Nevertheless, the prevalence of negative sentiments reveals that there was considerable dissatisfaction and political tension during the protests.

Chapter 7

Conclusion

This thesis sought to investigate the public opinion reflected in the YouTube comments concerning the 2025 Nepal protests. The aim was to identify the performances of various machine learning and deep learning algorithms in processing the political and unstructured nature of social media data. Moreover, it sought to ascertain the extent to which the integration of deep contextual learning with classical classifiers enhances performance and interpretability.

7.1 Summary of Work

The research started with gathering YouTube comments through the use of YouTube Data API. As online discussions can include inconsistencies in the format, multi-lingual text, and other types of messy text, several preprocessing methods were used before model development. Among those, lowercasing, removal of URLs and symbols, negations, lemmatization, and tokenization were used.

Following the preprocessing step, textual data were represented in numeric form through TF-IDF vectors and sequential embeddings. Next, several baselines were implemented for comparison with the proposed method. The traditional machine learning techniques considered included the Multinomial Naive Bayes, Logistic Regression, and Support Vector Machine models. Furthermore, CNN and BiLSTM were developed through deep learning.

In our proposed approach, we have merged the process of extracting the features from the data with the help of contextual information using the BiLSTM model with the prediction of sentiments using the SVM model.

To improve the transparency of the model, an analysis module for SHAP was incorporated, enabling the extraction of significant words used during the process of predicting sentiments.

The performance measures used while evaluating the results of different models include accuracy, precision, recall, and F1 score.

7.2 Key Findings

Several important findings emerged from the experiments conducted in this study:

- **Outstanding performance of SVM:** In the classical machine learning algorithms, the SVM had a very high performance rate, especially when used with the TextBlob data labeling process.
- **Contextual benefit of BiLSTM:** The performance of the BiLSTM was higher compared to that of the CNN since it could handle context-dependent comments, negations, and political statements in general.
- **Benefits of the proposed architecture:** Integrating the BiLSTM for feature extraction and SVM for classification helped enhance the prediction process through both learning contextual features and improving the decision boundary.
- **Effects of data annotations:** Models built using TextBlob labels performed better than models built using VADER labels.
- **Better interpretability using SHAP:** The SHAP helped to recognize which factors played an essential role regarding each of the sentiment classes. Namely, negative sentiments were usually related to the presence of corruption, whereas positive sentiments were characterized by unity and reforms.
- **Mixed public sentiment during the protests:** Sentiment analysis revealed both the presence of optimistic opinions and feelings of frustration among internet users. Namely, while many comments displayed optimism about reforms and citizen engagement, a great number of comments also displayed dissatisfaction with political organizations and institutions.

Overall, the findings demonstrate that hybrid NLP architectures can provide effective sentiment classification performance while also offering greater interpretability for politically sensitive social media analysis.

7.3 Limitations

Even though many achievements have been observed due to the employment of the proposed model, several limitations have been identified.

The first limitation is that all the data analyzed in this study have been provided in the English language. Therefore, the majority of information written in other languages has not been analyzed. The next problem is associated with the complexity of identifying and comprehending irony, sarcasm, and subtle expressions of emotions. While the use of context has allowed the improvement in terms of this issue, it is still challenging for some models.

Concerning the annotations that were provided for this task, it should be said that the labeling was fully automated. In other words, no human labor involved, and the comments were labeled via the use of two automated techniques: VADER and TextBlob.

Lastly, the conducted experiments have been carried out using a specialized dataset related only to one particular political event.

7.4 Future Work

The following avenues are worth exploring in future work for enhancing the effectiveness of the proposed technique even further:

- **Multilingual Sentiment Analysis:** Future research can incorporate multilingual transformer models that would allow the processing of Nepali and other regional languages along with English. This would give better insights into the general public sentiment surrounding politically sensitive topics from a linguistic point of view [26].

- **Aspect-Based Sentiment Analysis (ABSA):** Rather than assigning sentiments to whole comments, future techniques could analyze sentiment towards certain aspects of discussion such as government policy, actions of law enforcement officers, and economic problems.

- **Integration with Pre-trained Language Models:** More advanced transformer architectures like BERT and RoBERTa would likely outperform RNNs in terms of context-awareness and sarcasm detection tasks.

- **Real-Time Sentiment Monitoring System:** Finally, the proposed hybrid model framework can be extended into a real-time sentiment dashboard for monitoring live social media streams during political or social events.

Overall, it is concluded that the use of deep contextual features alongside classic machine learning algorithms gives good accuracy levels while still allowing researchers to make sense of their data.

Bibliography

[1] M. Ranjan, A. Kumar, R. Sharma, and P. Singh, "A new approach for carrying out sentiment analysis of social media comments using natural language processing," *Engineering Proceedings*, vol. 62, no. 1, pp. 44–51, 2024.

[2] A. Diwali, B. Patel, and C. Verma, "Sentiment analysis meets explainable artificial intelligence: A survey," *IEEE Transactions on Affective Computing*, vol. 15, no. 3, pp. 580–595, 2024.

[3] M. S. Islam, M. F. Hossen, and M. R. Islam, "Challenges and future in deep learning for sentiment analysis," *Artificial Intelligence Review*, vol. 57, no. 2, pp. 889–912, 2024.

[4] P. Garg, A. Sharma, and N. Kumar, "Improving hate speech classification through ensemble learning and explainable ai techniques," *Arabian Journal for Science and Engineering*, vol. 50, no. 1, pp. 1245–1259, 2025.

[5] T. T. Prama, C. M. Danforth, and P. S. Dodds, "Story and essential meaning dynamics in bangladesh's july 2024 student-people's uprising," *arXiv preprint arXiv:2511.01865*, 2025.

[6] R. K. Singh and A. Thomas, "A systematic literature review of youtube comments sentiment analysis: Challenges and emerging trends," *ICTACT Journal on Data Science and Machine Learning*, vol. 7, no. 1, pp. 947–961, 2025.

[7] U. Ozdemir, "Predicting social unrest using sentiment analysis," Master's thesis, Tilburg University, Netherlands, 2018.

[8] F. M. Carina, A. Salma, D. Permana, and Z. Martha, "Sentiment analysis of x application users on the conflict between israel and palestine using support vector machine algorithm," *UNP Journal of Statistics and Data Science*, vol. 2, no. 2, pp. 204–212, 2024.

[9] 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, p. 100019, 2021.

[10] K. Anam and Kusnawi, "Comparison of sentiment labeling using textblob, vader, and flair in public opinion analysis post-2024 presidential inauguration with indobert," *Jurnal Teknik Informatika (JUTIF)*, vol. 6, no. 2, pp. 803–818, 2025.

[11] A. Al Maruf, Z. M. Ziyad, M. M. Haque, and F. Khanam, "Emotion detection from text and sentiment analysis of ukraine russia war using machine learning technique," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 12, 2022.

[12] U. B. Mahadevaswamy and S. Shashirekha, "Sentiment analysis using bidirectional lstm network," *Procedia Computer Science*, vol. 218, pp. 2341–2349, 2023.

[13] S. Minaee, A. Azimi, and A. Abdolrashidi, "Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models," *arXiv preprint arXiv:1904.04218*, 2019.

[14] R. H. Hamdi *et al.*, "Benchmarking logistic regression, svm, and lightgbm against bilstm with attention for sentiment analysis on indonesian product reviews," *arXiv preprint arXiv:2604.25452*, 2026.

[15] M. M. Rahman, A. I. Shiplu, Y. Watanobe, and M. A. Alam, "Roberta-bilstm: A context-aware hybrid model for sentiment analysis," *arXiv preprint arXiv:2406.00367*, 2025.

[16] M. A. Metu, O. C. Asogwa, C. O. Nwoye, and A. N. Ikechukwu, "Hybrid svm-bidirectional long short-term memory model," *Journal of Advances in Information Technology*, vol. 15, no. 2, pp. 110–118, 2024.

[17] S. Wafa and A. Saadi, "Hybrid nlp framework for enhanced sentiment analysis and topic detection on youtube," *Journal of Information Systems and Engineering Management*, vol. 10, no. 1, pp. 63–72, 2025.

[18] N. S. Jonnala *et al.*, "Leveraging hybrid model for accurate sentiment analysis of twitter data," *Scientific Reports*, vol. 15, p. 24438, 2025.

[19] B. R. Babu, S. Ramakrishna, and S. K. Duvvuri, "Hybrid nlp framework for enhanced sentiment analysis and topic detection on youtube," *International Journal of Basic and Applied Sciences*, vol. 14, no. 1, pp. 304–313, 2025.

[20] M. S. Hossen, M. Saiduzzaman, and P. Shaha, "Social media sentiments analysis on the july revolution in bangladesh: A hybrid transformer based machine learning approach," *arXiv preprint arXiv:2507.11084*, 2025.

[21] R. K. Das, M. Islam, M. M. Hasan, S. Razia, M. Hassan, and S. A. Khushbu, "Sentiment analysis in multilingual context: Comparative analysis of machine learning and hybrid deep learning models," *Heliyon*, vol. 9, p. e20281, 2023.

[22] V. Bidve, P. Sarkar, and M. Joshi, "Use of explainable ai to interpret the results of nlp models for sentimental analysis," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 2, pp. 1820–1829, 2024.

[23] T. Thogesan, R. Krishnamurthy, and S. Balasubramanian, "Integration of explainable ai techniques with large language models for sentiment analysis," *IEEE Access*, vol. 13, pp. 1104–1115, 2025.

[24] E. Mosca, F. Szigeti, S. Tragianni, D. Gallagher, and G. Groh, "Shap-based explanation methods: A review for nlp interpretability," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022.

[25] M. Das, S. Kamalanathan, and P. Alphonse, "A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset," *CEUR Workshop Proceedings*, 2023.

[26] P. Gupta, "A breadth-first catalog of text processing, speech processing and multimodal research in south asian languages," *arXiv preprint arXiv:2501.00029*, 2024.