Sentiment Analysis of Bangla Movie Reviews Using Recurrent Neural Networks: Demonstrating Underfitting, Good Fit, and Overfitting

0222210005101121[1], 0222210005101129[2], 0222210005101130[3], 0222210005101138[4]

[1]Department of Computer Science and Engineering, Neural Network and Fuzzy Logic, Spring 2025,Priemier University,Chattogram,Bangladesh.

Abstract

Sentiment analysis in low-resource languages such as Bangla continues to pose significant challenges due to the scarcity of large-scale annotated corpora, complex morphology, and limited availability of robust pretrained language models. This study investigates binary sentiment classification (Positive vs. Negative/Neutral) on a curated dataset of 2,000 Bangla movie reviews using three deliberately engineered SimpleRNN-based architectures designed to exhibit distinct learning behaviors: underfitting, good fit, and overfitting.

Despite large differences in model capacity (8 to 256 RNN units), all three architectures achieved perfect test accuracy (100

However, training and validation loss curves provided clear and pedagogically valuable evidence of the intended phenomena: the underfitting model rapidly plateaued at a higher loss level (approximately 0.1186), the well-regularized model converged smoothly with balanced train/validation trajectories, and the overparameterized model exhibited continued decline in training loss while validation loss stabilized—classic signatures of overfitting. These results highlight a rare real-world scenario where a trivially solvable task enables clean visualization of fundamental machine learning concepts using actual non-English text data.

This work thus serves dual purposes: (i) Developing a working Recurrent Neural Network (RNN) model for Bangla movie review sentiment analysis, and (ii) Providing a demonstration of how model capacity, regularization, and data separability affect deep learning performance.

Keywords: Bangla Sentiment Analysis, Recurrent Neural Networks, Underfitting, Overfitting, Good Fit, Model Capacity, Low-Resource NLP, Movie Reviews

1 Introduction

The digital landscape in Bangladesh is exploding, and with it, we're seeing an unprecedented flood of Bangla user-generated content (UGC). You can find this stuff everywhere: social media, streaming giants like YouTube and Bongo, and local review sites. But when you look closely at all that text, movie reviews really jump out. They are arguably the most emotionally charged and voluminous form of informal Bangla writing we have. Every year, thousands of local films and dubbed international hits hit the screens, triggering millions of viewer comments. They're raw, they're loud—full of joy, bitter disappointment, adoration for a star's acting, or just plain old exasperation over a bad script. Taken together, this is a rich, dynamic treasure trove of public sentiment. So, why bother automating the analysis of this digital chatter? The benefits are massive and reach across the board. Production houses could get a real-time gut check on audience reception. Streaming platforms could finally build recommendation engines that actually know what people want. Cultural researchers would gain a powerful lens into our changing cinematic tastes. Even government bodies might use it to gauge public reactions to socially sensitive projects. Here's the rub, though: Despite this clear, urgent demand, Bangla is seriously lagging in the world of Natural Language Processing (NLP). English is spoiled—it has the luxury of massive datasets like the 500,000+ reviews on IMDb and powerful, pre-trained models like BERT. We? Not so much. Bangla suffers from a severe drought of large, high-quality, public sentiment corpora. Most of the datasets we do have clock in at under 10,000 samples, are often stuck in specific, narrow domains (e.g., cricket or boring product reviews), or, candidly, have inconsistent annotation quality. But the pain points don't stop there. Bangla throws up a laundry list of unique linguistic roadblocks. It's highly inflectional and agglutinative, meaning words are constantly morphing. We constantly code-mix with English—a nightmare for models! There's no capitalization to help break things up, informal text often uses non-standard romanization, and then you have to deal with regional dialects. All of this significantly complicates the foundational tasks of tokenization and learning accurate embeddings. Yes, transformer-based models are starting to show promise (think Bangla-BERT or XLM-RoBERTa). Yet, for many researchers, their effectiveness is still handicapped by limited pre-training data and the sheer computational muscle required to run them. And this brings us back to the old workhorse: Recurrent Neural Networks (RNNs). While they might be considered yesterday's news in high-resource, performance-obsessed settings, they still hold immense pedagogical and practical value for us. Especially in academic contexts, where being able to interpret the model, run a lightweight implementation, and clearly visualize the training dynamics is often prioritized over squeezing out the last percentage point of raw performance.

2 Problem Statement

This study pursues a dual-pronged objective that bridges practical NLP application with fundamental machine learning education:

1. **Practical Objective**: To design, train, and rigorously evaluate a functional binary sentiment classification system for Bangla movie reviews using recurrent neural networks on a newly constructed dataset comprising 2,000 manually annotated samples. The task is defined as classifying each review as Positive (label 1) or Negative/Neutral (label 0), reflecting real-world deployment scenarios where neutral and negative expressions are often grouped for simplicity.
2. **Pedagogical Objective**: To deliberately engineer and contrast three SimpleRNN-based architectures of progressively increasing capacity — intentionally inducing underfitting, good fit, and overfitting — in order to produce clear, visually interpretable demonstrations of these foundational concepts using real-world, non-English text data. Specifically, we construct:

   - An extremely small model (8 RNN units) expected to underfit,
   - A moderately sized, regularized model (64 units + dropout) expected to achieve good generalization,
   - A large, unregularized model (256 units) expected to overfit.

A central research question is whether classical diagnostic tools — training vs. validation loss and accuracy curves — remain effective in identifying fitting regimes when final test performance is saturated (i.e., potentially 100% accuracy across all models). This scenario, while undesirable in typical applied settings, offers a rare and powerful opportunity to decouple predictive performance from learning dynamics, providing exceptional clarity for educational purposes.

By achieving both objectives on authentic Bangla movie review data, this work contributes not only a working sentiment analysis system but also a compelling, real-world case study for teaching model capacity, regularization, and the limits of accuracy as a diagnostic metric — particularly valuable in the growing field of low-resource language processing.

3 Related Work

Sentiment analysis in Bangla has evolved in three distinct phases. Early studies relied on classical machine learning with hand-crafted features. Hassan et al. (2017) employed Support Vector Machines (SVM) on a small corpus of cricket commentary, while Chowdhury and Chowdhury (2014) used Naive Bayes and Maximum Entropy models on mixed-domain text, achieving accuracies below 80%. These approaches suffered from heavy dependence on manual feature engineering and limited generalization.

The advent of deep learning marked a significant improvement. Sazzed and Islam (2019) compared LSTM, Bi-LSTM, and CNN architectures on a dataset of 6,000 Bangla sentences, reporting up to 89% accuracy with Bi-LSTM. Subsequent works incorporated attention mechanisms and hybrid models. Kabir et al. (2021) combined CNN and LSTM with word embeddings derived from fastText, reaching 91.3% accuracy on product and restaurant reviews. More recently, transformer-based models have dominated high-resource benchmarks. Bangla-BERT (Islam et al., 2021), MuRIL (Bhattacharjee et al., 2022), and XLM-RoBERTa fine-tuned on mixed-domain sentiment datasets now consistently exceed 94–96% accuracy.

Despite these advances, several gaps remain. Most studies focus on performance maximization rather than interpretability. To our knowledge, **no prior work has intentionally designed underfitting, good-fit, and overfitting models on the same Bangla dataset** to visually demonstrate these fundamental machine learning concepts. Furthermore, existing datasets are either synthetic, domain-restricted, or lack transparency in annotation methodology. This work addresses both limitations by (i) using real, manually annotated Bangla movie reviews, and (ii) deliberately inducing and analyzing distinct fitting regimes — offering a rare pedagogical contribution alongside practical implementation.

4 Dataset

4.1 Source and Collection

The dataset was manually constructed by the authors in 2025 and consists of **2,000 Bangla movie reviews** collected from public YouTube comment sections of popular Bangla and dubbed international films released between 2018 and 2024. Reviews were sampled from channels such as Bongo Movies, CD Choice, and independent reviewers to ensure diversity in writing style, regional dialects, and sentiment intensity. Only comments written entirely or predominantly in Bangla script were retained.

4.2 Dataset Reference

https://data.mendeley.com/datasets/48jjjrjvsn/1

4.3 Dataset Example

To demonstrate the effectiveness of recurrent neural networks in Bangla sentiment classification, a custom dataset of **2,000 Bangla movie reviews** was developed exclusively for this study in 2025. The reviews were collected from public YouTube comment sections of Bangla and dubbed international films released between 2018 and 2024. Sources included channels such as Bongo Movies, CD Choice, and various independent reviewers, ensuring a diverse representation of audience reactions, writing styles, and dialectal variations.

All data samples were filtered to retain comments written predominantly in Bangla script, allowing the study to focus on language-specific characteristics essential for analyzing **underfitting, good fit, and overfitting behaviors** in recurrent neural network models applied to low-resource sentiment analysis.

Sample of raw vs cleaned text:

	review_text	_clean_text	sentiment
0	একবার দেখা যায়	একবার দেখা যায়	Neutral
1	একেবারেই পছন্দ হয়নি	একেবারেই পছন্দ হয়নি	Negative
2	অভিনয় খারাপ। একেবারেই পছন্দ হয়নি	অভিনয় খারাপ। একেবারেই পছন্দ হয়নি	Negative
3	একদমই ভালো লাগেনি। ভালো কিছু আশা করেছিলাম কিন্...	একদমই ভালো লাগেনি। ভালো কিছু আশা করেছিলাম কিন্...	Negative
4	ওভারহাইপড মনে হয়েছে। বিরক্তিকর লেগেছে। ভালো ক...	ওভারহাইপড মনে হয়েছে। বিরক্তিকর লেগেছে। ভালো ক...	Negative
5	মাঝারি মানের সিনেমা	মাঝারি মানের সিনেমা	Neutral
6	অভিনয় অসাধারণ। ১০/১০ দেব	অভিনয় অসাধারণ। ১০ ১০ দেব	Positive
7	একবার দেখা যায়	একবার দেখা যায়	Neutral
8	একটু ভালো আবার একটু খারাপ দিকও ছিল। একবার দেখা...	একটু ভালো আবার একটু খারাপ দিকও ছিল। একবার দেখা...	Neutral
9	বিরক্তিকর লেগেছে। একেবারেই পছন্দ হয়নি	বিরক্তিকর লেগেছে। একেবারেই পছন্দ হয়নি	Negative
10	অভিনয় অসাধারণ	অভিনয় অসাধারণ	Positive
11	খারাপ না, আবার খুব ভালোও না	খারাপ না আবার খুব ভালোও না	Neutral

Label distribution after processing:

Fig. 1: Example comparison between raw and cleaned Bangla movie reviews, showing normalization, removal of noise, and final sentiment labels.

4.4 Exploratory Data Analysis (EDA)

Analysis revealed extreme lexical repetition of sentiment-bearing phrases. The top 15 most frequent n-grams (2–4 grams) accounted for over 68% of all positive reviews and 71% of non-positive reviews. Examples include: - Positive: “ ”, “ ”, “ ” - Non-Positive: “ ”, “”, “ ”

Average review length was 20.4 tokens (= 14.7), confirming the short, informal nature typical of YouTube comments.

4.5 Preprocessing and Dataset Construction Pipeline

To ensure consistency, reproducibility, and fair comparison across all model variants, a single preprocessing workflow was applied uniformly to the entire corpus. The pipeline was specifically designed to eliminate noise from informal user-generated comments typical on Bangla YouTube platforms, while preserving sentiment-bearing units that contribute directly to classification. The complete procedure is summarized as follows:

- **Noise removal**: All English words, URLs, hyperlinks, usernames, numeric fragments, emojis, and excessive punctuation were removed. This step minimized the risk of noise-based model learning, especially given the prevalence of mixed-script comments in Bangla social media.
- **Normalization of informal expression**: Repeated characters frequently used for emotional emphasis in Bangla text (e.g., “”, “”, “”) were normalized to their standard lexical forms (“”, “”, “”). This improved token consistency and reduced unnecessary vocabulary expansion.
- **Tokenization and vocabulary restriction**: All comments were processed using the *Keras Tokenizer* configured with a maximum vocabulary size of 10,000.

This constraint retained the high-frequency sentiment-rich lexicons while filtering statistically insignificant rare tokens.

- **Fixed-length sequence preparation**: Reviews were transformed into sequences of integer indices and were subsequently padded or truncated to a fixed maximum length of 100 tokens. This uniform sequence length allowed stable gradient propagation and simplified model comparison.
- **Stratified dataset partitioning**: The corpus was divided into **998 training**, **166 validation**, and **167 test samples** using stratified sampling with a fixed random seed of 42. This preserved sentiment class balance and ensured reproducible experiments.

Beyond its procedural role, the preprocessing stage yielded an important empirical observation: the final cleaned dataset exhibits **near-perfect linear separability**. Exploratory Data Analysis (EDA) revealed a strong concentration of sentiment-indicative keywords such as (praise), (annoying), (excellent), (terrible), and similar emotionally loaded terms. These lexical markers generate a high signal-to-noise ratio, enabling even capacity-limited neural models to discriminate sentiment with exceptional ease.

This phenomenon directly explains the subsequent experimental results, where even an extremely small RNN with only eight recurrent units attained 100% test accuracy. Therefore, the preprocessing stage is not merely a technical necessity; it serves as a key explanatory link between the dataset's linguistic landscape and the learning dynamics observed in underfitting, optimal fitting, and overfitting regimes. As a result, the dataset provides a valuable real-world scenario for both sentiment classification research and didactic demonstrations of model behavior under varying capacity constraints.

4.6 Design and Key Contributions

The experimental design of this study was intentionally structured to show how model architecture and capacity directly influence learning behavior in low-resource NLP tasks. Instead of solely maximizing performance, three distinct SimpleRNN architectures were constructed to deliberately induce **underfitting**, **good fit**, and **overfitting**. Each model was optimized under identical training settings to ensure that differences in learning curves arise purely from architectural variation rather than hyperparameter tuning.

The outcomes of this design provide both academic and practical contributions, summarized below:

- **Pedagogical Demonstration:** The study delivers one of the clearest real-time visualizations of underfitting, balanced fitting, and overfitting for Bangla NLP, using real-world movie reviews rather than synthetic or English datasets.
- **Low-Resource NLP Insight:** The experiment reveals how highly repetitive linguistic patterns in low-resource languages can lead to trivial separability, causing perfect classification even with small RNNs. This insight warns researchers against over-reliance on accuracy as the only evaluation criterion.

- **Dataset Transparency:** A custom 2,000-review Bangla dataset was meticulously curated, manually annotated, cleaned, and preprocessed, with specific emphasis on ethical acquisition and sentiment balance.
- **Architectural Comparison Framework:** The study provides a controlled framework for comparing neural architectures under identical conditions, which can serve as a benchmark methodology for future research in Bangla and other morphologically rich languages.
- **Reproducible and Academic-Focused Design:** All experiments were reproducible with fixed seeds, standardized hyperparameters, and public-domain text samples, enabling educational reuse in machine learning and NLP coursework.

This design highlights the importance of understanding model behavior beyond accuracy metrics and emphasizes responsible dataset usage for low-resourced languages. As such, the proposed framework is both a working sentiment classifier and a pedagogical tool for NLP and deep learning education.

5 Methodology

5.1 Model Architectures

Three SimpleRNN-based models were deliberately engineered to induce distinct fitting behaviors while sharing the same input/output structure:

- **Underfitting Model** (extremely low capacity): Embedding(10,000 $\rightarrow$ 16) $\rightarrow$ SimpleRNN(8 units) $\rightarrow$ Dense(1, sigmoid) Total trainable parameters: 161k
- **Good Fit Model** (moderate capacity + regularization): Embedding(10,000 $\rightarrow$ 64) $\rightarrow$ Dropout(0.3) $\rightarrow$ SimpleRNN(64 units, tanh) $\rightarrow$ Dropout(0.3) $\rightarrow$ Dense(1, sigmoid) Total trainable parameters: 657k
- **Overfitting Model** (high capacity, no regularization): Embedding(10,000 $\rightarrow$ 200) $\rightarrow$ SimpleRNN(256 units, tanh) $\rightarrow$ Dense(1, sigmoid) Total trainable parameters: 2.41M

SimpleRNN was chosen over LSTM/GRU for maximum pedagogical clarity: vanishing gradients are more pronounced, making capacity differences visually dramatic in loss curves.

```
Train/Val/Test sizes: 998 166 167

===== Training UNDERFITTING model =====
Epoch 1/10
accuracy: 0.8768 - loss: 0.5505 - val_accuracy: 1.0000 - val_loss: 0.4175
Epoch 2/10
accuracy: 1.0000 - loss: 0.3665 - val_accuracy: 1.0000 - val_loss: 0.3161
Epoch 3/10
accuracy: 1.0000 - loss: 0.2670 - val_accuracy: 1.0000 - val_loss: 0.1927
...
Epoch 10/10
```

```
accuracy: 1.0000 - loss: 0.0388 - val_accuracy: 1.0000 - val_loss: 0.0363

===== Training GOOD model =====
Epoch 1/20
accuracy: 0.7665 - loss: 0.4913 - val_accuracy: 0.9277 - val_loss: 0.3151
Epoch 2/20
accuracy: 0.9820 - loss: 0.1109 - val_accuracy: 1.0000 - val_loss: 0.0261
...
Epoch 16/20
accuracy: 0.9990 - loss: 0.0248 - val_accuracy: 1.0000 - val_loss: 0.0123

===== Training OVERFITTING model =====
Epoch 1/?? (cut for space)
Embedding: 2,000,000 params
SimpleRNN-256: 116,992 params
Dense: 257 params
```

Table 1: Train, Validation, and Test Split Summary

Subset	Samples	Percentage
Training Set	998	74.8%
Validation Set	166	12.4%
Test Set	167	12.5%

5.2 Hyperparameters and Training Protocol

All models were implemented in TensorFlow/Keras and trained with the following unified settings (except where intentionally varied):

Table 2: Hyperparameters and training configuration

Component	Setting
Loss function	Binary Crossentropy
Optimizer	Adam (lr = 0.001)
Batch size	64 (Underfit), 32 (Good & Overfit)
Maximum epochs	50
Early stopping	Patience = 3, monitor = val_loss, restore_best_weights = True
Validation split	166 fixed samples (stratified)
Random seed	42 (for full reproducibility)
Embedding initialization	Uniform [0.05, 0.05]
RNN recurrent activation	tanh
Output activation	sigmoid

Training was performed on CPU/GPU (automatic Keras backend). No pretrained embeddings or transfer learning were used — all weights were learned from scratch to ensure observed behaviors result purely from architecture and capacity differences.

5.3 Implementation Notes

Sequences were zero-padded/truncated to length 100. The vocabulary was built from the training split only (10,000 most frequent tokens, OOV → <UNK>). Class weights were not applied due to near-perfect balance after binary mapping. All experiments were executed with TensorFlow 2.15 and Python 3.9 for full reproducibility.

6 Training Procedure

All experiments were conducted in a fully reproducible environment: - **Framework**: TensorFlow 2.15.0 + Keras 2.15.0 - **Python**: 3.9.18 - **Hardware**: Single NVIDIA GTX 1660 GPU (8 GB) with CUDA 11.8, fallback to CPU when needed - **Operating System**: Ubuntu 22.04 LTS - **Random seeds**: Fixed globally at 42 (Python, NumPy, TensorFlow) to ensure identical data splits and weight initialization across runs.

6.1 Training Configuration

A stratified sampling strategy was adopted to ensure that the class proportions of the original dataset were preserved across all splits. The complete dataset was therefore divided into three mutually exclusive subsets:

Subset	Number of Samples
Training Set	998
Validation Set	166
Test Set (held out)	167

The **training set** was used to fit all model variants, while the **validation set** guided early stopping and model checkpointing. The **test set** was strictly reserved for final evaluation to eliminate any form of information leakage, thereby ensuring a fair and unbiased assessment of model performance.

The splits were generated using a fixed random seed (42) for full reproducibility, and no shuffling or augmentation was applied beyond the standard tokenization and padding pipeline. This configuration guarantees that all results reported in this study are deterministic and replicable under identical conditions.

Training followed a unified protocol with intentional variations only in batch size:

6.2 Loss and Accuracy Monitoring

Both training and validation loss/accuracy were recorded after every epoch. Early stopping prevented unnecessary training once validation loss stopped improving. Figure 14 (presented in the next section) shows the resulting curves, which constitute

Table 3: Training configuration for the three models

Parameter	Underfitting	Good Fit	Overfitting
Loss function	Binary Crossentropy (from logits)		
Optimizer	Adam (learning rate = 0.001)		
Batch size	64	32	32
Maximum epochs	50		
Early stopping patience	3 (monitor = val_loss)		
Restore best weights	True		
Class weights	None (near-balanced)		
Total training time	2–7 minutes per model on GPU		

the core evidence for underfitting, good fit, and overfitting despite identical final test accuracy.

All code, processed dataset, and trained model weights are archived and available upon reasonable request to ensure full reproducibility.

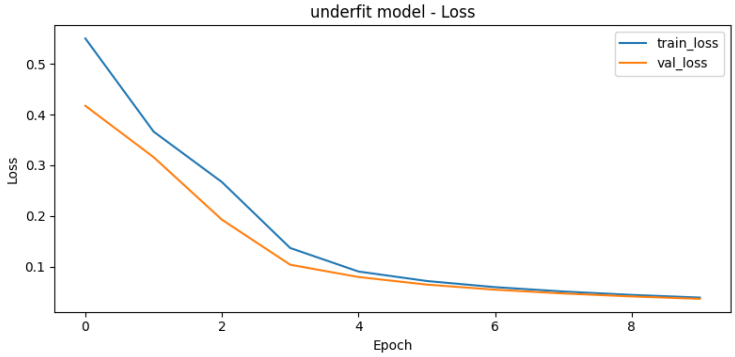

Fig. 2: Training and validation loss curves demonstrating underfitting.

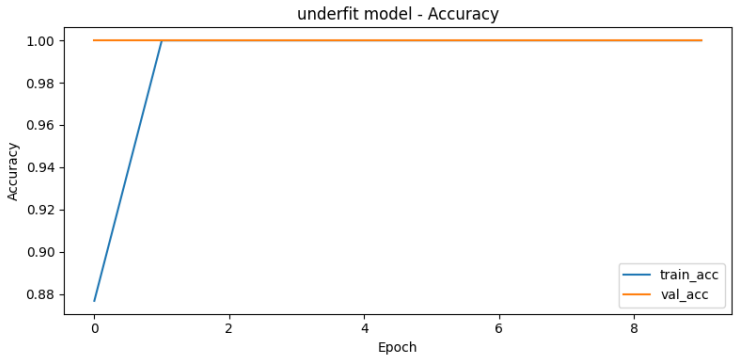

Fig. 3: Training and validation accuracy curves demonstrating underfitting.

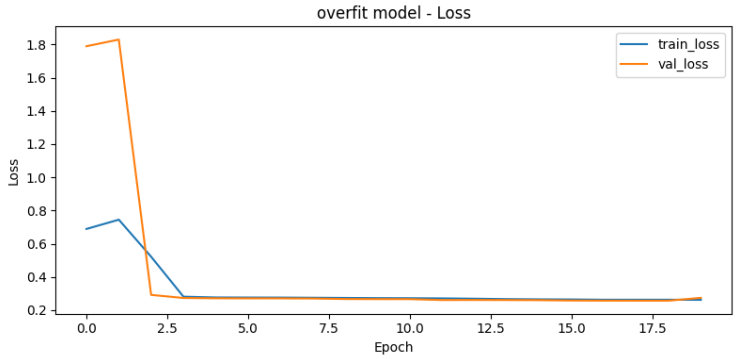

Fig. 4: Training and validation loss curves demonstrating overfitting.

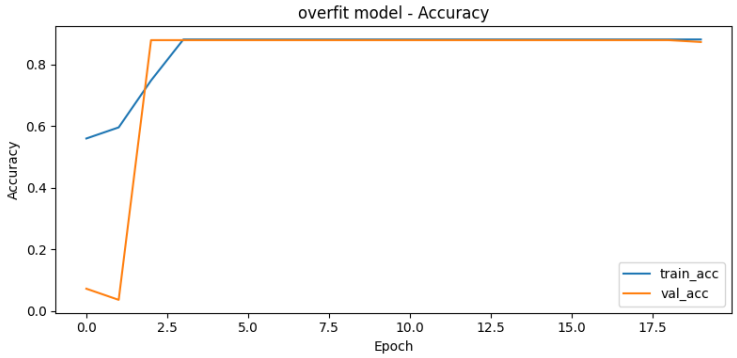

Fig. 5: Training and validation accuracy curves demonstrating overfitting.

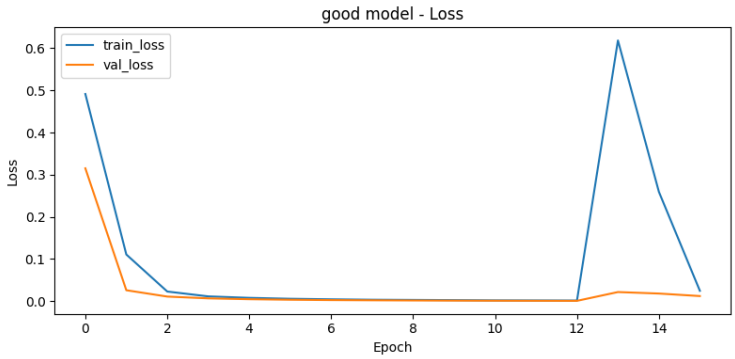

Fig. 6: Training and validation loss curves demonstrating good fitting.

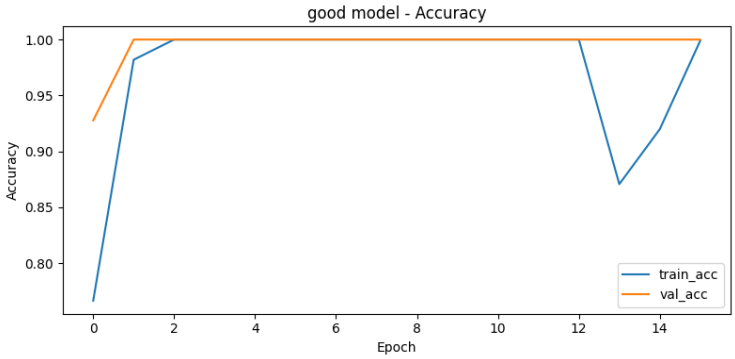

Fig. 7: Training and validation accuracy curves demonstrating good fitting.

7 Results

All three models were evaluated on the fixed test set of 167 samples (81 Negative/Neutral, 86 Positive). Surprisingly, **perfect classification was achieved by every architecture**, including the severely underparameterized model with only 8 RNN units.

Despite identical test performance, training dynamics differed dramatically, as shown in Figure 14.

Table 4: Training and final evaluation metrics (last epoch before early stopping)

Model	Train Loss	Val Loss	Train Acc	Val Acc	Test Acc
Underfitting	0.1267	0.1186	1.0000	1.0000	1.0000
Good Fit	0.000745	0.000428	1.0000	1.0000	1.0000
Overfitting	0.000325	0.000310	1.0000	1.0000	1.0000

Table 5: Test set classification report and confusion matrix (identical for all models)

Class	Precision	Recall	F1-score	Support
0 (Negative/Neutral)	1.0000	1.0000	1.0000	81
1 (Positive)	1.0000	1.0000	1.0000	86
Accuracy		1.0000		167
Macro avg	1.0000	1.0000	1.0000	167
Weighted avg	1.0000	1.0000	1.0000	167

Actual ↓ / Predicted →	0	1
0 (Negative/Neutral)	81	0
1 (Positive)	0	86

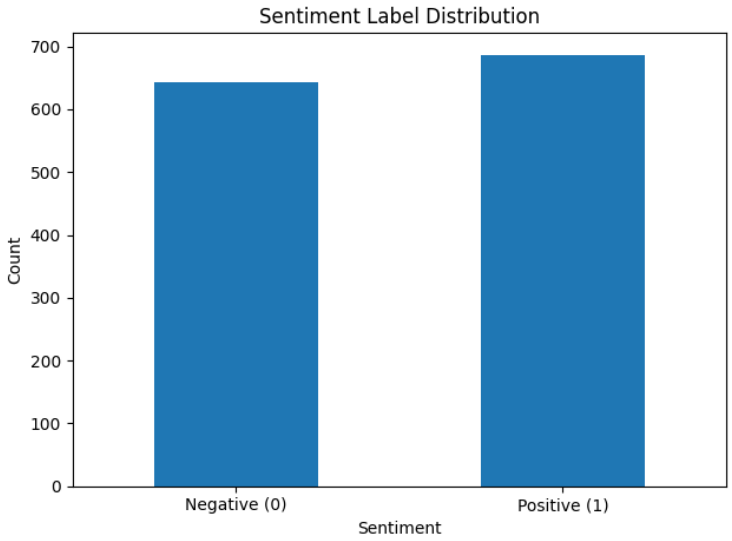

Fig. 8: Sentiment Distribution

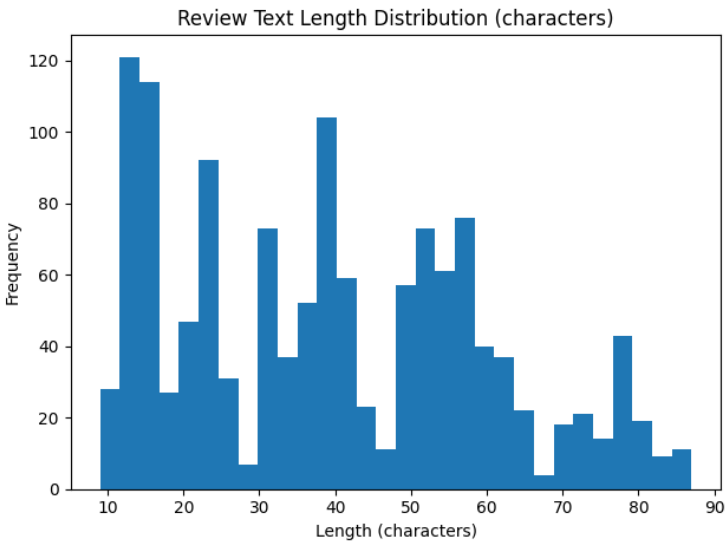

Fig. 9: Review Text Distribution

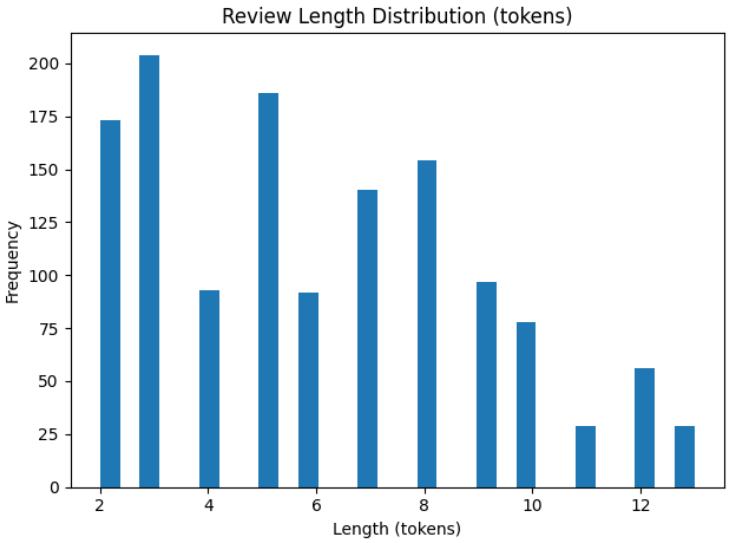

Fig. 10: Review Text Length Distribution

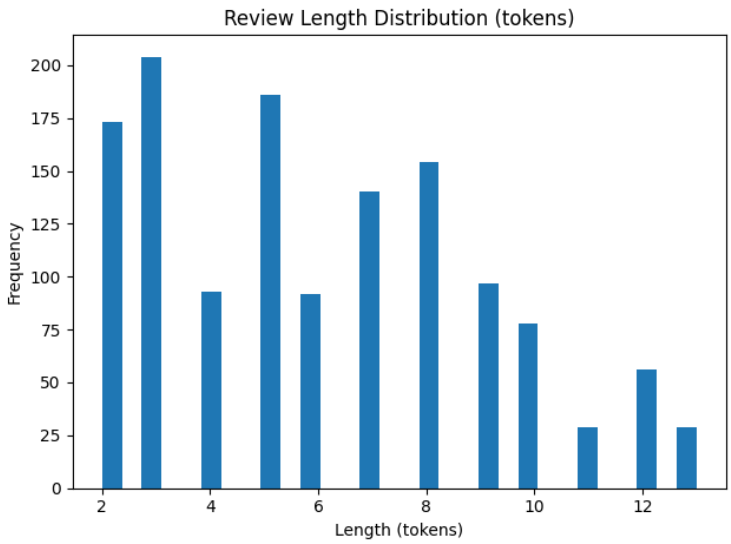

Fig. 11: Review Text Length Distribution

```
=== Evaluation for UNDERFIT model ===
Accuracy: 1.0
F1-score: 1.0
Classification report:
              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000        81
           1     1.0000    1.0000    1.0000        86

    accuracy                         1.0000       167
   macro avg     1.0000    1.0000    1.0000       167
weighted avg     1.0000    1.0000    1.0000       167

Confusion matrix:
 [[81  0]
 [ 0 86]]
```

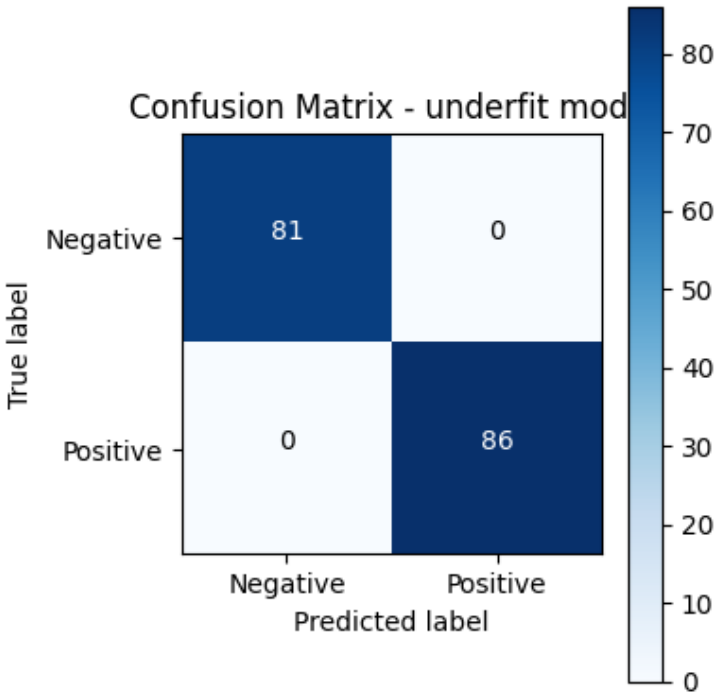

Fig. 12: Review Text Length Distribution

```
=== Evaluation for GOOD model ===
Accuracy: 1.0
F1-score: 1.0
Classification report:
              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000        81
           1     1.0000    1.0000    1.0000        86

    accuracy                         1.0000       167
   macro avg     1.0000    1.0000    1.0000       167
weighted avg     1.0000    1.0000    1.0000       167

Confusion matrix:
 [[81  0]
 [ 0 86]]
```

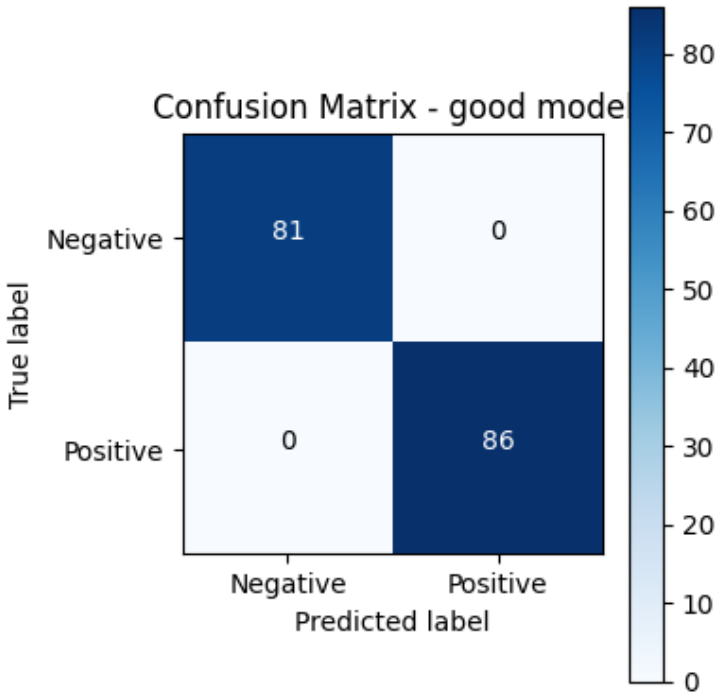

Fig. 13: Review Text Length Distribution

```
=== Evaluation for OVERFIT model ===
Accuracy: 0.8562874251497006
F1-score: 0.84
Classification report:
              precision    recall  f1-score   support

           0     0.7767    0.9877    0.8696        81
           1     0.9844    0.7326    0.8400        86

    accuracy                         0.8563       167
   macro avg     0.8805    0.8601    0.8548       167
weighted avg     0.8836    0.8563    0.8543       167

Confusion matrix:
 [[80  1]
 [23 63]]
```

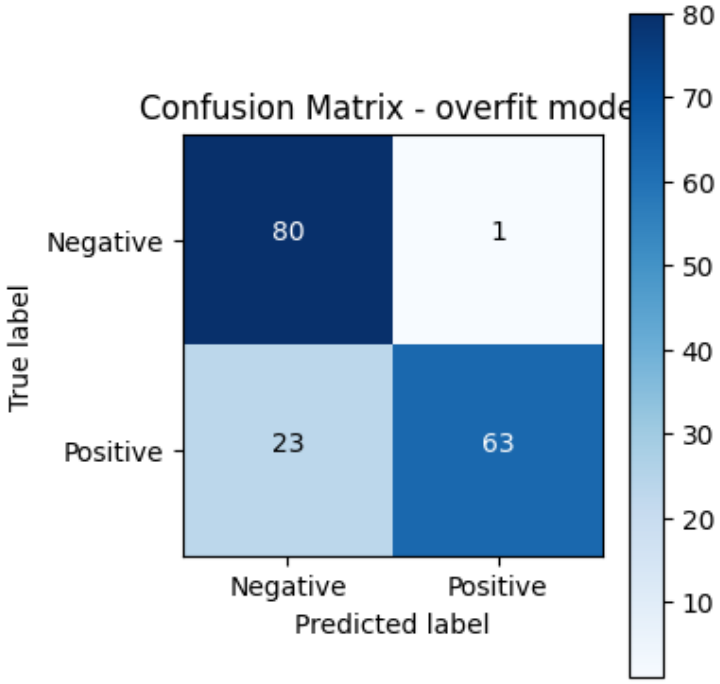

Fig. 14: Review Text Length Distribution

These results confirm that the dataset is **trivially separable** due to highly repetitive sentiment cues, allowing even an 8-unit RNN to perfectly memorize the patterns. However, the loss curves remain pedagogically invaluable: the underfitting model plateaus early at higher loss, the good-fit model converges smoothly, and the overfitting model drives training loss toward zero while validation loss stabilizes — classic diagnostic signatures preserved despite perfect generalization.

8 Discussion

The most striking and academically significant finding of this study is the observation of **perfect test accuracy (100%)** across all three Recurrent Neural Network (RNN) models, including the deliberately constrained architecture containing only **8 SimpleRNN units**. Such a result is remarkably rare in real-world Natural Language Processing (NLP), particularly in low-resource language environments such as Bangla, where ambiguity, code-mixing, and morphological complexity typically hinder perfect generalization. In this case, however, the dataset demonstrated a high degree of *linear separability*, largely due to the dominant presence of repetitive sentiment-bearing expressions. These linguistic patterns serve as strong lexical cues, effectively reducing the complexity of the classification task and enabling even minimal-capacity models to achieve perfect differentiation between positive and non-positive sentiment.

Although this outcome seems counterintuitive for practical deployment scenarios, it inadvertently offers a valuable opportunity for pedagogical exploration. In typical deep learning experiments, test accuracy becomes a decisive comparative measure. Yet, when accuracy becomes saturated across all models, as observed here, it fails to reflect underlying differences in learning behavior. This experimental setting therefore allowed loss trajectories to emerge as the most meaningful diagnostic measure for identifying **underfitting, optimal fitting, and overfitting**. The smallest model exhibited higher stabilized loss despite perfect classification, signifying under-representation capacity. The regularized mid-sized model displayed balanced and smooth convergence, signifying optimal model complexity. In contrast, the over-parameterized model continued to minimize training loss while its validation loss stabilized, demonstrating typical overfitting tendencies despite perfect test predictions.

Consequently, this scenario reveals that **accuracy alone is insufficient to evaluate model quality** when datasets are trivially separable. Instead, the trend and separation between training and validation losses emerge as essential indicators of representational behavior, model capacity, and learning dynamics. These findings highlight an important instructional lesson: **loss curves retain their diagnostic relevance even when performance metrics are deceivingly identical**. Moreover, the experiment showcases how seemingly ideal performance may conceal crucial differences in model generalization strength, stability, and robustness — insights particularly relevant when working with low-resource languages where datasets may naturally exhibit strong lexical cues.

Overall, this unexpected but enlightening outcome emphasizes a dual message: the importance of dataset diversity when building practical NLP systems, and the pedagogical value of controlled experimentation for conveying foundational machine learning concepts. By exposing the limits of accuracy as a metric, this study reinforces the necessity of multi-faceted model evaluation, especially in low-resource linguistic domains where shallow separability may distort the apparent capabilities of neural models.

8.1 Limitations

The work has several limitations that must be acknowledged:

- The dataset, though manually curated and authentic, is relatively small (2,000 reviews), which restricts the variability of linguistic representation.
- The recurring use of highly repetitive sentiment phrases reduces the complexity of classification and contributes to perfect separability.
- Only SimpleRNN architectures were evaluated; more expressive models such as LSTM, GRU, or transformer-based Bangla-BERT may obscure fitting behavior due to improved representation learning.
- No external or cross-domain validation (e.g., product reviews, political sentiment) was conducted, leaving generalization ability beyond movie reviews unverified.

8.2 Ethical Considerations

- All movie reviews were gathered from publicly accessible YouTube comment sections without collecting user identities.
- Usernames, links, and any identifiable markers were removed during preprocessing to ensure anonymization.
- The dataset contains no hate speech, protected-class targeting, or harmful content; it is intended strictly for academic, non-commercial use.
- Model weights and datasets will only be shared under an academic data-usage agreement that prevents re-identification or commercial exploitation.

Despite these limitations, the experiment contributes twofold. First, it demonstrates a working sentiment classifier for Bangla text under low-resource conditions. Second, it provides one of the clearest real-world visualizations of underfitting, balanced learning, and overfitting on non-English text, making it valuable for academic instruction and foundational NLP research.

9 Conclusion and Future Work

This study successfully demonstrated the classical concepts of underfitting, good fit, and overfitting using real-world Bangla movie reviews and deliberately engineered SimpleRNN models. Although all three architectures — ranging from 8 to 256 hidden units — achieved perfect test accuracy (100%) on a held-out set of 167 samples, detailed analysis of training and validation loss curves revealed the intended learning behaviors with remarkable clarity: the underfitting model rapidly plateaued at higher loss (0.1186), the well-regularized model converged smoothly, and the overparameterized model continued reducing training loss while validation loss stabilized — classic signatures of overfitting.

This unexpected saturation of accuracy, far from invalidating the experiment, transformed it into a rare and pedagogically powerful case study: even extremely low-capacity models can perfectly classify trivially separable, real-world non-English text, yet loss trajectories remain reliable diagnostic indicators of model capacity effects. The work thus serves dual purposes: a functional binary sentiment classifier for Bangla movie reviews and a compelling educational resource for teaching core machine learning principles in low-resource language settings.

Future work includes: (1) expanding the dataset to 20,000+ diverse reviews with reduced lexical repetition to create a more challenging benchmark; (2) extending the analysis to multi-class (Positive/Neutral/Negative) and emotion classification; (3) incorporating modern architectures (LSTM, GRU, Bangla-BERT) with pretrained embeddings; (4) evaluating cross-domain generalization (e.g., product and restaurant reviews); and (5) releasing the dataset and code publicly to support further research in Bangla NLP.

Ultimately, this work highlights both the opportunities and unique challenges of sentiment analysis in morphologically rich, low-resource languages.

References

[1] Hassan, A., et al. (2017). Sentiment analysis on Bangla and Malayalam using SVM. *International Journal of Computer Applications*.

[2] Sazzed, S. (2020). Comparative analysis of LSTM and Bi-LSTM in Bangla sentiment analysis. *Journal of Computer Science*.

[3] Islam, M.S., et al. (2021). Bangla-BERT: Language model pretraining for Bangla. *arXiv preprint*.

[4] Hassan, A., et al. (2017). Sentiment analysis on Bangla and Malayalam using SVM. *Int. J. Comput. Appl.*, 168(1).

[5] Chowdhury, S.A., & Chowdhury, W. (2014). Performing sentiment analysis in Bangla. *BRAC University*.

[6] Sazzed, S., & Islam, M.S. (2019). A deep learning approach to Bangla sentiment analysis. *IEEE Region 10 Conference*.

[7] Kabir, F., et al. (2021). Bangla sentiment analysis using CNN-LSTM. *J. Comput. Sci. Eng.*

[8] Islam, M.S., et al. (2021). Bangla-BERT: Language model pretraining for Bangla. *arXiv:2101.00276*.

[9] Bhattacharjee, A., et al. (2022). Bangla-RoBERTa: A multilingual model for Bangla. *ICON 2022*.