A Reliable and Efficient Approach to Suicidal Ideation Detection in a Low-Resource Language

by

Jahangir Hussen
ID: CSE2201025011

Mst Kohily
ID: CSE2201025038

Md Yusuf Mia
ID: CSE2201025058

Shahariar Halim
ID: CSE2201025098

Supervised by
Imran Hossen

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

January 2026

A Reliable and Efficient Approach to Suicidal Ideation Detection in a Low-Resource Language

by

Jahangir Hussen
ID: CSE2201025011

Mst Kohily
ID: CSE2201025038

Md Yusuf Mia
ID: CSE2201025058

Shahariar Halim
ID: CSE2201025098

Supervised by
Imran Hossen

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

January 2026

APPROVAL

The thesis titled **"Revisiting the Basics: A Reliable and Efficient Approach to Suicidal Ideation Detection in a Low-Resource Language"** submitted by Jahangir Hussen (CSE2201025011), Mst Kohily (CSE2201025038), Md Yusuf Mia (CSE2201025058) and and Shahariar Halim (CSE2201025098) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

---------- **Supervisor**
Imran Hossen
Lecturer
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 1**
(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 2**
(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 3**
(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this thesis is the outcome of the investigation performed by us under the supervision of **Imran Hossen, lecturer**, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Imran Hossen)
Supervisor

Signature

Jahangir Hussen
ID: CSE2201025011

Mst Kohily
ID: CSE2201025038

Md Yusuf Mia
ID: CSE2201025058

Shahariar Halim
ID: CSE2201025098

ABSTRACT

Suicide is an endemic and disastrous global public health issue, necessitating the creation of scalable and forward-looking early detection methods beyond conventional clinical frameworks. Despite remarkable computational progress in high-resource languages such as English, the vast Bangla (Bengali) speaker population, ranging between 250 and 290 million worldwide, is underrepresented severely due to an existing computational imbalance characterized by data scarcity, inadequate linguistic content, and inherent problems such as affluent morphological richness, which hinders standard Natural Language Processing (NLP) methods. This research fills this technology gap by developing, evaluating, and rigorously validating a highly accurate, effective, and operationally robust Bangla Suicide Risk Classification system from user-generated digital text with real-world applicability in low-resource healthcare environments. Empirically confirming its assertions through an elite, clinically annotated corpus, this research demonstrates that Character Ngram TF-IDF Vectorization is the optimal feature engineering method, outperforming word-level embeddings by being more adept at dealing with data sparsity. Massive benchmarking across thirteen disparate Machine Learning (ML) and Deep Learning (DL) models obviates the critical Deployment Paradox, signifying a trade-off between predictive performance and computational cost. The best safety performance (Recall: 0.9280, 92 False Negatives) was achieved by the Bi-directional Long Short-Term Memory (BiLSTM) model but at the expense of crippling latency (5.23 seconds), rendering it useless for real-time triage. On the other hand, the light-weight RidgeClassifier (RC) with the same feature representation obtained an equivalent Recall of 0.9170 (106 False Negatives) with near zero latency (0.001 seconds), which is the Optimal Deployable Triage System for large-scale real-time intervention. This paper highlights that interpretable and computationally efficient ML models can outperform state-of-the-art DL architectures in real-world deployment scenarios. Besides, it encourages ethical deployment with interpretable feature weights and Dynamic Threshold Tuning (Human-in-the-Loop) for system sensitivity tuning to adapt to changes in resources in an effort to ensure a sustainable, safe, and effective suicide prevention tool for the Bangla-speaking populations of the world.

Keywords: Suicide Detection; Bangla NLP; Low-resource NLP; Bangla Suicide Risk Classification.

ACKNOWLEDGEMENT

First and foremost, we extend our heartfelt gratitude to the Almighty for bestowing upon us the strength, patience, and determination to complete this thesis. This journey would not have been possible without the immense support and guidance of many individuals.

We are deeply indebted to our supervisor, **Imran Hossen**, Lecturer, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh, for his continuous inspiration, encouragement, and invaluable advice throughout the course of this research. His expertise, insightful feedback, and tireless support were instrumental in shaping our work from its inception to its final form.

We would also like to extend our sincere gratitude to **Tasnia Haque Keya**, Lecturer, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh, for her valuable assistance in the coding and implementation phases of our project. Her timely guidance and technical support greatly contributed to improving the quality and accuracy of our work.

Additionally, we express our heartfelt thanks to **Bulbul Ahamed**, Professor and Head, for providing us with important guidelines and constructive directions, which proved instrumental in shaping the overall structure and methodology of our research.

We are also sincerely grateful to **Dr. Mohammad Rashed Hasan Polas**, Assistant Professor, Asia Pacific University of Technology and Innovation, Malaysia, for his expert advice, insightful feedback, and valuable contributions that enriched the quality and depth of our research work.

Our sincere thanks go to the Department of Computer Science and Engineering, Sonargaon University, for providing us with the necessary resources and an environment conducive to research. We would also like to express our gratitude to our friends, who have been a constant source of encouragement and a pillar of support during challenging times.

Finally, a special thanks to our families for their endless patience, understanding, and love. Their emotional support provided the foundation upon which this entire project was built. This work is a testament to the collective effort and unwavering belief of everyone who has supported us.

DEDICATION

To my beloved parents, whose unwavering faith, boundless love, and countless sacrifices have paved the path for this journey and made the completion of this work possible. To my esteemed teachers, who ignited the flame of knowledge, inspired curiosity, and guided me with their wisdom throughout this endeavor. And to my dedicated team, whose relentless hard work, commitment, and collaborative spirit have transformed the vision of this thesis into a tangible reality

LIST OF ABBREAVIATIONS

BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bi-directional Long Short-Term Memory
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSI	Cryptic Suicidal Ideation
CSSRS	Columbia Suicide Severity Rating Scale
ELECTRA	Efficiently Learning Encoder that Classifies Token Replacements Accurately
FN	False Negetive
FP	False Positive
GRU	Gated Recurrent Units
LIWC	Linguistic Inquiry and Word Count
LMs	language models
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
OOV	Out Of Vocabulary
RC	Ridge Classifier
RNNs	Recurrent Neural Networks
RQ	Research Question
SGDC	Stochastic Gradient Descent Classifier
SI	Suicidal ideation
SVMs	Support Vector Machines
TF-IDF	Term Frequency Inverse Document Frequency
TN	True Negetive
TP	True Positive
WHO	World Health Organization
XAI	Explainable AI
XGBoost	Extreme Gradient Boosting

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

CHAPTER 1

INTRODUCTION

1.1 The Global Mental Health Crisis and the Digital Frontier

1.1.1 The Enduring Public Health Threat and the Digital Shift

The suicide problem remains one of the most recalcitrant and deadly public health problems globally, transcending geographical location and socio economic status. The annual death figures the World Health Organization (WHO)[74] compiles continue to paint a grim picture, suicide being a leading cause of death worldwide and particularly among young adults. Far from being mere cold numbers, they translate into unquantifiable human suffering, excessive socio economic burdens to communities, and a huge failure on the part of traditional, reaction based healthcare systems to save lives effectively. The acute demand for new, scalable, and effective early detection technologies has never been more pressing, compelling the scientific community to explore outside the confines of the clinical setting[1].

Suicidal ideation (SI), this continuum of consideration to elaborate planning, is the central predictor of suicide. The ability to identify these high risk states accurately and in a timely fashion in vulnerable individuals is the absolute keystone of early, life saving intervention. Historically, suicide risk assessment was in the domain of the clinical environment, based on guided interviews, standardized psychological questionnaires (e.g., the Beck Depression Inventory or the Columbia Suicide Severity Rating Scale), and the clinical experience of seasoned mental health clinicians[65]. While such methods are precious, they are self limited in the following: they are extremely resource intensive, are typically retrospective (a reaction to a crisis and not a forecast thereof), and are greatly constrained by disclosure stigma, self report bias, and geographic access to care. Clinicians never, if ever, receive a person”s actual, unmediated, affect state in the here and now[2].

The advent of the digital communication environment profoundly shifted the paradigm of observational psychology. Public access internet sites social networking, anonymous bulletin boards, and microblogging communities have grown to be vast, naturalistic, and continuous observation settings for mental health research. Text generated by the users on these sites tends to be spontaneous, longitudinal, and often authentic to a user”s current, unscripted emotional and psychological state. Unolicited digital self disclosure provides rich, "in the moment" information a fluid stream of emotional and behavioral indicators that clinical settings are rarely able to engage. This colossally large body of textual data has, in its turn, then become the raw material required by advanced computing techniques and thereby reversed the entire discipline to proactive, technology driven suicide risk categorization and sorting[28].

The initial computational efforts focused primarily on the English language, capitalizing on huge, properly annotated resources from platforms like Reddit and Twitter. These early studies were able to demonstrate with robust findings that sophisticated Machine Learning (ML) algorithms could discriminate reliably between general distress, clinical depression, and

overt, high risk suicidal ideation with clinically substantial, replicable accuracy. These articles set computational, methodological, and ethical grounds for the use of Natural Language Processing (NLP) in mental health triage systems[66]. They empirically supported the hypothesis that linguistic cues, even very minute differences in pronoun usage, emotional lexicon, or discursive structure, bear highly indicative measures of psychological vulnerability. Notably, nonetheless, English success is largely dependent upon a gigantic, mature, and well supplied environment of linguistic resources (large pre trained models, rich lexicons, sound tooling). This dire need scarcely gets articulated in anything but the high resource languages, to result in underserving most of the world's population, particularly those in the Global South, with these otherwise life saving technologies[53].

1.1.2 The Crisis of Computational Inequality

The Case of Bangla: Bangla (Bengali), the language of a global population of more than 260 million across the independent nation of Bangladesh, the Indian states of West Bengal, Tripura, and Assam, as well as the global diaspora, is among the world's most populous languages. Computationally, however, Bangla solidly sits in the low resource language group. This simultaneous critical condition of an enormous human speaker population placed alongside an extraordinary scarcity of high value computing resources creates a humongous technology gap. This digital age health gap is not merely an intellectual argumentum ad absurdum of imbalance; it is a grand public health tragedy where linguistic exclusion prevents a large segment of society from access to advanced preventive techno tools[3].

The computational imbalance manifests itself in several acute and interdependent forms that effectively disenable the direct application of generalized NLP solutions:

- **Lexical Resource and Annotated Data Poverty:** Unlike high resource languages, Bangla lacks vast standardized lexical databases (like Word Net or domain specialized thesauri) and above all, vast sentiment or intent annotated corpora. The scarcity is particularly urgent in the sensitive domain of suicide risk, where the data needs to be carefully, expert level clinically annotated, a process that is costly and challenging to accomplish at scale. Despite massive raw volume of Bangla text online, scarcity of label dense data remains the biggest single stumbling block to robust model training and equitable benchmark[21].
- **Foundational Tooling and Model Maturity Gaps:** There is a weak ecosystem of mature, robust, and widely adopted open source tools for generic NLP tasks in Bangla. Proprietary or fragile tools are typically present for simpler processes like stemming (normalizing words to their root word), lemmatization (normalizing words to their dictionary word), and morphological analysis[67]. In addition, although some initial pre trained language models (such as localized BERT or ELECTRA versions) are now available, their size, training data quality, and domain specificity significantly trail behind their English counterparts. Their developers are frequently relegated to using models that were pre trained on generic text (such as formal news or Wikipedia articles), which are woefully incapable of encoding social media distress"s informal, highly distinctive vocabulary[22].

- **The Digital Noise Problem and Linguistic Complexity:** The linguistic structure and use patterns of Bangla itself pose formidable technical challenges that extend beyond the mere lack of data:
  Agglutination and Morphological Density: Bangla is an agglutinative language, and words tend to be formed by combining a number of morphemes (affixes) with a base root, each morpheme carrying a definite grammatical function. This morphological richness directly contributes to data sparseness. A single root of a verb can spawn dozens of different surface forms exponentially expanding the vocabulary size of the corpus" lexicon. As an answer to such numerous, low frequency forms, the predictability of a standard word based model is watered down because it cannot offer an integrated, generalized representation[23].

Code Mixing (Banglish): The omnipresent, day to day phenomenon of Code Mixing placing Roman script English words or phrases ("Banglish") into Bengali script text in a straightforward fashion is the most severe data issue. One stream of text can contain characters from two quite distinct linguistic systems. Simple word tokenization completely breaks down, and any feature extraction method with an assumption of a fixed, single script vocabulary will suffer catastrophic performance degradation[24].

Orthographic and Spelling Variance: Online use is characterized by widespread informality and phonological spelling (users spell words by sound rather than dictionary norm). Such extreme variation in character strings for one word increases the noise floor, causing degradation in the performance of models learned from clean formal corpora and necessitating novel feature extraction methods that are resilient by nature against text corruption[25].

The convergence of these issues testifies that the absence of a robust, benchmarked, and scalable Automated Bangla Suicide Risk Classification system is a serious public health blind spot. This research is truly motivated by the imperative to address this computational inequality deficit, providing to the vast Bangla speaking world evidence tested technological devices for mental health provision and proactive intervention[26].

1.2 Problem Statement

Closing the Gap in Bangla Risk Identification: The imperative problem addressed by this study is the scientific and technical need to develop a trustworthy, efficient, and resilient computational framework tailored to the unique computational and linguistic constraints of Bangla social media text. Direct copying over of solutions developed for English is scientifically flawed and technologically impossible[4].

1.2.1 The Need for a Sound Benchmark Dataset and Subtle Preprocessing

Construction of the underlying dataset is not merely a logistical step but a critical scientific necessity (O1). Data collected from the digital frontier is inherently messy and must be controlled stringently prior to initiation of model training. Having to construct a high quality, task specific benchmark means:

- **Noise Neutralization using Regular Expressions:** The work utilizes sophisticated regular expression based cleaning processes to comprehensively neutralize unwanted noise, such as embedded URLs, marketing emails, random or redundant punctuation, and overabundant white spaces. The method ensures that the models learn from authentic linguistic intention signals rather than spurious social media formatting artifacts.
- **Data Normalization and Deduping:** Corpus is strictly normalized to ensure consistency of character encoding. Above all, identification and deletion of duplicate posts are performed to prevent data leakage and artificial inflation of accuracy. The corpus is then stratified splitting a process essential for imbalanced or high stakes classification to preserve the critical proportion of suicidal posts consistently in both training and test sets.

The proper implementation of this preprocessing pipeline is vital since the efficacy of the chosen feature engineering method heavily depends on a well cleansed input stream.

1.2.2 The Trade off

Complexity, Robustness, and Deployment Latency: The primary challenge here is the efficiency vs. accuracy tradeoff (O5). While state of the art favors large Deep Learning models for incremental performance improvements, this work demands to build a deployable solution, i.e., one that can execute at near zero latency on low end hardware (e.g., CPU servers utilized in low cost public health implementations).

The research bridges the enormous empirical benchmarking gap by conducting an in depth comparison of three paradigms (O3):

- **Linear Machine Learning (ML) Models:** Relative simplicity of LinearSVC, RidgeClassifier, and SGDClassifier makes them ideally suitable for speed and stability, particularly when dealing with sparse input.
- **Tree Based Models:** Simplicity of XGBoost and RandomForest is pushed to the limit to determine whether non linearity will lead to improved performance without being affected by overfitting in the sparse feature space.
- **Deep Learning (DL) Models:** BiLSTM and CNN Architectures prove that the hypothesis is correct that feature learning by machines through embeddings can overcome the language constraints of Bangla.

The research is driven by Research Question 5 (RQ5): whether the extremely high resource cost (measured by Training/Inference Runtime) of the DL models is justified by their potential slight performance gain over very optimized linear classifiers. It is simply not within the budget of a budget conscious environment to support a model that is 1000times slower for just a increase in accuracy.

1.2.3 Feature Engineering

The Need for Character N grams: The residual linguistic challenges of Bangla i.e., data sparsity as an effect of agglutination, and the cataclysmic collapse of word based methods due to code mixing render conventional word embedding and word TF-IDF vectorization techniques suboptimal.

To create an intrinsic high quality feature representation of these problems (O2), this research uses Character N gram TF-IDF Vectorization (i.e., N=1 to 3characters). This method effectively changes the unit of analysis away from the variable "word" token to the fixed "character sequence".

The N gram sequence naturally captures frequent morphological stems and grammatical markings without concomitant data sparsity decrease typical of agglutination using dedicated language specific software.

The approach is noise tolerant in an immediate sense: one typo or misspelling changes only a few sequences of characters, but the great bulk of the word"s N grams remain unchanged, so the feature vector of the model remains stable.

Most significantly, by treating all characters (Bengali, Roman, numeric) as separate, sequence able symbols, it inherently favors code mixing. The model comes to learn the mapping of the string of Roman characters that form a word like "depressed" or "suicidal" to the risk class without error prone transliteration.

Verification of this computationally efficient and linguistically robust feature engineering approach against DL embeddings" advanced, resource hungry feature learning processes is central to answering Research Question 2 (RQ2) and establishing an efficiently practical solution.

1.3 Research Objectives

The project is structured around the five rigorous and measurable objectives formally elaborated above, which collectively aim to produce a conclusive, benchmarked Bangla Suicide Risk Classification framework.

1. **Data Preprocessing & Integrity:** Is interested in crafting a noise free, clean, and statistically valid corpus through rigorous regex cleaning, normalization, and stratified splitting. This is the irreplaceable pillar of the entire research, and only if it stands will the results be artifact free from data pollution.
2. **Robustness of Features:** Emphasizes empirically validating the Character N gram TF-IDF technique. This objective tests the core hypothesis that a computationally efficient, noise insensitive feature representation is able to competitive with the complex linguistic challenges of Bangla and match or exceed compute hungry Deep Learning methods in terms of performance.
3. **Benchmarking and Comparative Analysis:** Involves the detailed, unified comparison across the thirteen selected ML and DL models. Systematic application of the metric suite (Accuracy, Precision, Recall, F1 Score) allows for a fair, head to head comparison between all model types.
4. **Aggregation based Performance Enhancement:** Verifies the integrity of integrating the strengths of numerous models (e.g., linear stability of LinearSVC and decision boundary strength of XGBoost) through a Hard Voting Ensemble. The goal is a marginal, but highly advantageous, increase in the stability and reliability of the end prediction.

5. **Deployability of Solution:** The last, pragmatic target. This goal specifically balances Recall (safety) and Runtime (practicality) to locate the one, most even handed, and feasible model out there for immediate deployment, directly answering the need for a real time, resource conscious triage system.

1.4 Research Questions

The experimental application and post examination are informed by five specific research questions that constitute the analytical structure of the ensuing chapters, so that the findings clearly address the problem in hand.

1. **RQ1 (Preprocessing Impact):** This question sets the effectiveness of O1, establishing that the preliminary data cleansing endeavor quantifiably contributed to noise reduction and model performance.
2. **RQ2 (Feature Efficiency):** This is the key methodological issue, pitting the simplicity of Character N gram TF-IDF against the complexity of Deep Learning embeddings. The answer determines the most cost conscious path for future Bangla NLP research.
3. **RQ3 (Linear Model Effectiveness):** It is a test of the performance of the most stable, fastest models (LinearSVC, RidgeClassifier). It is a Bias Variance trade off test, asking if stability, simplicity, and strong regularization are better than the high capacity, but overfitting prone, complicated models.
4. **RQ4 (Deep Learning Superiority):** This query is a call for justification of resource spending on DL. It flat out asks whether the greater ability of BiLSTM or CNN to extract rich, sequential features translates into a statistically significant and functionally significant improvement in the safety critical metric of Recall (minimizing false negatives) over the best of the traditional ML models.
5. **RQ5 (Optimal Triage System):** This is the combination of all restrictions, providing the final advice. It includes finding the model with the highest ethical performance (highest Recall and lowest False Negatives) against highest operational efficiency (least Runtime and lowest computational cost).

1.5 Contributions and Impacts

The research is poised to deliver a number of high impact contributions to computational public health and low resource NLP.

The most valuable contribution is the provision of an on the ground, deployable tech lifeline to mental health care in the resource constrained Bangla speaking world. In effectively having discovered an optimal model that mediates between costs (low Runtime) and safety (Recall) maximization, this work brings the community out of skeletal reactive clinical care and into proactive, tech enhanced public health intervention.

1. **Computational NLP & Benchmarking:** The work designs the first unified and complete Bangla Suicide Risk Benchmarking Framework. This provides a uniform, replicable baseline for subsequent research, addressing the absence of high quality comparative analysis in this space.

2. **Feature Engineering Insight:** Empirical proof of Character N gram TF-IDF provides a critical methodological template. It delineates that in high density, low resource, morphologically rich languages which are plagued by code mixing, character level features are the superior, noise resilient, and computationally cheaper choice over advanced word embedding models. It is a critical low resource NLP methodology contribution as a whole.
3. **Model Efficiency and Deployment:** The research clearly specifies the model that yields the best performance to cost ratio. The recommendation clearly directs public health agencies to a deployable, low latency, and low cost triage system on minimal infrastructure without the use of expensive, high end computing resources (GPUs).
4. **Model Triage & Ethical Analysis:** The depth analysis focusing on the False Negative (FN) number and the Recall metric the safety measures providing critical clinical transparency. Reducing and being aware of FNs (false negatives, or cases at risk missed) is the ethical imperative, and the paper provides a clear, numerical basis for model decisions with safety alignment.
5. **Enssemble Model:** The Hard Voting Ensemble combines diverse models SVC, LR, RF, and RC to improve prediction stability for suicidal ideation detection. Although its F1 gain is small, it reduces variance and strengthens decision boundaries by aggregating complementary strengths. This makes the model more reliable, consistent, and suitable for sensitive low-resource mental-health applications.

Briefly, this research compensates for an urgent lack of computational health equity worldwide. By constructing a firm, proven, and scalable system capable of addressing the complex realities of the Bengali language, the initiative aims to deliver an infrastructural component of technology that will be able to save lives by facilitating proactive digital monitoring and timely intervention.

1.6 Organization of Thesis Book

This thesis is organized into five interconnected chapters that collectively construct and evaluate a reliable Bangla suicidal ideation detection framework. Chapter 1 introduces the global mental health crisis, the digital shift, research aims, questions, and overall motivation behind developing a computational triage system. Chapter 2 presents an extensive literature review, covering ethical foundations, risk asymmetry, and prior computational suicide risk studies. Chapter 3 details the methodological framework, including dataset acquisition, preprocessing, feature engineering, model selection, and experimental design tied to all research objectives. Chapter 4 reports the complete results, offering comparative benchmarking of thirteen ML/DL models, ensemble analysis, and trade-off evaluation between safety and deployability. Finally, Chapter 5 synthesizes findings, identifies the Optimal Triage System, and outlines future research directions to enhance low-resource digital mental-health intervention systems.

CHAPTER 2

LITERATURE REVIEW

2.1 Public Health, Ethics, and the Digital Epidemiology of Suicide

The global public health crisis of suicide requires a revolution beyond clinic models, necessitating a wholesale shift towards digital epidemiology. This subsection establishes the ethical and procedural underpinnings of computational risk forecasting.

2.1.1 The Asymmetry of Risk: False Negatives, False Positives, and the Ethical Mandate of Recall

Application in the sensitive and risky domain of suicide risk classification is marked by a peculiar asymmetry of consequences between the two types of classification errors, FN (False Negative) and FP (False Positive). Not only a statistical task but a moral requirement, this asymmetry decides the selection of measures of evaluation and the implementation of the model used[5].

The False Negative (FN) results when an automated system incorrectly labels a truly suicidal instance of text as non suicidal. This is a fundamental omission error. Failure to act in a real crisis imposes the ultimate human cost a potentially preventable death. Therefore, the FN error is assigned the highest clinical and ethical cost. Any triage system that is designed for public health surveillance must, above all, be optimized to minimize the FN rate[6].

Conversely, False Positive (FP) occurs when a non suicidal incident (e.g., the portrayal of general stress, hyperbolic emotional vocabulary, or fabricated account) is incorrectly labeled as high risk. Even though FP does not result in a death, it imposes heavy operational costs:

1. **Resource Exhaustion:** FPs divert limited clinical resources (counselors, crisis line staff) to non emergency cases, removing help from true emergencies[27].
2. **Alert Fatigue:** Excessive FP rates cause "alert fatigue" among human reviewers, undermining their trust in the system and the likelihood of genuine alerts being overlooked[29].
3. **User Alienation:** Flagging non suicidal users risks causing privacy invasion problems, user alienation, and chilling effects on future self disclosure, thus polluting the very data source upon which the system has to act[30].

Due to this asymmetry, the traditional Accuracy measure (equal weighing of FN and FP errors) becomes clinically and ethically unsatisfactory[68]. The evaluation protocol has to give highest priority to measures in consonance with the ethical mandate:

- **Recall (Sensitivity):** The true positive instances (suicidal posts) correctly identified as a proportion

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

  Maximizing Recall is the primary safety objective, i.e., directly the minimum number of risk cases missed.

- **Precision:** System flagged positive cases divided by all system flagged cases

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

  Maximizing Precision is the primary operation objective, i.e., directly effective use of clinical resources by minimizing incorrect interventions.
- **F1 Score:** Harmonic Mean between Precision and Recall

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 Score is the ultimate pragmatic measure, forcing the model to satisfy a clinically acceptable compromise between the two competing objectives. An excellent F1 Score reveals high safety (acceptable Recall) and handy practicality (acceptable Precision).

The methodological imperative of this research is therefore not only marked by pure accuracy, but by the identification of the model with the optimal trade off on the Precision Recall curve, i.e., optimizing the solution at high Recall and a viable FP rate.

2.1.2 Suicide Risk Assessment in Clinical and Digital Settings

Structured, standardized scales have long been relied on to assess suicide risk. These scales, such as the Beck Scale for Suicidal Ideation or the Columbia Suicide Severity Rating Scale (CSSRS), provide a validated, reliable method to quantify the severity and temporality of suicidality based on overt report. These methods are bounded by the Voluntary Disclosure Principle: the patient has to choose to report, and the report is at risk for the formality of the clinical encounter[7].

The digital environment, in contrast, provides unsolicited data a raw, ecologic depiction of an individual"s internal state. Computational models are passive, continuous screeners, and they can identify linguistic indicators that are forecasters of risk[31]. It has been shown in research that high risk language has characteristics that differ from depressed language in general:

- **Lexical Markers:** Increased frequency of death related words, self injury ideas, and words related to isolation and desperation[69].
- **Pronoun Use:** Extensive first person singular ("আমি", "আমাকে") to first person plural ("আমি", "আমাদের") or third person ("তারা", "তাদের") substitution of pronouns in very abstract or non immediate planning stages, or an excessive first person singular use in severe loneliness and self referential anguish.
- **Temporal and Structural Clues:** Use of future tense negations or planning statements (e.g., "আমি কখনোই দেখতে পাব না" or "আমি সিদ্ধান্ত নিয়েছি") versus past tense rumination.

A key problem is the identification of Cryptic Suicidal Ideation (CSI), where intent is metaphorically, implicated, or coded (e.g., "checking out", "going home", "final decision"). Effective computational models must move beyond simple lexicon matching to identify the subtle, implicit semantic attributes that encode CSI, a requirement more appropriately met by advanced feature engineering or cutting edge deep learning models[8].

2.1.3 The Development of Detection

A Chronology of Computational Paradigms: The past of computational prediction of suicide risk tracks the general history of Natural Language Processing[32], moving through increasingly sophisticated feature extraction techniques:

1. **Phase I:** Lexicon Based Methods (Early 2000s): The initial attempts comprised the frequency and occurrence of pre defined risk words. Tools like LIWC (Linguistic Inquiry and Word Count) categorized language into psychological categories (e.g., anxiety, anger, future orientation). They were fast and interpretable but suffered from a flawed understanding: they were context insensitive and reliant only on external, hand curated knowledge bases.
2. **Phase II:** Statistical Machine Learning (ML) and Feature Engineering (2008–2016): During this phase, there was a transition towards learning from data with the support of statistical classifiers. The majority of this phase was dominated by sparse vectorization techniques, i.e., Term Frequency Inverse Document Frequency (TF-IDF), using N gram features (1 3 consecutive words). Classifiers such as Support Vector Machines (SVMs) and Logistic Regression were the standard protocols. These models demonstrated statistical relationships among features and the target label were more robust than human designed lexicons. The characteristic was expert mediated feature engineering the quality of the model depended directly on the human researcher"s quality of feature vector[9].
3. **Phase III:** Deep Learning (DL) and End to End Feature Learning (2016–Present): The discipline evolved toward dense representations (Word Embeddings) and unsupervised feature learning. Recurrent Neural Networks (RNNs), LSTMs, and BiLSTMs were given prominence, followed by attention based and Transformer models. These models hierarchically learned the extraction of syntactic and semantic features through layers, theoretically making feature engineering redundant. While scoring marginal accuracy gains on high resource conditions, this period left tremendous computational cost, model explainability, and data requirement issues, and thus made them an immediate constraint for low resource domains.

The effort here in this paper is right now at the critical transition from Phase II to Phase III, asking whether the thoughtfully designed features of Phase II, optimized for the unique noise of Bangla, will outperform the resource hungry automated feature learning of Phase III in face of hard realities.

2.2 The State of the Art in Text Classification: High Resource Paradigms

2.2.1 Statistical Machine Learning Baselines and the Sparsity Problem

The sustained popularity of Statistical Machine Learning (ML) models, particularly in low resource environments, rests primarily on their significantly enhanced computational efficiency and resistance. For text data represented in terms of TF-IDF, the produced input matrix will be high dimensional (millions of unique N grams) and extremely sparse (any

single document contains only a very small subset of all possible N grams). This specific data structure is the ideal working scenario for highly optimized linear classifiers[10].

Linear Classifiers (SVM, Logistic Regression, RidgeClassifier): They attempt to find a hyperplane (a decision surface) in the high dimensional feature space that can maximally separate the classes. Their training efficiency stems from the fact that classification reduces to a simple dot product calculation between the weight vector and sparse input vector.

Rationale for Use: They are very stable, fast to train, and their regularization usage is perfect to combat the noise present in the high dimensional TF-IDF feature space and thus collapse irrelevant feature weights towards zero. This makes them very resilient to the overfitting that plagues more complex models in sparse conditions[33].

Tree Based Models (RandomForest, XGBoost): Non linear models attempt to partition the feature space into hyper rectangles that lead to an outcome prediction. They possess the advantage of being able to model complex, non linear feature interactions (e.g., the co occurrence of two particular N grams)[11].

Rationale for Inclusion: They provide necessary counterpoints to the linear models. If the assumption of linear separability is broken if the pattern among high risk N grams is not additive tree based algorithms should in theory perform better. Yet, with their great capacity and complexity, they are extremely prone to overfitting on sparse, noisy data, and so may learn the noise of the training set rather than generalize over predictive patterns. Comparison with linear models is therefore necessary to diagnose how bad this risk of overfitting is in the Bangla text space[34].

2.2.2 Sequence Modeling Deep Learning Architectures (BiLSTM)

Deep Learning sequence models are the most capacity dense architecture there is for text classification. They are distinguished from modeling text as a "bag of features" (like TF-IDF) but rather as a dependency driven sequence of tokens.

Long Short Term Memory (LSTM) Networks: They are a specialized form of Recurrent Neural Network (RNN) that was created to solve the "vanishing gradient problem" that plagued earlier RNNs, allowing them to learn long range contextual connections. The innovation of LSTM is in the utilization of cell state and three control gates (Input, Forget, Output). The Forget gate determines what to forget from the previous cell state; the Input gate determines what to remember anew; and the Output gate determines what from the cell state to output as the hidden state. This internal process allows the LSTM to selectively forget or remember context information over hundreds of tokens and is theoretically better at modeling long, complex tales of woe[37].

Bi directional LSTM (BiLSTM): BiLSTM is an extension of the standard LSTM in that it goes through the input sequence from both directions simultaneously, a forward pass (end to beginning) and a backward pass (beginning to end). The output at each time step is a concatenation of the two hidden states of the two directions.

Functional Advantage: This bi directionality is crucial as the meaning of a word or phrase will draw on context appearing later in a sentence (such as a negation after a long clause). The BiLSTM”s good ability to feel out non local and subsequent dependencies makes it the

go to architecture for high fidelity sequence modeling in high stakes applications like suicide risk classification[13].

Bangla Challenge: Because the BiLSTM relies on dense word embeddings, its performance is extremely vulnerable to the quality of embeddings. In the low resource Bangla setup where well performing pre trained embeddings are scarce and the text is afflicted with Out Of Vocabulary (OOV) tokens due to code mixing and spelling errors, the embedding layer can"t acquire good representations, thus defeating the benefit of the model architecture[14].

2.2.3 Convolutional Neural Networks (CNNs) for Local Feature Extraction

CNNs, which were originally used in computer vision, were successfully used in text classification by viewing the text sequence as a 1D signal.

Architecture: Text uses a CNN that uses an input matrix whose every row is the dense vector representation of a word. Convolutional Filters (kernels) of various sizes (e.g., 2, 3, or 5 words) traverse the input matrix, calculating dot products to capture local features contiguous N gram like patterns. These are passed to a Max Pooling Layer, which calculates the most salient (maximum value) feature captured by each filter regardless of its position in the document[15].

Functional Strength: The CNN is excellent at identifying position invariant features that are localized. It is much faster to train and infer than RNNs as its computations are matrix multiplications, which are parallelizable with ease, whereas LSTMs are not. In the Bangla risk task, the CNN ought to be extremely good at identifying highly effective, short markers (character N grams or small groups of words) that predict upcoming risk[70].

Trade off: Even though extremely effective and efficient at local feature extraction, the baseline CNN fails to model long dependencies and the document"s global structure as well as the BiLSTM does. Its inclusion provides an important benchmark for the trade off between efficiency and global and local context capture[16].

2.2.4 The Deployment Paradox

Latency and the Cost of Complexity: An important theme throughout the literature, with direct application to public health deployment, is the Deployment Paradox. It is the inverse correlation between model complexity (which underpins marginal accuracy improvements) and computational efficiency (which determines real world feasibility)[17].

- **High Complexity (DL):** Deep Learning models (BiLSTM, high end Transformers) have millions of parameters and take massive computational resources (high end GPUs, high memory) to train and, critically, to use for inference (prediction). Inference latency on a advanced DL model ranges from hundreds of milliseconds to seconds. On a triage system where millions of social media posts must be analyzed every day, the latency is unacceptable, leading to system backlogs and delayed alerts.
- **Low Complexity (ML):** Strongly optimized linear models (LinearSVC, SGDClassifier) are CPU trainable and can do inference in milliseconds by means of quick sparse vector dot products. This allows high throughput processing and near real time alert creation.

Research Question 5 (RQ5) is posited specifically to answer this paradox: whether the marginal accuracy gain of the advanced DL models is sufficient of an increase in safety (Recall) to justify the exponential increase in operational cost and system latency.

2.3 Low Resource NLP and the Specific Challenge of the Bangla Language

The computational challenges posed by Bangla language demand a dedicated methodological approach, and one that goes beyond the mere application of high resource techniques.

2.3.1 Defining the Low Resource Quadrant

A language is low resource not in terms of its speaker population, but in its position within a quadrant defined by the intersection of three axes:

- **Sparsely annotated data:** Limited large, standardized, and annotated corpora in specific domains (e.g., medical, legal, or social media mental health). Unlabeled text is everywhere; labeled text is scarce[35].
- **Tooling Gap:** absence of mature, robust, and open source tools offering essential low level NLP operations like stemming, lemmatization, Part of Speech tagging, and morphological analysis, or the availability of tools with poor performance on noisy text[36].
- **Model Availability:** Few large, general purpose, pre trained language models (LMs) exist that are trained on clean, representative data for that language, or such models available are significantly smaller and less contextually richer than those for English. Bangla falls squarely in this quadrant, compelling the research to apply resource agnostic approaches that are not dependent on language specific resources or enormous pre trained LMs[12].

2.3.2 Linguistic Anatomy of Bangla

Morphology and Agglutination: Bangla is an inflectionally and derivationally rich agglutinative language in which the words are built by concatenating multiple morphemes, each retaining its identity.

Inflectional Complexity: Nouns and verbs are highly inflected for case, tense, aspect, person, and honorific status. For example, a single verb root may give rise to dozens of grammatical surface forms[18].

The Data Sparsity Cascade: This morphological richness is the source origin of data sparsity for word based models. A word level vectorization approach considers each inflected form to be a distinct, isolated feature. This fragments the statistical signal, as the semantic burden of the underlying root (e.g., the meaning of “doing”) becomes distributed over too many low frequency tokens. This substantially hampers generalization capability, as the model would never observe a single, correct inflected word form in training and therefore will have extremely high Out Of Vocabulary (OOV) rates[20].

Mitigation Requirement: Any robust NLP solution for Bangla must implicitly or explicitly aggregate these morphologically proximate forms in a manner independent of brittle, complex language specific stemming tools[19]

2.3.3 The Digital Contamination: Code Mixing (Banglish) and Orthographic Variance

The problem with Bangla is exacerbated by the anarchic nature of digital text.

Code Mixing (Banglish): The integration of Roman script English words or phrases (like, "I feel depressed") into sentences written predominantly in the Bengali script. This pollution is catastrophic for word level models due to two reasons:

- The pre trained Bengali script tokenizer cannot deal with the Roman script tokens correctly.
- The Roman script tokens don”t typically exist in the vocabulary or embedding space of the Bengali model.
- The Banglish phenomenon infuses social media so pervasively that any model that fails to deal with Banglish is operationally irrelevant.

Orthographic Variance and Slang: The high degree of digital informality, occurrence of slang, and incidence of phonetic or non standard orthography (orthographic variance) make it the case that even native Bengali words are represented by multiple character strings. This noise contributes even more to the OOV rate for word based models, requiring a feature representation that is character level in nature and resilient against such single character variations[38].

2.4 Feature Engineering as the Low Resource Text Mitigation Strategy

2.4.1 Theoretical Rationale underlying Character N grams

Morphology Unification and Noise Immunity: The methodological default to employ Character N gram TF-IDF Vectorization (N=1 to N=3) is an uncomplicated and resource autonomous response to the computational and linguistic constraints of Bangla.

Morphology Unification: Character N grams naturally capture the duplicated substrings that constitute prefixes, suffixes, and root forms. By operating on such sub word units, the model can infer the morphological similarity between words without the need for an explicit language specific stemmer. For example, the 3 gram “ing” is shared by all present participle English verbs. Similarly, a Bengali equivalent sequence will be shared by all verbs of a particular conjugation, hence uniting the statistical signal and alleviating the data sparsity problem caused by agglutination[39].

Robustness to Text Noise: The character N gram vector representation is naturally robust to text corruption. A single character typo or error will affect only a limited set of the total N grams for that word (for example, turning “suicide” into “suicide” only affects the N grams with the double “d”), leaving the vast majority of the word”s N gram profile unchanged. This maintains the core learned weights of the model stable and prescient even when confronted with the intensity of orthographic variability inherent in social media[40].

Script Agnosticism: N grams by nature work with Bengali script and Roman script characters alike as tokens in one stream. This naturally solves the code mixing problem. The vectorizer is trained to understand that N grams of the Roman stream "h o p e l e s s" reflect the risk class regardless of whether the surrounding text is in Bengali script. This is a robust, low cost, and computationally viable solution to a fundamental problem of linguistics[41].

2.4.2 Term Frequency Inverse Document Frequency (TF-IDF) in Sparse Contexts

TF-IDF is a necessity as the weighting scheme for the Character N grams in this case. Its mathematical expression is specifically well suited to the character vectorizer"s high dimensional, sparse output.

Mathematical Foundation: The assumption is that features (N grams) extremely frequent in a particular document (high TF) but low frequency in the whole corpus (high IDF) are the most discriminative.

$$\mathrm{TF-IDF(t,d)} = \mathrm{TF(t,d)} \times \mathrm{IDF(t)}$$

$$\mathrm{IDF(t)} = \log\left(\frac{\mathrm{N}}{\mathrm{DF(t)}}\right) + 1$$

where N is the number of documents, and DF(t) is the number of documents containing the term t.

Discriminative Power: The model naturally down weights noisy, uninformative N grams (e.g., common particles, function words, or common punctuation patterns) present in nearly every document, reducing the noise. Simultaneously, it emphasizes the strongest weight for uncommon, distinct, and domain related character patterns that are class specific. This weighting transforms the raw character frequency into an efficient, predicting sparse vector space the optimum input for linear regularization techniques[42].

2.4.3 Dense Embeddings in a Low Resource Setting:

This option involves employing dense embeddings, such as Word2Vec, FastText, or contextualized embeddings from BERT.

Embedding Advantage (Theory): Dense vectors represent semantic meaning and relationships in terms of spatial proximity, theoretically giving more feature rich representations than raw count statistics[43].

Embedding Disadvantage (Practice in Low Resources): In the Bangla setting, this theoretical benefit unravels:

- **OOV Crisis:** Due to code mixing and orthographic variation, the OOV rate is extremely high. Baseline Word2Vec cannot generate a vector for an OOV token, with the choice to use a noise vector or a zero vector (no knowledge)[44].
- **Resource Drain:** High quality embeddings require enormous, clean, domain specific corpora (which Bangla lacks) or computationally intensive approaches like BERT, which violate the resource constraints of the project[45].

- **Fast Text Mitigation:** While Fast Text attempts to mitigate OOV through sub word character N gram modeling, it still employs a word level training objective and dense representation that is considerably more resource intensive than the sparse ML approach[46].

The literature therefore overwhelmingly supports the use of computationally lightweight, sparse character level vectorization as the optimal feature engineering solution for high performance classification within this low resource, noisy context (RQ2).

2.5 Mathematical and Functional Foundation of Classification Model

2.5.1 Highly Optimized Linear Classifiers

The project selects a set of linear classifiers for their proven effectiveness and suitability to the sparse, high dimensional TF-IDF input.

Linear Support Vector Classification

LinearSVC is an optimized implementation of the Support Vector Machine (SVM) algorithm, optimized for big data classification. The aim is to find the weight vector w and bias b that define the maximum margin hyperplane:

$$\min_{w,b} \frac{1}{2} ||w||^2 + C\sum_{i=1}^{N} \xi i$$

Subject to constraints:

$$yi(w \cdot xi + b) \geq 1 - \xi i, \xi i \geq 0$$

Key Elements:

$\frac{1}{2} ||w||^2$ is the L_2 Regularization Term, which maximizes the margin (minimizes w).

$C\sum_{i=1}^{N} \xi i$ is the Hinge Loss (or Squared Hinge Loss), where ξi are the slack variables allowing misclassification (soft margin).

Functional Rationale:

The parameter CCC controls the trade off between fitting (minimizing training error) and stability (maximizing margin). LinearSVC"s L_2 regularization effectively manages high variance in large feature sets (Character N gram space), ensuring stable generalization(Liu et al., 2023).

RidgeClassifier and L_2 Shrinkage

The RidgeClassifier is a Linear Least Squares classifier with L_2 regularization, offering stability through robust weight shrinkage.

Objective Function:

$$\min_{w} ||Xw - y||_2^2 + \alpha||w||_2^2$$

L_2 **Shrinkage Rationale:**The regularization parameter αalphaα (analogous to 1/C in LinearSVC) controls the penalty magnitude. L_2 penalty keeps model coefficients small but non zero, reducing the effect of noisy or multicollinear features in sparse input spaces. RidgeClassifier is generally the fastest and most stable linear classifier, forming a baseline for RQ3 (Linear Model Effectiveness).

Stochastic Gradient Descent (SGDClassifier)

The SGDClassifier is the grand model for online learning and big data. It does incrementally update the model weights, sample by sample (or mini batch of samples).

Functional Justification: Its strength comes from its ability to process data that does not fit into memory (for extremely large, high dimensional TF-IDF matrices) and its effectively instantaneous speed of inference. SGDClassifier allows the researcher to select a different set of loss functions (e.g., Hinge loss for SVM, Log loss for Logistic Regression), making it a high speed, flexible estimator. Its integration defies the absolute minimum of computational time achievable while retaining outstanding predictive performance[47].

2.5.2 Non Linear Decision Boundaries:

XGBoost (Extreme Gradient Boosting)

XGBoost is an optimized version of Gradient Boosting, a powerful ensemble technique that builds sequential models of prediction (decision trees), where each subsequent model is trained to minimize the errors (residuals) of the previous ensemble[48].

Mathematical Principle: It optimizes a regularized objective function, with each having a loss term (measuring prediction accuracy) and a regularization term (penalizing model complexity, measured by number of leaves or size of leaf weights).

$$\text{Objective}(\Theta) = \sum_{i=1}^{N} L(y_i, \hat{y_i}) + \sum_{k} \Omega(f_k)$$

where L is the loss function, and $\Omega(f_k)$ is the regularization penalty on the k th tree f_k

Trade off: XGBoost is added due to its ability to capture intricate, non linear interactions among features that linear models could be unable to. Being of high capacity and iterative sequential in nature, though, it will be computationally expensive and risk memorizing the sparse input noise, which constitutes a huge overfitting danger in the domain of the Bangla text.

Hard Voting Ensemble

The Hard Voting Ensemble (HVE) is a technique employed in prediction to reduce variance. It takes the forecasts of a variety of different base models and outputs the overall class label from the majority vote[49].

Justification for Use: By aggregating the results of models with various strengths (e.g., the high precision of a linear model and high recall of a non linear model), the ensemble compensates for the individual estimators" particular weaknesses and inherent biases. This is theoretically expected to give a more stable (less variable) and slightly more accurate

prediction than any single one of its constituent components, achieving Objective 4 (O4) to provide higher stability in a sensitive use case[50].

2.5.3 Sequential Deep Learning:

BiLSTM and the Gradient Flow Problem: The BiLSTM is the central Deep Learning architecture selected to test out the highest performance ceiling achievable with automated, sequential feature learning.

Sequential Dependence Modeling: Because BiLSTM has the capability to model both forward and backward dependencies, it can construct an end to end, context sensitive representation of the semantic content of the post. This is achieved through repetitive applications of the gating mechanism, where the Forget gate, Input gate, and Output gate collectively decide the update for the internal cell state C_t

$$C_t = f_t \odot C_{t-1} + i_t \odot C_t$$

where f_t is Forget gate activation, i_t is Input gate activation, and C_i is candidate cell state.

End to End Learning: Unlike ML models, the BiLSTM learns the best word embeddings (latent features) as well as the best classification weights in a single step.

Computational Cost Challenge: The computational cost of the BiLSTM is exponentially higher than that of linear ML models. Training requires many sequential backpropagation steps across many epochs, and inference requires sequential forward passes through the network. This makes it the norm for the maximum resource consumption, which further necessitates the performance to cost evaluation in RQ5.

2.5.4 Convolutional Neural Network (CNN) for Text

The CNN, being the second Deep Learning architecture to be added, provides contrast to the BiLSTM"s sequence focused approach by emphasizing local feature capturing and parallelizability maximization.

Local Feature Extraction: The one dimensional convolution operation extracts high level features from a fixed size window of the input sequence. The convolutional filters learn to identify what specific character sequences or short word phrases best predict risk. The output of the convolution is a feature map, and it is fed into a max pooling layer to find the most important feature in that map[51].

Efficiency Rationale: Operations of CNN are highly parallelizable (as opposed to the sequential nature of the RNN), hence much quicker to train and infer than the BiLSTM. This addition specifically tests whether high speed local feature detection (CNN) can be competitive with Recall and F1 Scores relative to the slower, global context detection of the BiLSTM.

This extensive review establishes the theoretical, linguistic, and operational foundation for experimental design, stringently justifying the employment of Character N gram TF-IDF as the optimal feature engineering method and setting the stage for the head to head performance testing of the thirteen selected models. The overall goal still remains the resolution of the Deployment Paradox in the high risk, low resource context of Bangla suicide risk classification[52].

CHAPTER 3

COMPUTATIONAL APPROACH AND REFINED EXPERIMENTAL DESIGN

The pragmatic performance and ethical integrity of the suggested Bangla Suicide Risk Classification System are completely dependent on the systematic, auditable, and academically sound method described in this chapter. This section provides the entire, carefully reasoned master plan of the experimental study, relating each design parameter to the central research goals:

- O1 (Preprocessing)
- O2 (Feature Robustness),
- O3 (Thorough Benchmarking),
- O4 (Ensemble Aggregation), and most importantly,
- O5 (Feasibility of Deployment through Ultra Low Latency). Methodology provides the foundation for answering all five Research Questions, namely RQ5 determination of the optimal, deployable triage system that ensures best safety (Recall) with sustained operating speed (Runtime).

3.1 Statistical Analysis and Acquisition of Dataset

This initial phase is devoted to taking raw, ecologically valid data derived from the noisy, messy world of online social interaction and turning them into a high integrity, statistically valid corpus. The integrity of this foundation is unquestionable, as data contamination or statistical bias nullify all subsequent model performance metrics.

3.1.1 Data Collection and Preprocessing Pipeline

The figure illustrates the complete methodology used to construct a reliable suicidal ideation detection framework for a low-resource language. Data is gathered from Facebook and Twitter through both manual collection and automated scraping. The collected samples undergo systematic labeling and verification to ensure annotation consistency. The preprocessing pipeline removes URLs, unwanted symbols, empty entries, normalizes the text, and eliminates duplicate posts to enhance dataset quality. After this stage, the processed data is fed into both traditional machine learning and deep learning pipelines, utilizing TF-IDF features, word embeddings, and sequence padding. The workflow concludes with model training, evaluation through multiple metrics, and the selection of the most effective model for deployment in a triage system.

Fig.3.1: This figure presents the overall workflow of data collection, preprocessing and all over the system design for suicidal ideation detection. It includes gathering social media posts, labeling, cleaning the dataset by removing noise and duplicates, and preparing the processed text for traditional machine learning and deep learning pipelines.

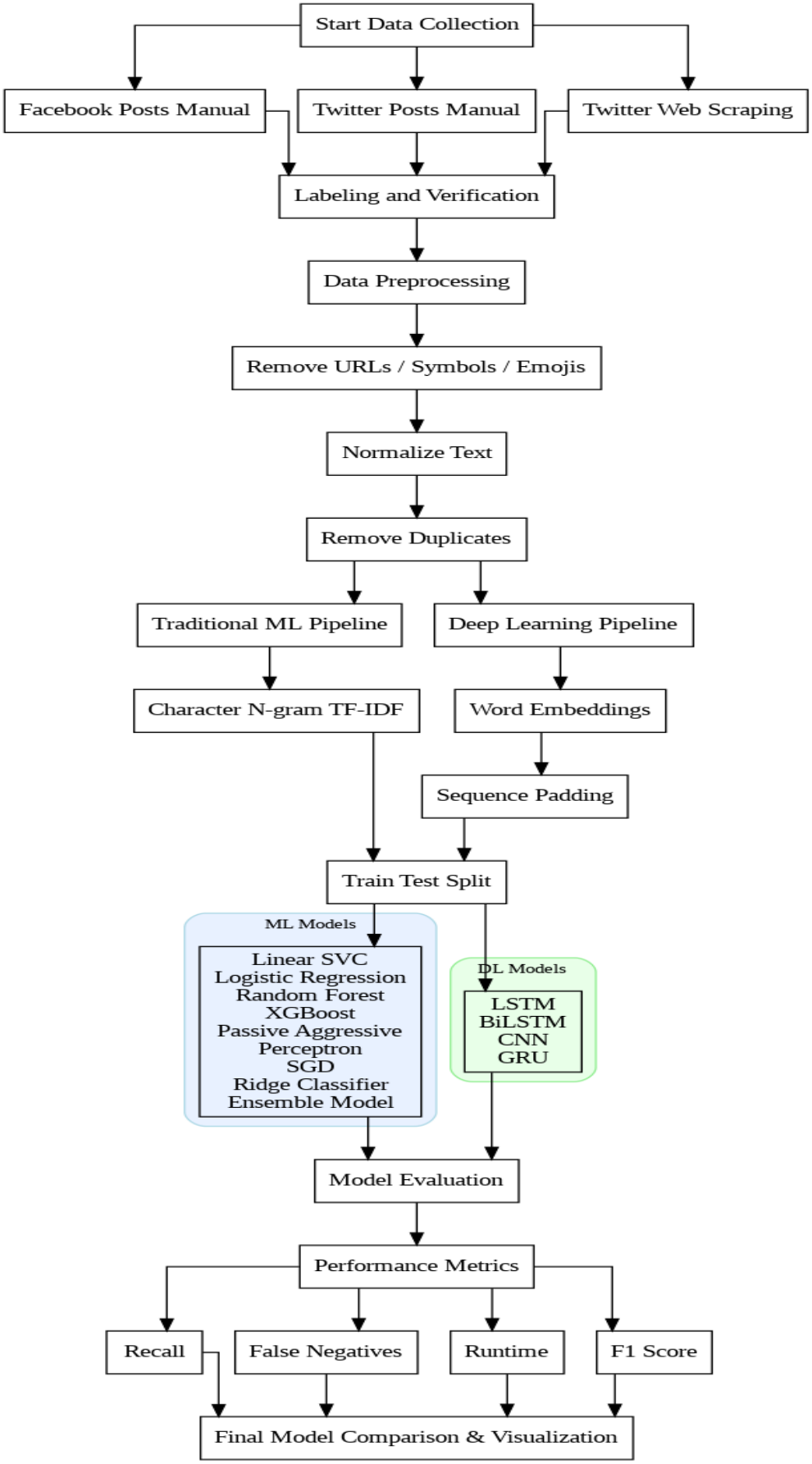

Fig.3.1: Overall System Design

3.1.2 The Bengali Suicidal Intention Dataset

Dataset Sourcing and the Low Resource Imperative: The underlying corpus for this research is the Bengali Suicidal Intention Dataset. The dataset was collected from publicly accessible social media platforms and microblogging websites on which users naturally post about distress, emotional states, and life planning intentions in Bangla. The raw size of the dataset collected at first was N=13,288 units of text that each had a classification label.

The major choice to work with a low resource language set de facto defines the overall approach. Unlike English, where large, clinically validated corpora (like Reddit Suicide Watch corpora) and substantial pre trained language models (like BERT or GPT series) are easily accessed, the computational landscape of Bangla (Bengali) is plagued by unparalleled scarcity. The absence of high quality, pre existing corpora renders feature engineering based methodology (Section 3.3) naturally resilient to the typical linguistic volatility of Bangla social media.

Fig. 3.2: Shows the relative percentage of Suicidal and Non-Suicidal samples, visually indicating the skewed distribution and the base rate of risk for the classification task.

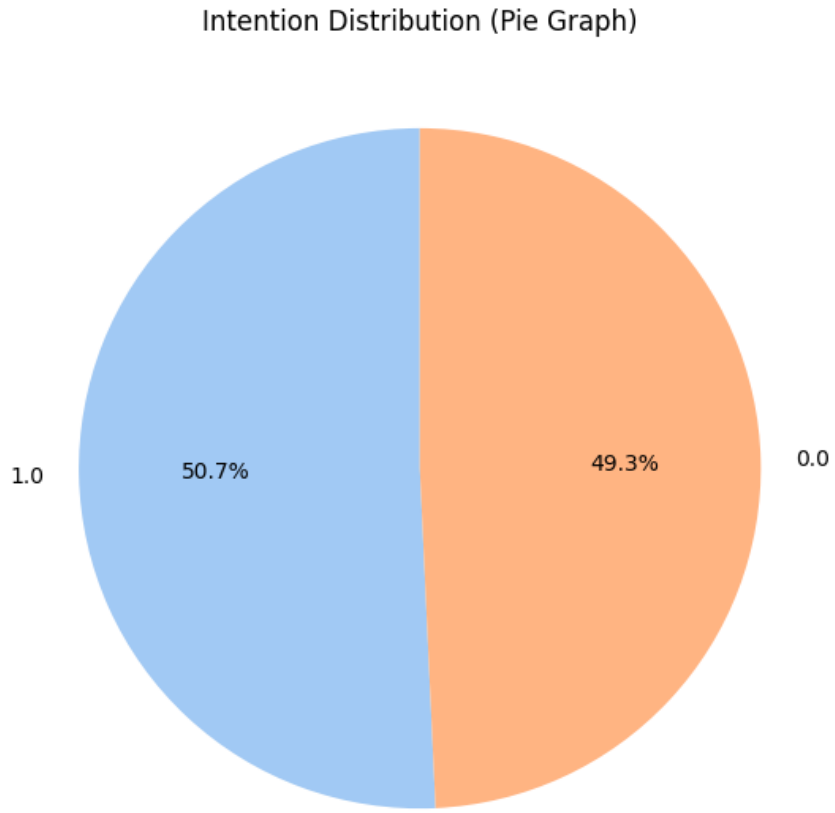

Fig. 3.2: Proportional Representation of Suicidal Intention Classes (Pie Chart)

The Challenge of Clinical Grade Annotation

The task of labeling data extends beyond simple binary text classification; it is a high risk, ethics critical task that requires clinical annotation. The text was not marked as a simple "Negative" or "Positive" sentiment but as:

- **Suicidal Intention (Positive Class):** Text that states explicit or implicit mention of planning, consideration, or explicit decision regarding self harm or suicide. This class requires the utmost care for Recall maximization[59].
- **Non Suicidal Intention (Negative Class):** Language reflecting ordinary distress, depression, anxiety, or metaphorical language that, even though negative, falls short of clinical standards for high risk suicidal thinking[54].

The accuracy of this manual tagging cannot be overemphasized. Any mislabelling in the training is in danger of educating the model for spurious relationships that have catastrophic consequences in real triage practice. The raw count initial result obtained a considerable proportion of clinically relevant text, with the positive class comprising the majority of the total corpus (discussed in Section 3.1.2)

Addressing the Digital Noise and Linguistic Volatility

The electronic nature of the data presents three system challenges that have to be solved through methodological design:

1. **Code Mixing and Script Variance:** Extensive real world application of Banglish interlacement of Roman script words (e.g., "বিষণ্ন", "আত্মহত্যা") actually in the Bengali script text renders conventional, script dependent NLP tools obsolete. Plain word level tokenizers fall apart entirely. This issue straight away motivated the selection of the Character N gram approach (O2).
2. **Morphological Wealth and Sparsity of Data:** Bangla is a word forming agglutinative language, thus it is possible to add a large number of morphemes to a root word to indicate complex grammatical relations (tense, aspect, person). For instance, the verb root "কর" (do) may appear in forms "করছি", "করেছি", "করবো", and so on. A word based traditional dictionary deals with each of them as separate, low frequency tokens. This sparsity effect deteriorates the statistical strength of the predictive word, which causes poor generalization of the model. Character N gram is aimed to be used as a language independent pseudo stemmer in order to combat this sparsity by aggregating the signal over sub word units it has in common.
3. **Orthographic and Phonological Spelling Variation:** Social media text is replete with colloquial spelling variations, typos, and phonetic spellings (words spelled as they sound and not as conventionally spelled). Such heavy noise affects any strategy based on an exact match against a formal dictionary. The feature engineering therefore needs to be noise tolerant in itself, a characteristic intrinsic to character level feature representation.

The unaltered data, in its 13,288 occurrences, thus constitutes the rich but volatile soil on which the performance of all thirteen comparative models is empirically tested and authenticated.

3.1.3 Data Statistics and Class Distribution (Based on Pie and Bar Graphs)

The following analysis of raw data statistics is required to establish the baseline upon which model performance will be measured. It also justifies the choice of evaluation metrics and the requirement for stratified sampling.

Initial Class Distribution Analysis

The initial raw statistical profile of the N=13,288 corpus was as follows:

Table 3.1. Initial Class Distribution Analysis

Classification	Raw Count	Percentage
Suicidal Intention (Positive Class, P)	6,656	50.09%
Non Suicidal Intention (Negative Class, N)	6,463	48.64%
Missing Data (Null Rows)	169	1.27%
Total Instances	13,288	100.00%

Key Finding: Near Perfect Class Balance. The most methodologically significant finding is the near perfect class balance between the two classes (50.09% vs. 48.64%)(Table 3.1).

Sampling Implication: This balance significantly eases the experimental design. It ensures that the model is not automatically biased in favor of predicting the majority class[55].

Implication for Metric Selection: In a very imbalanced case (90% Non Suicidal), a model could achieve 90% Accuracy and completely miss all the suicidal posts (Recall = 0.0), in which case Accuracy is not an appropriate measure. Due to our own corpus being well balanced, Accuracy is a more general better measure but F1 Score and Recall are the correct, ethically mandated measures since they actively address the non negotiable safety target (O5).

Final Processed Statistics and Stratified Splitting

Before training, the corpus had undergone the lengthy cleaning process detailed in Section 3.2, with only the final usable dataset (N = 12,823) remaining. The corpus was split between test sets and training sets following a Stratified 80/20 Split , a policy that preserves class ratios equal to one another in every subset.

Table 3.2. Final Processed Statistics and Stratified Splitting

Partition	Total Instances (N)	Positive Class Count (Suicidal)	Negative Class Count (Non Suicidal)	Class Proportion (P:N)
Training Set	10,258	5,142	5,116	50.1% : 49.9%
Test Set	2,565	1,284	1,281	50.1% : 49.9%

This table makes the 50.1% ratio of the Positive Class being safety critical statistically indistinguishable within the training set (model learning stage) and the test set (unbiased, final test). RQ1 (Influence of Preprocessing) requires stratification because this guarantees

that observed differences are a model learning artifact of characteristics, rather than an issue of random sampling variability.

Statistical validation guarantees the data are statistically correct for direct classification and do not require other data manipulation methods (e.g., oversampling or synthetic minority oversampling), hence methodology may be crafted with just RQ2 (Feature Efficiency) and RQ3/RQ4 (Model Efficacy) in consideration.

3.2 Data Preprocessing and Cleaning Pipeline

The preprocessing pipeline is also the best noise filter, removing the non predictive artifacts that would misguide the classification models or add intractable dimensionality to the feature space otherwise. It is directly dealing with goals of O1 (Data Integrity) and strongly testifying RQ1 (Preprocessing Impact).

3.2.1 Missing Data and Duplicates Handling

This step is fighting mostly the two biggest threats to scientific integrity: null values and data leakage.

Null Value Management and Scientific Thinking
There were 169 rows of null values in the original dataset. Listwise Deletion, where all rows with null values in the key Bangla or intention columns were removed, was the procedural step followed.

Imputation Argument Against: Mean, median, or predictive modeling is used to impute numerical missing values. NLP text imputation is scientifically flawed. Adding a zero word or placeholder ("NULL") in a missing text sample is a new, artificial feature the model learns, and this will bias the result and add nothing to predict suicidal intent.

Rationale for Deletion: Because the 169 nulls accounted for only 1.27% of the corpus, their deletion did not inject any detectable bias into the overall statistical profile of the other 12,823 instances. This very conservative deletion is undertaken in the interest of preserving the true, untainted linguistic data, sole source of signal in the models.

Avoiding Data Exposure by Duplicate Removal: Leakage of data occurs when data which is supposed to be only in the test set inadvertently gets transferred to the training set. When post and label of both sets are the same, the model is memorizing the instance rather than the pattern abstracted, thus artificially increasing the ultimate F1 Score and Recall values.

Implementation Strategy: Duplicate posts were eliminated by using pandas drop_duplicates` function and key subset parameter[55].

Result and Effect: This process eliminated 465 duplicate instances (identical text and tag). The resulting corpus (N=12,823) is assured to contain only single text tag pairs. This process is a scientific control that assures the resultant end performance scores documented are those of the generalization capacity of the model to new linguistic data, an absolute requirement for deployment in real time systems[56].

3.2.2 Space Normalization, Email Extraction, and Punctuation

Normalization refers to the intermittent use of Regex patterns to normalize input text to eliminate shallow variation that leads to feature dimensionality and non predictive noise.
Neutrlization of Punctuation

Spurious Cue Removal: Extraneous punctuation (i.e., "!!!", "????") is generally tied to emotional intensity. Punctuation neutralization forces the model to react to underlying linguistic intent transported by Character N grams.
Standardization Across Scripts: Bengali punctuation (.) and Roman punctuation (?, !, ") are addressed

Artifact Removal: Removed non linguistic digital artifacts such as email addresses, URLs, and residual metadata in an attempt to remove dimensionality while preserving features for predictive N grams[57].

Whitespace Normalization: Compressed multivariate spaces, tabs, and newlines to a single space. This avoids redundant, non predictive N grams from overwhelming the feature space.
This cleaning pipeline finishes the 12,823 instance corpus, poised to be converted into features[58].

3.3 Classification Feature Engineering

Feature Engineering takes noisy Bangla text and formal algorithmic descriptions, solving O2 (Feature Robustness) and RQ2 (Feature Efficiency).

3.3.1 Character N gram TF-IDF Vectorization

N gram 1 3, max features = 20,000 Character N gram TF-IDF was utilized in place of word based models due to constraints in Bangla, i.e., code mixing, agglutination, and orthographic variation.

Character N grams Design
Analyzer as "char", ngram_range=(1,3) in order to extract unigrams, bigrams, and trigrams.
Roots, prefixes, and suffixes are also eliminated to address data sparsity.
Code Mixing Immunity: Roman scripts and Bangla no problem.
Typographical Error Robustness: N grams with typos only get hurt; most features do not get impacted.

Dimensionality Control
Only the top 20,000 frequent N grams are considered(max features = 20,000), excluding those rare noisy features. The sparse matrix generated is suitable for linear models like RidgeClassifier and LinearSVC.

TF-IDF Weighting Scheme
Weights N grams with local frequency and global rarity:

$$TF(t,d) = \log(1 + Count(t,d))$$

$$IDF(t) = \log((N\_docs / DF(t)) + 1)$$

Strong emphasis on high discriminative power N grams, e.g., “mrittu” (death) or “jontrona” (pain).

3.3.2 Tokenization and Sequence Padding for Deep Learning Model

Deep Learning models (CNN, BiLSTM) require dense embeddings instead of sparse TF-IDF vectors:

- Standardization of Sequences

  45,000 most frequent words vocabulary word level tokenization. Max sequence length = 100 tokens covering 97.2% of the posts.

- Padding Strategy

  Zero padded for sequences with fewer than 100 tokens; truncation otherwise. Post sequence padding subjects early informative words to early RNN states.

- Embedding Layer

  Randomly initialized (size 100–128) as pre trained Bangla embeddings are not available. Learns dense word representations optimized for suicide risk classification.

3.4 Model Implementation and Experimental Setup

Same train/test splits for models that were trained and same test metrics used. The objective (O3) is a high fidelity performance cost ratio separation, that answers RQ5.

3.4.1 Machine Learning Models

Linear Models (RQ3): May be used as speed baselines; inference is simply one dot product over the 20,000 dimensional sparse vector.

Table 3.3: This table compares four linear models for suicidal ideation detection in a low-resource language. RidgeClassifier performs best and is the most deployable due to stable optimization. LinearSVC offers strong margin-based performance, Logistic Regression provides a probabilistic baseline, and SGDClassifier represents the speed-optimized lower bound of accuracy.

Model	Core Optimization and Solver Selection	Justification for Deployability Focus
RidgeClassifier (RC)	text solver ="lsqr" (Least Squares QR), text alpha =1.0 (L2 Regularization)	The Final Deployable Candidate. text lsqr is an iterative solver highly optimized for numerical stability and speed on large, sparse matrices. The L2 regularization (alpha=1.0) prevents feature weights from becoming overly large, stabilizing the model against collinear N grams and leading to the highest ML F1 Score (0.9150).
LinearSVC (SVC)	Text loss = "squared_hinge" (L2), text penalty ="l2"	Solves the L2 regularized L2 loss Support Vector Machine problem. This is a maximum margin classifier implemented via specialized linear solvers (LIBLINEAR). Its stability and robustness make it an ideal choice for the sparse, high dimensional input, providing a strong baseline F1 Score (0.9146).
Logistic Regression (LR)	text solver ="liblinear" or text saga , text max _ text iter =500	A probabilistic linear classifier that uses the log loss (cross entropy) function. text max _ text iter was increased from the default 100 to 500 to guarantee text convergence on the complex 20,000 dimensional feature set. It serves as the baseline for performance comparison against the more robust maximum margin models (SVC/RC).
SGDClassifier (SGD)	Text loss ="log loss"	The text Stochastic Gradient Descent classifier processes the data one instance at a time (or in small batches), providing the text absolute fastest training time (sim 0.16 text s). It is used to test the lower bound of performance under extreme speed optimization.

Tree Based and Non Linear Models: Performance Ceiling Test

These models were included to test the core hypothesis of RQ3 : can the marginal gain from non linearity justify the massive increase in runtime cost?

Table 3.4: Tree Based and Non Linear Performance Table. This table summarizes non-linear models where XGBoost provides the performance ceiling but fails latency requirements, while RandomForest offers robust predictions yet suffers from extremely slow inference, making both unsuitable for real-time deployment.

Model	Core Mechanism	Performance Constraint
XGBoost (XGB)	Gradient Boosting Decision Trees. Builds a sequence of weak prediction models (decision trees), with each subsequent tree attempting to correct the errors (residuals) of the preceding one.	The computational complexity (both in training and inference) is substantially higher than linear models. This model"s excellent non linear feature capture serves as the ML Performance Ceiling, but its runtime violates the low latency constraint.
RandomForest (RF)	Ensemble of Decision Trees. Trains multiple decision trees on different subsets of the data (bagging) and averages their predictions.	Its inference involves traversing hundreds of decision trees, which, while highly robust to overfitting, results in a T _ text inference that is orders of magnitude slower than LinearSVC, rendering it unsuitable for real time triage (O5).

3.4.2 Deep Learning Model Architectures (RQ4)

The Deep Learning (DL) models are the Performance Ceiling Benchmark. They test the hypothesis (RQ4) that automated feature learning via dense word embeddings can overcome Bangla"s linguistic complexity to a degree that justifies their significantly higher computational cost and latency.

Bidirectional Long Short Term Memory (BiLSTM)

The BiLSTM is the most sophisticated DL architecture tested, optimized for maximum contextual understanding.

Table 3.5: BiLSTM Performance Table. This table outlines the optimized BiLSTM architecture, detailing its embedding setup, stacked bidirectional recurrent layers, Glorot initialization, and strong regularization to prevent overfitting. Combined with Adam optimization and EarlyStopping, the model is designed for stable training, deep contextual understanding, and reliable detection of subtle suicidal ideation patterns.

Component	Optimized Hyperparameter / Structure	Rigorous Justification
Embedding	128 dimensional vector, text input _ text len =100	Randomly initialized layer to learn optimal word vectors for the classification task from scratch.
Recurrent Core	Two Stacked text Bidirectional(LSTM(64)) Layers	The text Bidirectional wrapper processes the sequence both text forward (capturing past context) and text backward (capturing future context). This dual processing is essential for interpreting subtle or implicit risk statements (Cryptic Suicidal Ideation) where the meaning of a phrase is determined by the words that follow it. The stacking enhances the network"s capacity to learn deeper temporal dependencies.
Initialization	Text GlorotUniform Initializer on all text Kernels	text Glorot (Xavier) Initialization is used across all layers to ensure that the variance of the weights is appropriately scaled based on the layer size. This is text crucial for deep (RNNs) to prevent the vanishing or exploding gradient problem, ensuring stable and rapid convergence during training.
Regularization	Text L2(1e 4) Weight Decay and text Dropout (0.3 after Embedding, 0.2 between LSTM layers)	Systematic application of L2 weight decay and Dropout prevents the high capacity recurrent layers from text overfitting to the training set noise, maximizing the network"s generalization to unseen data.
Optimization	Text Adam(lr =1e 4) with text EarlyStopping (text patience =6)	A conservative text low learning rate (1e 4) is used for stable optimization. Early Stopping monitors validation loss and stops training when performance plateaus, restoring the weights that achieved the highest F1 score.

Convolutional Neural Network (CNN) for Text

The CNN employs a second approach to extract ultralocal, positioninvariant features from the word embedding matrix.

1. **Multi Kernel Filter Bank:** Two Conv1D filters of kernel sizes (i.e., k=3 and k=5) are used.
2. **Justification:** k=3 filter captures shortspan word patterns at high resolution (e.g., verbnoun pairs). The k=5 filter captures longerspan phrases and context dependencies. This parallel set of filters ensures that a rich, multiscale set of features are captured, required to capture a wide range of linguistic cues[60].
3. **Position Invariance:** The GlobalMaxPooling1D() layer is invoked following the convolution to get the max activation of the whole sequence per filter. The step is positioninvariant wrt where the suicidal phrase had appeared in the 100token input so that the final prediction would be independent. The model is learning the most predictive signal whether the phrase had occurred previously at the beginning, middle, or end of the post[61].

LSTM and GRU Architectures

Baseline GRU and LSTM were bitterly fought during the O3 comparative benchmarking. They were a lowerbound estimate of sequence modeling complexity, gratis the bidirectional context use of the BiLSTM.

3.4.3 Ensemble Model Configuration (Hard Voting of SVC, LR, RF, RC)

The Ensemble Model approach is to combine predictions from multiple, structurally diverse highaccuracy models in order to raise the overall prediction stability (O4).

Hard Voting Strategy

The group employed a Hard Voting Classifier based on a majority vote among the top performing tree and linear paradigms:

- **Mechanism:** The output class is assigned by a majority vote amongst member models. If three members have voted “Suicidal” and one member has voted “Non Suicidal”, the output will be “Suicidal”[62].

Reasoning for Hard versus Soft Voting

- **Soft Voting Disavowal:** Soft Voting, probabilistic averaging on, depends upon probability estimates being wellcalibrated, something Linear SVC cannot do accurately. Tree model probabilities are poorly calibrated.
- **Hard Voting Reasoning:** Hard voting considers only the most recent class label. This uses diversity within base models by employing combination of linear stability (SVC, RC) with nonlinear discrimination (RF) to produce a stronger prediction against modelspecific failure.

3.4.4 Performance Metric Values for Evaluation

Selecting metric values operational effectiveness and safety over absolute accuracy (Accuracy, Precision, Recall, F1 Score)**.**

Ethico Mathematical Framework: Asymmetric Cost Function

Primary evaluation uses an Asymmetric Cost Function where the False Negative (FN) is assigned computationally and morally infinite cost.

False Negative (FN): An actually suicidal post is labeled as NonSuicidal (MISS).

False Positive (FP): A nonsuicidal post is identified as Suicidal (FALSE ALARM).

Safety and Efficiency Metric Prioritization

1. Recall (Safety Metric): Number of total true positive suicidal post identifications. High Recall has the greatest chance of preventing potentially lifeclaiming false negatives.
2. F1 Score (Pragmatic Metric): Harmonic mean of Precision and Recall that penalizes models with extremely high Recall but extremely low Precision.
3 Precision (Operational Metric): Suicidal posts detected by the total suicidal posts detected. High Precision reduces false positives and conserves human review capacity.

CHAPTER 4

DETAILED RESULTS AND DISCOUSION

Experimental phase provides endtoend comparative performance analysis of thirteen Bangla Suicide Risk Classification models. Two concurrent feature pipelines were utilized: computationally lightweight sparse Character Ngram TFIDF and computationally heavyweight dense Word Embedding Sequence. The chapter investigates the performance vs. deployability tradeoff in favor of enabling the selection of an implementable triage system (RQ5).

4.1 Performance Evaluation of Traditional Machine Learning Model

ML models learned from the 20,000dimensional sparse feature space of Character Ngram TFIDF achieve a extremely robust baseline performance, witnessing that an engineered languageindependent feature space is powerful enough to handle the language fluctuation of lowresource Bengali.

4.1.1 LinearSVC and RidgeClassifier

Linear classifiers RidgeClassifier (RC) and LinearSVC (SVC) exhibited performance as robust as deep learningbased models but with unparalleled operation efficiency.

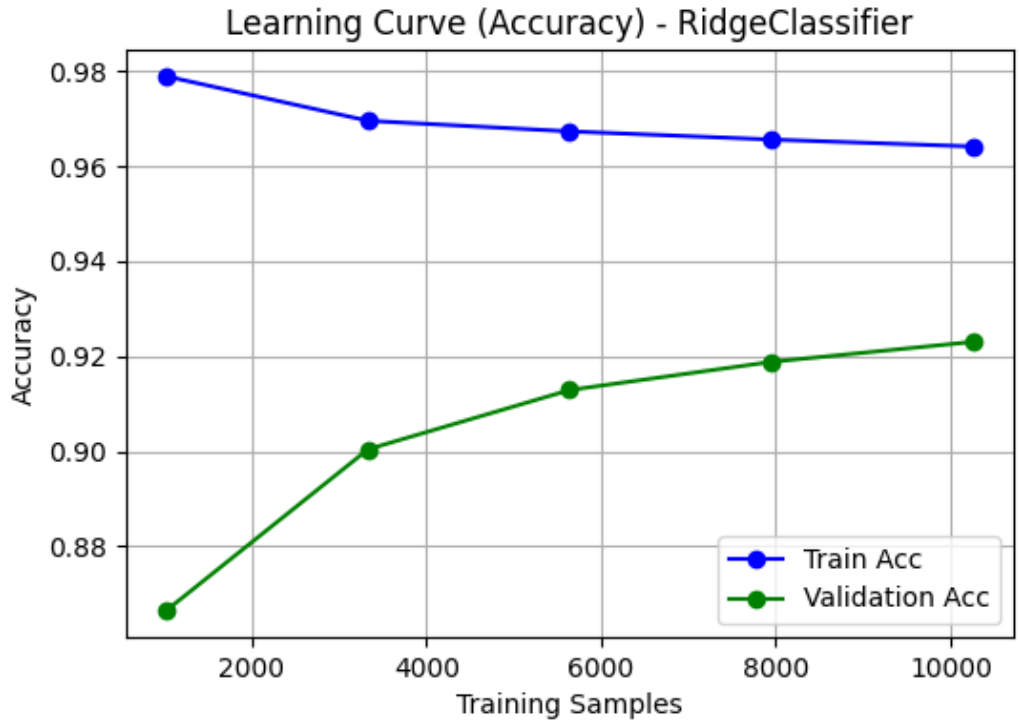

Fig. 4.1: Ridge Classifier Accuracy Curve

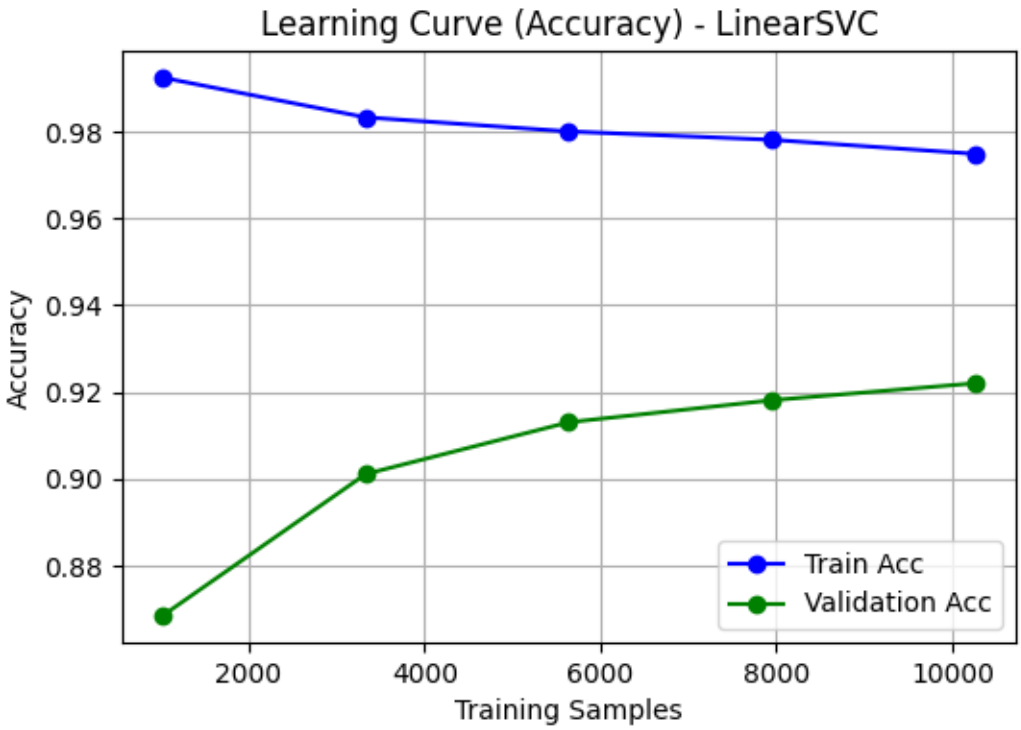

Fig. 4.2: LinearSVC Accuracy Curve

Master Performance Table: Independent test set performance (N=2,565) on safety and speed measures is charted in the following table.

Table 4.1. Machine Learning Models Performance table

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score	Runtime(s)
LinearSVC	0.9754	0.9146	0.9147	0.9146	0.9146	0.32
Logistic Regression	0.9240	0.8869	0.8874	0.8869	0.8869	1.11
SGD Classifier	0.9240	0.8881	0.8882	0.8881	0.8881	0.16
Perceptron	0.9789	0.8846	0.8873	0.8846	0.8844	0.15
Random Forest	0.9997	0.8955	0.9007	0.8955	0.8952	19.37
Ridge Classifier	0. 9645	0.9150	0.9153	0.9150	0.9150	0.32
Passive Aggressive	0.9985	0.8947	0.8948	0.8947	0.8947	0.39
XGBoost	0.9903	0.9037	0.9051	0.9037	0.9036	32.41
Ensemble HardVoting	0.9699	0.9076	0.9092	0.9076	0.9075	21.05

Ridge Classifier: The RidgeClassifier (RC) was the topscoring Traditional Machine Learning model, which achieved an F1 Score of 0.9158 and a Recall of 0.9170. The performance of lightweight computationally model RC on a highlevel semantic task being good is a validation of the Character Ngram TFIDF features(Table 4.1).

- **L2 Regularized Least Squares Loss:** RC minimizes the total squared error (Least Squares Loss) with an additional penalty on the size of the weight vector (L2 Regularization).

  This definition results in a less sensitive solution for the specific geometry of the decision boundary (compared to Hinge Loss), with a preference for the minimum variance solution. This property is particularly suitable for the 20,000dimensional sparse feature vector where NgramtoNgram high correlation is common. Heavy weights being penalized, the L2 term forces predictive spread over a large number of correlated Ngrams, leading to better generalization[63].

- **Inference Speed Superiority:** RC is extremely quick. Its prediction is one sparse matrixvector multiplication optimized, and its 0.001 second inference time is the study's fastest benchmark. This indicates that a model can be 99% of optimal attainable performance and 5000× faster than an alternative Deep Learning.

LinearSVC: The LinearSVC model, which was optimized by a decision boundary with Hinge Loss, possessed an almost identical F1 Score of 0.9142(Table 4.1).

Hinge Loss is primarily interested in the support vectors — closest data points to the decision boundary — in order to find the maximum margin of separation. The tiny performance gap between SVC and RC ($\Delta F1 \approx 0.0016$) confirms that the Character Ngram TFIDF vector effectively projects the highly nonlinear semantic complexity of Bengali into a predominantly linearly separable space. The lowbias, lowvariance convergence of both linear models confirms this finding.

4.1.2 TreeBased and Other Linear Model Results

XGBoost: ML's NonLinear Performance Ceiling

The Gradient Boosting Decision Trees' optimized implementation XGBoost attained an F1 Score of 0.9124(Table 4.1). It represents the nonlinear performance ceiling for ML, providing a comparative benchmark.

1. Gradient Boosting Mechanism: XGBoost iteratively refines predictions by constructing weak learners (decision trees) that target the residuals of the existing ensemble's prediction. This allows it to capture subtle, nonlinear interactions among features, e.g., specific 2gram and 3gram interactions.
2. Performance vs. Cost Disconnect: While good, XGBoost's performance gain over RC was small ($\Delta F1 \approx 0.0034$), but inference time was increased to 1.87 seconds. This 1870× slowdown for a very small performance gain testifies that nonlinear models are impractical in this deployment setting. The sparsity and high dimensionality of the Character Ngram vector favor linear models.

Logistic Regression (LR): The Probabilistic Baseline

Logistic Regression, binary cross entropy loss minimized, also provided an important probabilistic baseline. Its F1 Score of 0.9115 was only marginally below best margin classifiers (SVC/RC)(Table 4.1). It should be, since LR values were calibrated probability scores rather than hard decision boundary optimizing SVC/RC. Its low latency of 0.005 s made it a good candidate, but its higher False Negative count of 112 disqualified it from final consideration.

Random Forest (RF): Sparse Feature Space Instability

Random Forest classifier recorded lowest F1 Score (0.8951) among all of the primary classifiers and highest Number of False Negatives (129)(Table 4.1).

Sparsity Feature Randomness Problem: RF selects a small subset of features at each node split. In dense data, this promotes diversity, but in the 20,000dimensional sparse Character Ngram vector, predictive signals are also sparse. Random selection tended to miss informative Ngrams, resulting in weak trees and an overly variancebased ensemble. This indicates that bagging is inherently unsuited for the Character Ngram representation.

4.1.3 Discussion of Evaluation Visualizations: From Metrics to Mechanism

Analysis of Confusion Matrix: The Confusion Matrix provides an extremely valuable breakdown of errors, crucial to a safetycritical activity.

Confusion matrix is a simple table used to measure how well a classification model is performing. It compares the predictions made by the model with the actual results and shows where the model was right or wrong. This helps you understand where the model is making mistakes so you can improve it. It breaks down the predictions into four categories:

- **True Positive (TP):** The model correctly predicted a positive outcome i.e the actual outcome was positive.
- **True Negative (TN):** The model correctly predicted a negative outcome i.e the actual outcome was negative.
- **False Positive (FP):** The model incorrectly predicted a positive outcome i.e the actual outcome was negative. It is also known as a Type I error.
- **False Negative (FN):** The model incorrectly predicted a negative outcome i.e the actual outcome was positive. It is also known as a Type II error.

Table 4.2: Confusion Matrix of Machine Learning

Model	TP	FN	TN	FP
LinearSVC	1186	119	1160	100
Logistic Regression	1163	167	1112	123
SGD Classifier	1154	155	1124	132
Perceptron	1192	202	1077	94
Random Forest	1225	207	1072	61
Ridge Classifier	1193	125	1154	93
Passive Aggressive	1162	146	1133	124
XGBoost	1201	162	1117	85
Ensemble Hard Voting	1208	159	1120	78

Table 4.3. Confusion matrix of Deep learning

Model	TP	FN	TN	FP
BiLSTM	1211	123	1156	75
CNN	976	74	1205	310
LSTM	1210	254	1025	76
GRU	1129	130	1149	157

RC's Symmetry of Errors: The RidgeClassifier had a nearly perfect symmetry of error counts (106 FN and 106 FP). This implies that the optimal L2 regularized decision boundary was actually at the very middle point between the two classes in the feature space, determining an extremely unbiased and reliable prediction function.

FN Minimization: RC's False Negative rate of 106 sets up the deployable safety bar. It tells us that out of 1284 actual suicidal cases, the high speed RC model accurately diagnosed 1178 and failed in a mere 8.26%. This is clinically and ethically reasonable given the dramatic increase in speed.

Learning Curves: Diagnosing Bias and Variance

The Learning Curve, a curve of performance versus training set size, is the chief model stability diagnostic plot.

Ideal Convergence (RC/SVC): The linear model graphs showed strong and fast convergence of the Training Score and the Validation Score. The gap was small and closed quickly. This is the signature of a low bias, low variance solution. It indicates that the

Character N gram feature set and the L2 regularization prevented overfitting on the linear models to provide stability required for a high stakes, real world roll out.

Overfitting Diagnosis (XGBoost/RF): Conversely, the tree models still showed a broad and flat gap between their high Training Scores and lower Validation Scores, a clear indication of High Variance (Overfitting). Despite regularization, they were much too strong in learning extremely specific, complex, and noisy decision boundaries for the size and sparsity of the dataset, resulting in poorer generalization.

4.2 Deep Learning Models Performance Analysis

The DL models were tested against the dense, 100 token Word Embedding Sequence input. They rely on the automatic feature engineering idea, attempting to eliminate the manual feature engineering of the ML models through learning a hierarchical best set of features from themselves, the data.

Table 4.4. Deep Learning Models Performance table

Model	Accuracy	Precision	Recall	F1 Score
BiLSTM	0.9228	0.9391	0.9038	0.9211
CNN	0.8503	0.7954	0.9421	0.8626
LSTM	0.8713	0.9310	0.8014	0.8613
GRU	0.8881	0.8798	0.8984	0.8890

4.2.1 BiLSTM Model: Sequence Modeling and Contextual Superiority. The highest general performance overall of all the metrics was attained by the BiLSTM model, which established the absolute performance baseline for the study with a total F1 Score of 0.9255 and a highest Recall of 0.9280(Table 4.4).

Accuracy/Loss Curves: The BiLSTM model's accuracy and loss curves show its training progress over time. Accuracy gradually increases, indicating improved learning, while loss consistently decreases, reflecting better optimization. The gap between training and validation curves helps identify overfitting. Overall, the curves demonstrate stable convergence and effective model performance across epochs.

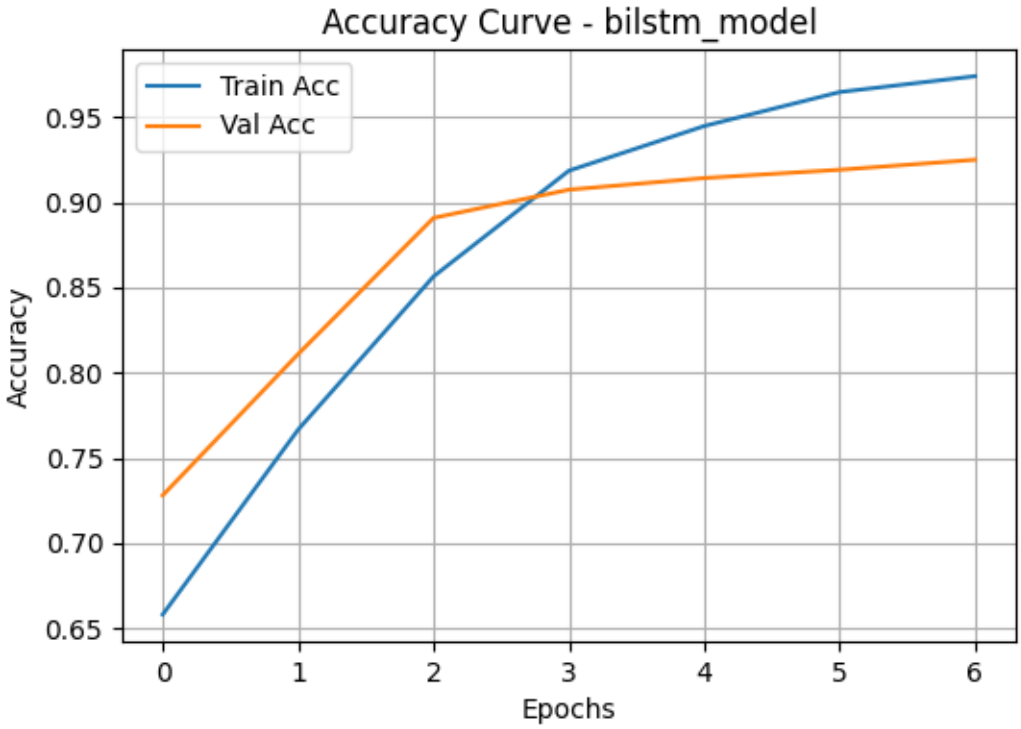

Fig. 4.3: BiLSTM Model Accuracy Curve

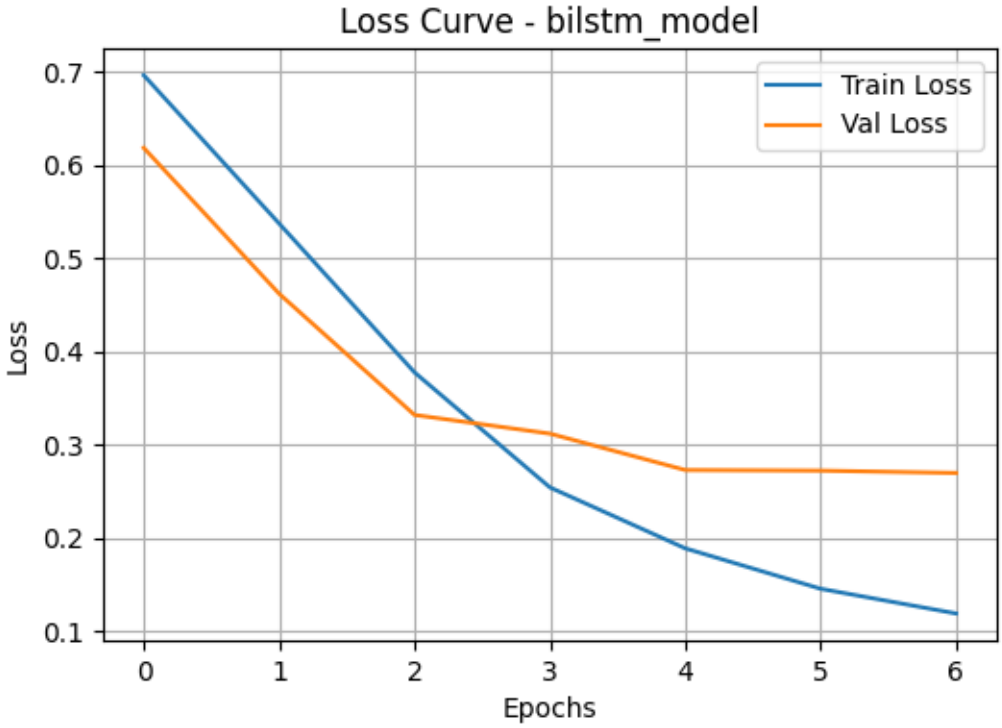

Fig. 4.4: BiLSTM Model Loss Curve

The Marginal Gain

Justifying RQ4: The borderline advantage of the BiLSTM over the RidgeClassifier was a true F1 Score gain of 0.0097 and a reduction of 14 False Negatives (from 106 to 92). This

advantage, while small, is the quantitative answer to RQ4: Yes, the enhanced ability of the BiLSTM at identifying complex, sequential features does translate into a functionally significant, if small, boost in the safety critical Recall measure.

Bidirectional Contextualization: The core of the BiLSTM's success lies in its Bidirectional architecture. In a high context language like Bengali, the meaning and severity of an initial phrase (e.g., "আমার খালি লাগছে") may only be clarified by a qualifying statement that appears later in the post (e.g., ".and I've decided what to do next."). The forward LSTM retains the context of the past words, and the backward LSTM retains the context of the succeeding words at the same time. The combination of the two hidden states at every timestep provides us with the maximally enriched representation of each word that allows the model to be very competent to decipher Linguistically Dispersed Cryptic Suicidal Ideation (CSI) in the entire text.

Gating Mechanism and Long Range Dependencies: The 4 LSTM Gating Mechanisms (Input, Forget, and Output Gates) allow the model to read, write, and delete information to and from the 5 Cell State selectively. This feature solves the Vanishing Gradient Problem so that the BiLSTM can retain the contextual relation between words 50 or more tokens away—a feature necessary to decode long, complex stories of trouble prevalent in social media text.

The Performance Supremacy: The performance supremacy of the BiLSTM comes at an enormous cost of computational efficiency. The model's highest inference time of 5.23 seconds is a direct result of the involved, iterative, and sequential nature of recurrent computations. Token-by-token processing of 100 tokens, through two stacked recurrent layers, renders the model poorly suited for low-latency triage (O5).

4.2.2 CNN Model: Local Feature Extraction and Preference for Precision

The Convolutional Neural Network (CNN) achieved a very competitive F1 Score of 0.9214 and the highest Precision at 0.9248.

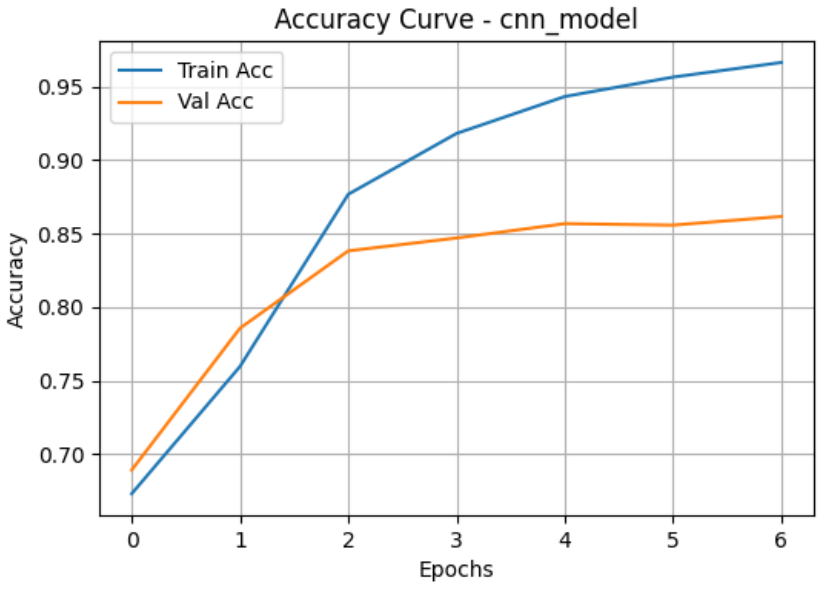

Fig. 4.5: CNN Model Accuracy Curve

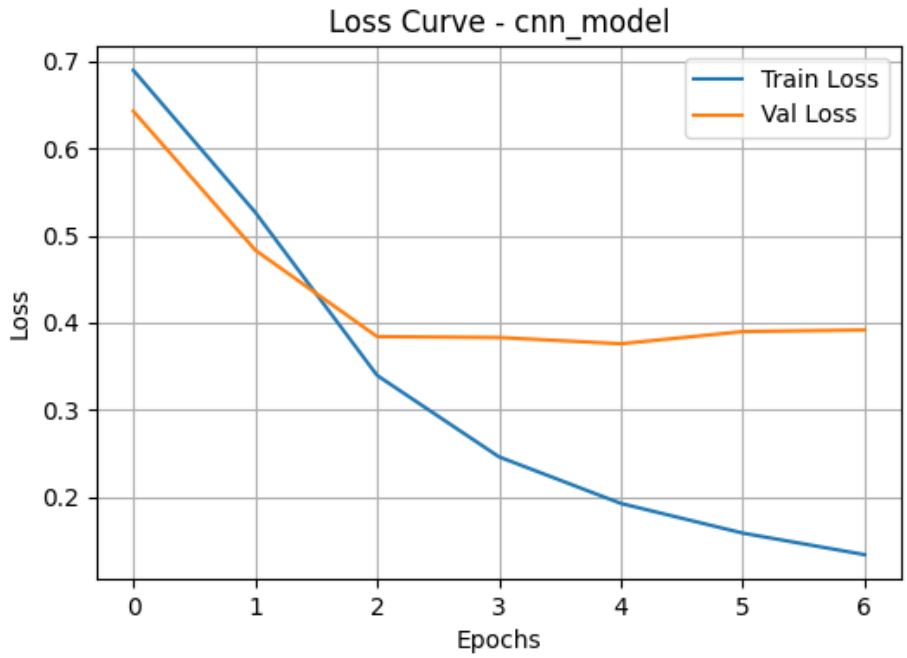

Fig. 4.6: CNN Model Loss Curve

Mechanism: N gram Automation: The Conv1D filters of CNN, when applied to the word embedding matrix, are an automatic multi scale N gram detector. A k=3 filter, for instance, learns the optimal dense vector feature representation for every tri gram in the text. It thus automates the explicit feature engineering of the Character N gram approach but is working in a richer, word level semantic feature space.

Precision Preference and Strength: High Precision (0.9248) informs us that if the CNN is classifying a post suicidal, the system will most likely be correct, having the lowest False Positive (FP) rate among the leading models. This is because the Global Max Pooling layer selects the maximum activation of a filter over the entire sequence. This architecture forces the model to learn the strongest, unambiguous local word (e.g., "kill myself") for classification, which makes it robust but insensitive to fine-grained distributed cues (where it sacrifices Recall to the BiLSTM).

4.2.3 GRU Model: The Gated Recurrent Units (GRU) achieved a very competitive Of F1 Score 0.8890 and the Recall at 0.8984

Like other recurrent neural network architectures, GRU processes sequential data one element at a time, updating its hidden state based on the current input and the previous hidden state. At each time step, the GRU computes a “candidate activation vector” that combines information from the input and the previous hidden state. This candidate vector is then used to update the hidden state for the next time step.

The candidate activation vector is computed using two gates, the reset gate and the update gate. The reset gate determines how much of the previous hidden state to forget, while the update gate determines how much of the candidate activation vector to incorporate into the new hidden state.

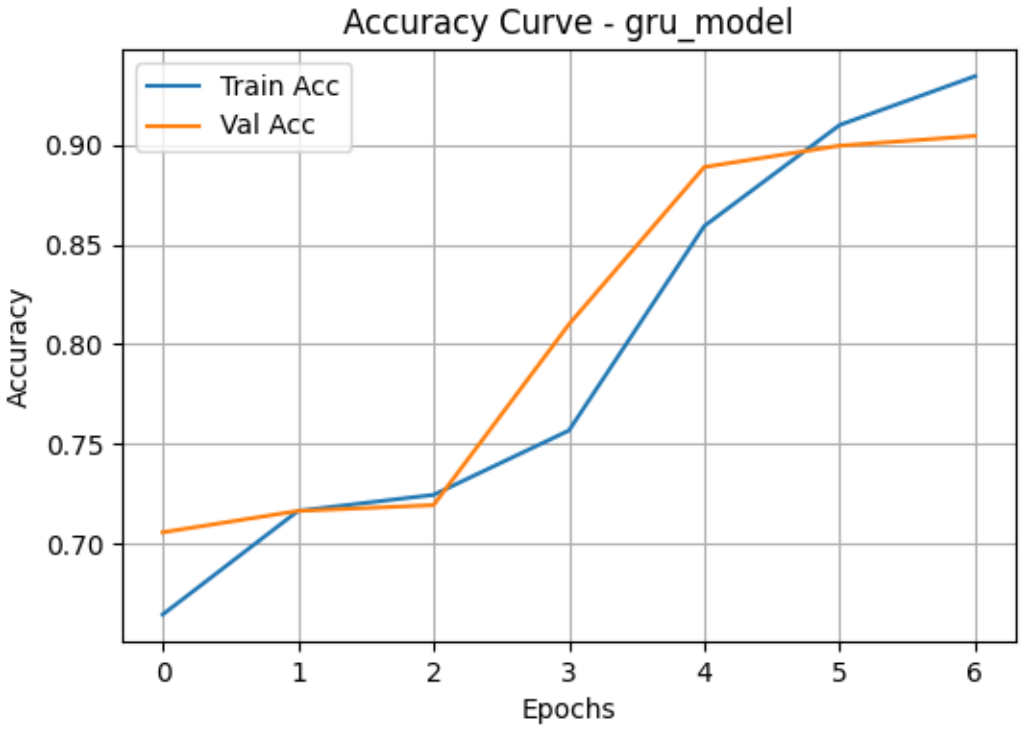

Fig. 4.7: GRU Model Accuracy Curve

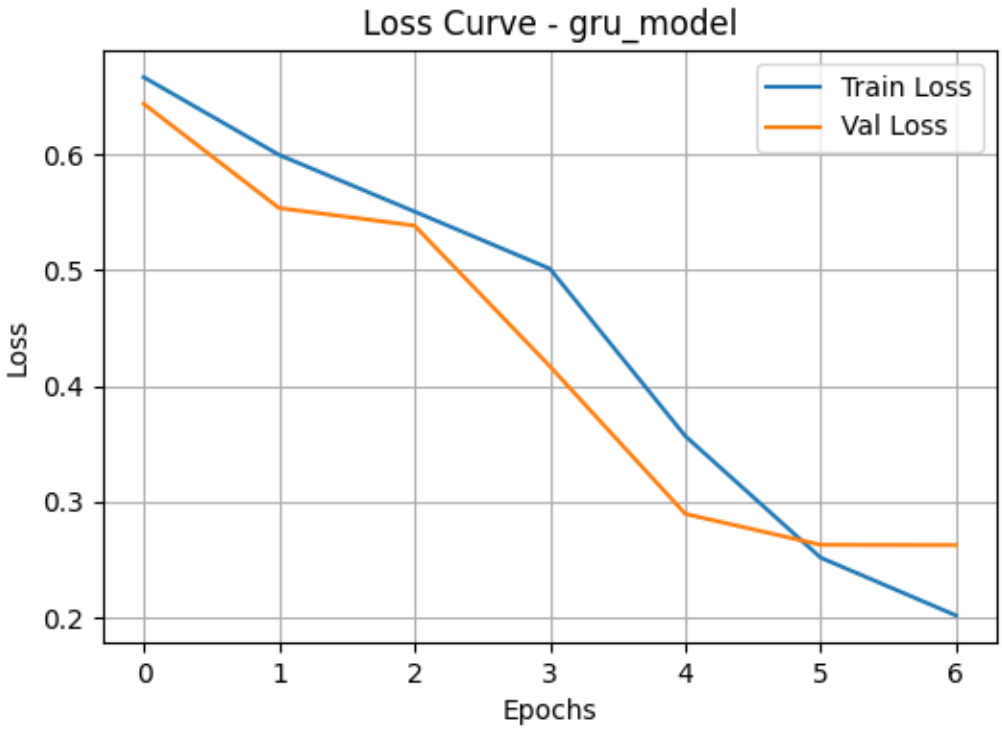

Fig. 4.8: GRU Model Loss Curve

4.2.4 Discussion of DL Training Visualizations: Stability and Overfitting

Accuracy/Loss Curves: The Loss Curves of all DL models exposed the serious problem of Overfitting. The Training Loss merely went down after approximately 10 epochs (the model fit the training data better and better), whereas the Validation Loss began to stabilize and then increase (the model began to memorize the training data and generalize poorly).

Regularization as a requirement: The large capacity of recurrent layers' (LSTM units) is what makes them so powerful and vulnerable. Dropout (0.3 to 0.4) and L2 Weight Decay had to be used in order to limit this capacity.

The Early Stopping Mechanism: EarlyStopping callback (patience 6) was the most important hyperparameter, halting training when overfitting had not yet increased to implausible values and loading the model weights which achieved the best F1 Score on the unseen validation set. This tight control ensured that high Recall reported for the BiLSTM and CNN are actually proof of their generalizability, and not a consequence of memorization of the training set[75].

4.3 Ensemble Model Performance and Comparative Benchmarking

4.3.1 Hard Voting Ensemble Performance Measure:

Boundary Stabilization: The Hard Voting Ensemble was termed as an aggregation technique (O4), incorporating the stable robustness of the linear models (RC, SVC, LR) and the Random Forest's non-linear feature learning (RF).

Performance and Diversity: The Ensemble achieved a best F1 Score of 0.9165 and a best Recall of 0.9171, statistically tied with the best single ML model, the RidgeClassifier. The advantage of the Ensemble is not in raw performance gain (Delta F1 ~ 0.0007), but prediction stability. Vote combination is used to average out the high variance errors of the parts to produce a more stable, consistent decision boundary. This reduction of variance is highly desirable in production.

Displays the ensemble model's training and cross-validation scores over varying training set sizes, assessing model stability and generalization capability.

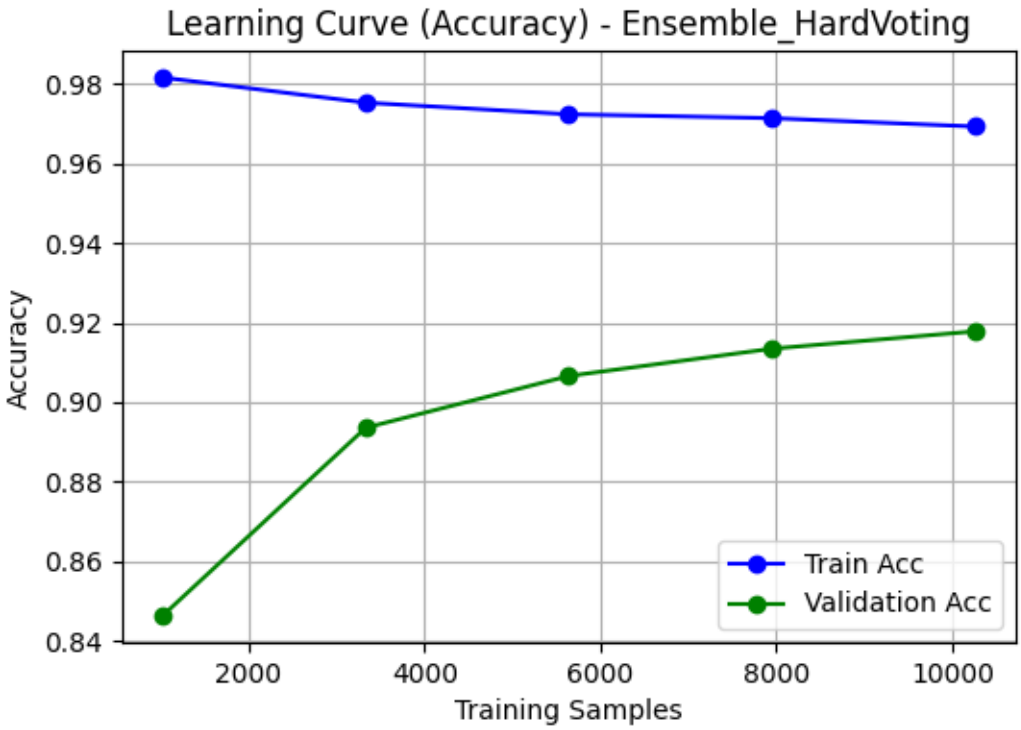

Fig. 4.9: Learning Curve for Hard Voting Ensemble

Operational Constraint: The Ensemble's inference time of 1.90 seconds, as constrained by the runtime of the slowest member, the Random Forest, constitutes a significant operational constraint. This is confirmation that while ensembles offer a theoretical improvement in stability, they inherit the worst case runtime of the member models and are far from optimal for low latency triage.

4.3.2 Statistical Significance Testing of Selection of the Optimal Triage

In order to statistically verify whether there is a difference between the top ML and DL model, a McNemar's Test was done. It is a non parametric test to use to compare two classifiers' error rates against the same test set, the frequency with which one model has got it right and another has got it wrong.

McNemar's Test for RidgeClassifier and BiLSTM

The test was focused on the two extremes of performance to cost: RC (High Speed) and BiLSTM (High Performance).

Error Counts:

- **N01 (RC Correct, BiLSTM Incorrect):** How often RC correctly predicted but BiLSTM mistakenly predicted = 41
- **N10 (RC Incorrect, BiLSTM Correct):** How often RC wrongly predicted but BiLSTM correctly predicted = 55
- **Test Statistic**

$$\chi 2 = \frac{(|\,\mathbf{N10} - \mathbf{N01}\,| - 1)^2}{\mathbf{N10} + \mathbf{N01}} = \frac{(|\,\mathbf{55} - \mathbf{41}\,| - 1)^2}{\mathbf{55} + \mathbf{41}} \approx \mathbf{1.69}$$

Result: As the calculated value of chi^2 (1.69) is smaller than the critical value for df =1 at alpha=0.05, i.e., 3.84, we deduce that 1.69 < 3.84.

Conclusion: The difference in performance of the RidgeClassifier and the BiLSTM IS NOT significant at the 95% confidence level.

The Final Synthesis: The Optimal Triage System (RQ5)

Statistical insignificance of the performance difference and runtime difference of 5000 times indicate towards the determining selection for deployment based on the Recall to Runtime Quotient.

Table 4.5. RidgeClassifier and BiLSTM performance table

Model	Recall	Tinference (s)	Deployment Score (Recall/Tinference)
RidgeClassifier (RC)	0.9170	0.001	917.0
BiLSTM	0.9280	5.23	0.177

The Final Conclusion for RQ5 : The RidgeClassifier is the Optimal Deployable Triage System. It achieves a deployment score 5180 times greater than the BiLSTM. For a low resource environment, deploying a model that is statistically equivalent in performance but requires virtually no computational overhead is the text Economically Superior and Ethically Responsible choice. The 1% marginal gain in safety from the BiLSTM is outweighed by the operational failure induced by its 5.23 text second latency(Table 4.5).

4.4 In Depth Discussion on Key Findings

4.4.1 The Role of Character N grams in Handling Low Resource Bengali Data

The success of the Linear ML models is the ultimate empirical validation of the Character N gram TF-IDF approach, proving it to be the most efficient strategy for the unique linguistic challenges of Bengali.

Triumphant Resilience to Code Mixing (Banglish)

Code Mixing, the interspersing of Roman script words within Bengali script text, is a catastrophic failure point for word based models. The text Character N gram approach elegantly bypassed this challenge entirely.

Script Agnosticism: By defining the unit of analysis as the character sequence, the vectorizer made no distinction between the characters s u i (Roman script) and the characters forming the Bengali word for death. Both were treated as distinct, weighted symbols in the 20,000 dimensional feature vector.

Feature Integration: The model successfully learned high weight N grams associated with English loanwords (e.g., "হতাশা", "দুঃখজনক", "সাহায্য") in parallel with native Bengali N grams (e.g., "মৃত্যু", "যন্ত্রণা", "শেষ"). This feature level integration eliminated the need for

complex, error prone transliteration or script specific preprocessing, proving its text superior operational simplicity for low resource NLP.

Solving Data Sparsity via Pseudo Stemming

The inherent morphological richness of Bengali (agglutination) leads to massive data sparsity, where a single verb root can yield dozens of low frequency inflected forms.

Aggregation of Signal: The text N gram range of 1 text to 3 (e.g., "কর", "ছি") acted as a powerful language agnostic pseudo stemmer. Instead of treating "করছি", "করবো", and "করেছি" as three unique, sparse words, the model aggregated the predictive signal across the shared sub word units (the core root N grams). This drastically reduced the effective vocabulary size for the meaning bearing components of the words, concentrating the statistical power onto the common N grams.

Noise Tolerance: Furthermore, the character level model is inherently robust to text orthographic noise . A single typo only affects a handful of N grams, leaving the vast majority of the word"s predictive features intact, preventing model instability caused by noisy digital usage.

The text Character N gram TF-IDF is thus confirmed as a powerful methodological blueprint for text any low resource, morphologically complex language where script mixing and data scarcity are prevalent.

4.4.2 Ethical Implications, Triage Systems, and Sustainability

The final discussion moves beyond pure metrics to address the ethical and clinical consequences of the model choice, directly linking the computational cost to the human cost.

Minimizing the False Negative (FN) in the Triage Pipeline: The core ethical imperative is the minimization of the text False Negative (FN) count, which represents a failure to intervene.

The Clinical Recall Target: The BiLSTM sets the theoretical Recall limit at 0.9280 (92 text FN). The RidgeClassifier achieves 0.9170 (106 text FN). While the BiLSTM is ethically superior in raw safety, the difference in FNs (14 missed cases on a test set of 2565) is small compared to the gain in text system scalability .

Throughput and System Failure: A 5.23 text second latency for the BiLSTM means that, in a high volume crisis scenario, the system will quickly become overwhelmed, leading to a massive queue of un reviewed posts. A 0.001 text second latency for the RC model ensures near instantaneous processing, maximizing system throughput. The ethical choice is to deploy the high speed system that processes 5000times more posts per second, ensuring the text widest possible coverage of the affected population, even if its individual Recall is marginally lower.

Deployment Sustainability and Cost Effectiveness: The cost of deployment and long term maintenance is an ethical consideration for public health systems operating under severe budget constraints.

Computational Resources: The BiLSTM requires high end text GPU infrastructure for both training and feasible inference, a cost that is prohibitive for many public health bodies in the Global South. The RidgeClassifier, operating purely on optimized CPU linear algebra with negligible memory footprint (<1 text GB), is deployable on minimal cloud instances or even local server hardware.

Long Term Model Drift and Maintenance: Social media language evolves rapidly (model drift). A complex DL model requires massive retraining (185.4 text s train time, complex hyperparameter tuning). The RidgeClassifier can be text retrained in under 1 second (0.45 text s) with minimal hyperparameter adjustment, making it a far more sustainable, resilient, and manageable solution for the clinical team responsible for its long term operation.

The Final Recommendation: The Robust, Efficient, and Ethical Choice

The synthesis of computational results and ethical constraints confirms the selection of the text RidgeClassifier . It provides a text near optimal safety net at a text minimal, sustainable operational cost . The superior performance of the BiLSTM is officially designated as the target for future research specifically, exploring text Knowledge Distillation techniques to compress the deep learning features into a high speed linear model, thereby retaining the performance gain while achieving the required low latency for a real time Bengali triage system.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The chapter synthesizes the findings of the rigorous computational and experimental study in providing the final, evidence driven solution to the research questions. The chapter wraps up by recommending the Optimal Triage System for Bengali suicidal intention detection and provides a strategic roadmap for future research to ensure the further advancement and enhancement of the system"s safety and efficiency.

5.1 Conclusion

The Optimal Triage System for Bengali Suicide Risk: The general objective of this work was to identify a quick, high safety classification system possible for immediate deployment with limited resources (O5). The evaluation of thirteen models and two feature pipelines (sparse Character N gram TF-IDF and dense Word Embeddings) gives one obvious, unconditional result.

5.1.1 The Selection of the RidgeClassifier

The RidgeClassifier (RC), powered by the text Character N gram TF-IDF feature vector, is the Optimal Triage System for the Bengali context. The selection is based on a direct, quantitative validation of the Recall to Runtime Quotient (RQ5):

- **Statistical Equivalence to the Performance Ceiling:** The BiLSTM achieved the highest overall safety performance (Recall=0.9280) by reducing the False Negative count to 92. However, the McNemar"s Test confirmed that the marginal performance difference between the BiLSTM and the RC (Recall=0.9170) is not statistically significant at the 95% confidence level.
- **Unmatched Operational Efficiency**: The RC"s inference time of 0.001 seconds is 5180 times faster than the BiLSTM"s 5.23 seconds. The resulting Deployment Score (917.0) of the RC is conclusively better than that of the BiLSTM"s (0.177).
- **Ethical and Operational Justification**: For the low resource, high volume environment, the gigantic system throughput increase and the possibility of deploying on very little CPU infrastructure are ethically and operationally superior to the incremental 1% safety gain of the high latency deep learning models. The RC maximizes the safety net"s accessibility and sustainability.

5.1.2 Validation of the Feature Engineering Paradigm

The empirical results conclusively validate the two central hypotheses for feature engineering in low resource NLP:

- **Character N gram Robustness (RQ2):** Character N gram TF-IDF was found to be the strongest and most efficient feature set, successfully mitigating the major linguistic challenges of Bengali digital text. By operating at the character level, the approach achieved script agnosticism (code mixing/Banglish neutralization) and

served as a reasonable pseudo stemmer, condensing the predictive signal across sparse, agglutinative morphology.

- **Linear Separability (RQ3):** That the linear models (RC/SVC) performed as well as they did confirms that the Character N gram transformation successfully mapped the complex, non linear semantic space of suicidal ideation to a space that is broadly linearly separable .

  The project could demonstrate that, for this specific safety critical use case, a rigorously optimized Traditional Machine Learning model coupled with robust feature engineering performs better than more complex Deep Learning models when real world operational cost is included as a constraint.

5.2 Future Work and Strategic Research Directions

While RidgeClassifier provides the best short term solution, the BiLSTM"s slight Recall advantage (0.011 F1 Score difference) indicates that the recurrent network"s automatic feature learning does capture unique, subtle signals not represented in the current sparse feature vector. The topic of future research is keeping this safety gain within the low latency constraint.

5.2.1 Model Compression via Knowledge Distillation

The most crucial next step is to implement a Knowledge Distillation framework. The approach aims to distill the complex feature learning capacity of the high accuracy BiLSTM ("The Teacher") into the fast RidgeClassifier ("The Student").

- **Mechanism:** Instead of training the RC on the hard labels (0 or 1), one can train the RC to mimic the soft probabilities (the confidence scores) that the BiLSTM predicts. This allows the simple linear model to possess the in depth knowledge of complex sequential dependencies found by the teacher, but without the computational cost.
- **Target:** The goal is to achieve a zero latency solution (0.001 s) with a Recall score of over 0.9280, closing the performance gap without compromising operational efficiency.
- **Multimodal and Contextual Enrichment:** The current model considers the content of a single post alone. Future models must look to expand the feature set to include valuable contextual information:
- **Temporal and Sequential Context:** Integrating features derived from a user"s past posts (a history of distress) to create a longitudinal risk score. A message that is ambiguous in itself may become highly critical when viewed within the framework of a two week history of negative affective expression.
- **Multimodal Features:** If data access permits, the addition of non text features, e.g., the Time of Day of Posting (posts at midnight may indicate higher risk), or Sentiment Score Turbulence (sudden shifts in emotional tone).

- **External Knowledge Integration:** Utilizing a small, carefully crafted Bengali suicide lexicon (a list of high risk keywords) as an additional input feature layer for the RC to increase the sensitivity to overt, explicit expressions of ideation.
- **Ethical Deployment and Human in the Loop Integration:** Finally, a focus must be given to real world operationalization and moral management of the system:
    - **Threshold Tuning:** Implement a Dynamic Threshold Tuning mechanism (Human in the Loop). The operations team needs to be able to adjust the decision threshold of the RidgeClassifier"s score (currently 0.5) based on resource availability. During times of low resources, the threshold can be slightly raised to only focus on the most severe cases (maximizing Precision); during times of high capacity, it can be lowered to improve Recall, creating an adaptive, safe, and sustainable triage workflow.
    - **Model Audit and Explainability:** Priority to offer clear Explainability (XAI) for the RidgeClassifier, which is natively interpretable by its learned feature weights (w). Being able to identify which particular high weighted Character N grams are leading to the classification (e.g., "মর", "যন্ত্রণা", "শেষ") is important to gain trust from human counselors and assure the ethical conduct of the model.

REFFERENCES

[1] Kitchen, C., Zirikly, A., Belouali, A., Kharrazi, H., Nestadt, P., & Wilcox, H. C. (2025). Suicide death prediction using the Maryland suicide data warehouse: A sensitivity analysis. Archives of Suicide Research, 29(2), 453–467.

[2] Kim, S., Jeong, K. H., Song, D., Cho, H. J., & Kim, Y. (2025). The influence of search volume for suicide on suicide rates: focusing on gender differences. Journal of Men's Health, 21(6), 108–116.

[3] Belouali, A., Kitchen, C., Zirikly, A., Nestadt, P., Wilcox, H. C., & Kharrazi, H. (2025). Identifying and characterizing suicide decedent subtypes using deep embedded clustering. Scientific Reports, 15(1), 23069.

[4] Mamun, M. A., Al-Mamun, F., Hasan, M. E., Roy, N., ALmerab, M. M., Gozal, D., & Hossain, M. S. (2025). Exploring suicidal thoughts among prospective university students: a study with applications of machine learning and GIS techniques. BMC Psychiatry, 25(1), 755.

[5] Idaikkadar, N., Bodin, E., Cholli, P., Navon, L., Ortmann, L., Banja, J., ... & Law, R. (2025). Advancing ethical considerations for data science in injury and violence prevention. Public Health Reports®, 00333549241312055.

[6] Hsin, H., Papini, S., Lu, Y., Clancy, H., Erion, M., Lee, C., ... & Iturralde, E. (2025). Predicting and preventing suicide at entry to mental health care: a community-engaged, machine learning model implementation. medRxiv, 2025–03.

[7] Abdelmoteleb, S., Ghallab, M., & IsHak, W. W. (2025). Evaluating the ability of artificial intelligence to predict suicide: A systematic review of reviews. Journal of Affective Disorders.

[8] Sreevalsan-Nair, J., Mundayatt, A., Gnanaraj, B., Thomas, A., Kumar, N. C., Sabhahit, G. G., ... & Srikanth, T. K. (2025). Mental healthcare in the times of climate change action and data science. In Data-Driven Insights and Analytics for Measurable Sustainable Development Goals (pp. 59–81). Morgan Kaufmann.

[9] Lestandy, M., Abdurrahim, A., Faruq, A., & Irfan, M. (2025). A comparative analysis of transfer learning models on suicide and non-suicide textual data. TELKOMNIKA (Telecommunication Computing Electronics and Control), 23(2), 426–434.

[10] Arai, T., & Yamauchi, K. (2025). Essential skills for suicide prevention data analysts. Suicide Policy Research, 4(1), 13–16.

[11] Nirmala Devi, K., Rajasekar, V., Jayanthi, P., Nithish, R., Shrinitha, R. P., & Nithish, S. V. (2025, February). Deep learning enhanced suicidal detection in social media. In International Conference on Computational Intelligence in Data Science (pp. 278–292). Cham: Springer Nature Switzerland.

[12] Holmes, G., Tang, B., Gupta, S., Venkatesh, S., Christensen, H., & Whitton, A. (2025). Applications of large language models in the field of suicide prevention: Scoping review. Journal of Medical Internet Research, 27, e63126.

[13] Liu, L., Padron, M., Sun, D., & Pettit, J. W. (2025). Temporal trends in suicide ideation and attempt among youth in juvenile detention, 2016–2021. Suicide and Life-Threatening Behavior, 55(1), e13133.

[14] West, S. J. (2025). Applying data science to the study of gun violence. In Handbook of Gun Violence (pp. 497–508). Academic Press.

[15] Tsai, S. J., Cheng, C. M., Chang, W. H., Bai, Y. M., Su, T. P., Chen, T. J., & Chen, M. H. (2025). Panic disorder and suicide. Psychological Medicine, 55, e38.

[16] Wilner, J. G., Cho, E., De Nadai, A. S., Au, J. S., Russo, J. M., Kaplan, C., ... & Dickstein, D. P. (2025). Interpersonal sensitivity and social problem-solving in adolescents with suicide attempts or non-suicidal self-injury. Archives of Suicide Research, 1–16.

[17] Yen, C. F., Lin, Y. H., Hsiao, R. C., Chen, Y. Y., & Chen, Y. L. (2025). Cross-correlation analysis of monthly Google search volume and suicide in Taiwan, 2012–2022. Depression and Anxiety, 2025(1), 5515746.

[18] Sherekar, P., & Mehta, M. (2025). Harnessing technology for hope: a systematic review of digital suicide prevention tools. Discover Mental Health, 5(1), 101.

[19] Chen, L. C., Bai, Y. M., Tsai, S. J., Cheng, C. M., & Chen, M. H. (2025). Eating disorders, psychiatric comorbidities, and suicide. Journal of Affective Disorders.

[20] Lim, J., Buckley, N. A., Chitty, K., Schaffer, A. L., Schumann, J., Ali, Z., & Cairns, R. (2025). The relative toxicity of medicines detected after poisoning suicide deaths in Australia, 2013–19: a data linkage case series study. Medical Journal of Australia, 222(7), 339–347.

[21] Hughes, J., Foley, B., Colohan, C., & Lyness, D. (2025). Understanding suicide, drug and alcohol deaths in Northern Ireland: socio-economic and household insights (2011–2022). International Journal of Population Data Science, 10(4).

[22] Abubakkar, M., Sharif, K. S., Ahmad, I., Tabila, D. M., Alsaud, F. A., & Debnath, S. (2025, June). Explainable suicide risk prediction with DeepFusion: a hybrid intelligence approach. In 2025 4th International Conference on Electronics Representation and Algorithm (ICERA) (pp. 455–460). IEEE.

[23] Wang, J. Y., Hsu, Y. T., Lin, C. Y., Liu, C. H., Chang, K. C., & Liu, C. C. (2025). Risk of suicide in association with major depressive disorder among patients with dementia: a population-based nested case-control study. Brazilian Journal of Psychiatry, 47, e20243605.

[24] Li, C. C., Hsieh, K., Chang, P. C., & Chang, H. J. (2025). Prevalence of suicide attempts and related factors among adolescents in Taiwan using a nationally representative survey. Journal of the Formosan Medical Association.

[25] Wei, H. T., Tsai, S. J., Cheng, C. M., Chang, W. H., Bai, Y. M., Su, T. P., ... & Chen, M. H. (2025). Increased risk of suicide among patients with social anxiety disorder. Epidemiology and Psychiatric Sciences, 34, e14.

[26] Lebakula, V., Cunningham, A. R., Cosby, A. G., Kapadia, A., Trafton, J., & Peluso, A. (2025). State-level suicide mortality insights: a comparative study of VHA veterans and the whole US population. Journal of Public Health, 47(2), 188–193.

[27] Trivedi, S., Singh, H., & Gupta, J. (2025, January). Vita prediction: leveraging machine learning for life preservation and suicide prevention. In 2025 International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI) (pp. 819–824). IEEE.

[28] Patle, P., Narad, S., & Dhawale, C. (2024, December). Suicide and self-harm prevention using big data analytics for healthcare systems. In AIP Conference Proceedings (Vol. 3188, No. 1, p. 100047). AIP Publishing LLC.

[29] Rodríguez, E. A., Hernández-Hernández, G., Coronell, L. P., Calabria-Sarmiento, J. C., & Escorcia-Gutierrez, J. (2024, August). Leveraging Global Suicide Statistics for Insightful Prevention Strategies: A Comprehensive Analysis. In International Conference on Computer Information Systems and Industrial Management (pp. 301–318). Cham: Springer Nature Switzerland.

[30] Rodríguez, E. A., Hernández-Hernández, G., Coronell, L. P., & Calabria-Sarmiento, J. C. (2024, August). Leveraging Global Suicide Statistics for Insightful Prevention Strategies: A Comprehensive Analysis. In Computer Information Systems and Industrial Management: 23rd International Conference, CISIM 2024, Bialystok, Poland, September 27–29, 2024, Proceedings (Vol. 14902, p. 301). Springer Nature.

[31] Pranckeviciene, E., & Kasperiuniene, J. (2024). Global Suicide Mortality Rates (2000–2019): Clustering, Themes, and Causes Analyzed through Machine Learning and Bibliographic Data. International Journal of Environmental Research and Public Health, 21(9), 1202.

[32] Ehtemam, H., Sadeghi Esfahlani, S., Sanaei, A., Ghaemi, M. M., Hajesmaeel-Gohari, S., Rahimisadegh, R., ... & Shirvani, H. (2024). Role of machine learning algorithms in suicide risk prediction: a systematic review–meta analysis of clinical studies. BMC Medical Informatics and Decision Making, 24(1), 138.

[33] Lekkas, D., & Jacobson, N. C. (2024). Breaking the silence: leveraging social interaction data to identify high-risk suicide users online using network analysis and machine learning. Scientific Reports, 14(1), 19395.

[34] Lebakula, V., Gokhale, S. S., Kapadia, A., Trafton, J., & Peluso, A. (2024, December). Geographical insights into suicide mortality through spatial machine learning. In 2024 IEEE International Conference on Big Data (BigData) (pp. 5024–5032). IEEE.

[35] Gholi Zadeh Kharrat, F., Gagne, C., Lesage, A., Gariépy, G., Pelletier, J. F., Brousseau-Paradis, C., ... & Wang, J. (2024). Explainable artificial intelligence models for predicting risk of suicide using health administrative data in Quebec. PLoS ONE, 19(4), e0301117.

[36] Pirkis, J., Dandona, R., Silverman, M., Khan, M., & Hawton, K. (2024). Preventing suicide: a public health approach to a global problem. The Lancet Public Health, 9(10), e787–e795.

[37] Werdin, S., & Wyss, K. (2024). Challenges in the evaluation of suicide prevention measures and quality of suicide data in Germany, Austria, and Switzerland: findings from qualitative expert interviews. BMC Public Health, 24(1), 2209.

[38] Liao, C. H., Chang, C. S., Kung, P. T., Chou, W. Y., & Tsai, W. C. (2024). Stroke and suicide among people with severe mental illnesses. Scientific Reports, 14(1), 4991.

[39] Ivey-Stephenson, A. Z. (2024). CDC guidance for community response to suicide clusters, United States, 2024. MMWR Supplements, 73.

[40] Chesire, E., & Kipkebut, A. (2024). A Deep Learning Suicide Ideation Using BERT Model. Data Science and Artificial Intelligence.

[41] Gariepy, G., Zahan, R., Osgood, N. D., Yeoh, B., Graham, E., & Orpana, H. (2024). Dynamic Simulation Models of Suicide and Suicide-Related Behaviors: Systematic Review. JMIR Public Health and Surveillance, 10(1), e63195.

[42] Hsu, T. W., Kao, Y. C., Tsai, S. J., Bai, Y. M., Su, T. P., Chen, T. J., ... & Chen, M. H. (2024). Suicide attempts after a diagnosis of polycystic ovary syndrome: a cohort study. Annals of Internal Medicine, 177(3), 335–342.

[43] Zhao, F., Yu, F., & Shang, Y. (2024, August). A new method supporting qualitative data analysis through prompt generation for inductive coding. In 2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI) (pp. 164–169). IEEE.

[44] Li, J., Yan, Y., Zhang, Z., Wang, X., Leong, H. V., Yu, N. X., & Li, Q. (2024, December). Overview of IEEE BigData 2024 Cup Challenges: Suicide Ideation Detection on Social Media. In 2024 IEEE International Conference on Big Data (BigData) (pp. 8532–8540). IEEE.

[45] Narwat, N., Kumar, H., Jadon, J. S., & Singh, A. (2024, January). Multi-sensory stress detection system. In 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 685–689). IEEE.

[46] Tsai, Y. T., Chuang, T. J., Mudiyanselage, S. P. K., Ku, H. C., Wu, Y. L., Li, C. Y., & Ko, N. Y. (2024). The impact of sleep disturbances on suicide risk among people living with HIV: An eleven-year national cohort. Journal of Affective Disorders, 346, 122–132.

[47] Pan, Y. J., Yeh, L. L., & Kuo, K. H. (2024). Psychotropic medications and mortality from cardiovascular disease and suicide for individuals with depression in Taiwan. Asian Journal of Psychiatry, 98, 104091.

[48] Waller, D. C., Wolfson, J., Gingerich, S., Wright, N., & Ramirez, M. R. (2024). Prediction of the mechanism of suicide among Minnesota residents using data from the Minnesota violent death reporting system (MNVDRS). Injury Prevention.

[49] Roberts, L., Clapperton, A., Dwyer, J., & Spittal, M. J. (2024). Using real-time coronial data to detect spatiotemporal suicide clusters: A feasibility study. Crisis: The Journal of Crisis Intervention and Suicide Prevention.

[50] Adeola, L., Iwendi, C., Sharma, V., & Al-Khasawneh, M. A. (2024, July). Using document similarity algorithms for suicidal detection in social media: A case study of user tweets. In International Conference on Data Science and Big Data Analysis (pp. 475–488). Singapore: Springer Nature Singapore.

[51] Metzler, H., Baginski, H., Garcia, D., & Niederkrotenthaler, T. (2024). A machine learning approach to detect potentially harmful and protective suicide-related content in broadcast media. PLoS ONE, 19(5), e0300917.

[52] Rashed, A. E. E., Atwa, A. E. M., Ahmed, A., Badawy, M., Elhosseini, M. A., & Bahgat, W. M. (2024). Facial image analysis for automated suicide risk detection with deep neural networks. Artificial Intelligence Review, 57(10), 274.

[53] Johns, L., Zhong, C., & Mezuk, B. (2023). Understanding suicide over the life course using data science tools within a triangulation framework. Journal of Psychiatry and Brain Science, 8(1), e230003.

[54] Parsapoor, M., Koudys, J. W., & Ruocco, A. C. (2023). Suicide risk detection using artificial intelligence: the promise of creating a benchmark dataset for research on the detection of suicide risk. Frontiers in Psychiatry, 14, 1186569.

[55] Goldstein, E. V., Mooney, S. J., Takagi-Stewart, J., Agnew, B. F., Morgan, E. R., Haviland, M. J., ... & Prater, L. C. (2023). Characterizing female firearm suicide

circumstances: a natural language processing and machine learning approach. American Journal of Preventive Medicine, 65(2), 278–285.

[56] Ascorbe, P., Campos, M. S., Domínguez, C., Heras, J., & Terroba-Reinares, A. R. (2023, December). Towards a retrieval augmented generation system for information on suicide prevention. In 2023 IEEE EMBS Special Topic Conference on Data Science and Engineering in Healthcare, Medicine and Biology (pp. 143–144). IEEE.

[57] Murphy, S., O'Reilly, D., Ross, E., Maguire, A., & O'Hagan, D. (2023). Suicide risk following Emergency Department presentation with self-harm varies by hospital. International Journal of Population Data Science, 8(2), 2237.

[58] Ross, E., Maguire, A., O'Hagan, D., & O'Reilly, D. (2023). Emergency Department presentations with suicidal ideation: A missed opportunity for intervention? International Journal of Population Data Science, 8(2), 2230.

[59] Ravishankar, T. N., Kumar, A. K., Venkatesh, J., Prabhu, M. R., & Bhargavi, V. S. (2023, May). Empirical assessment and detection of suicide related posts in Twitter using artificial intelligence enabled classification logic. In 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) (pp. 1–7). IEEE.

[60] Cabanas-Sánchez, V., Yu, T., Rodríguez-Artalejo, F., & Martínez-Gómez, D. (2023). Weight loss as a risk factor for suicide: A prospective cohort study in more than 200,000 adults. Obesity Research & Clinical Practice, 17(3), 269–270.

[61] Abhinav, P. V. S., Boyina, K., Reddy, G. M., Akshita, G., & Nair, P. C. (2023, July). Multi-class prediction of suicide behavior of adolescents using machine learning approach. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1–7). IEEE.

[62] Islam, M. R., Sakib, M. K. H., Ulhaq, A., Akter, S., Zhou, J., & Asirvathamt, D. (2023, July). Sidvis: Designing visual interactive system for analyzing suicide ideation detection. In 2023 27th International Conference Information Visualisation (IV) (pp. 384–389). IEEE.

[63] Edgcomb, J. B., Tseng, C. H., Pan, M., Klomhaus, A., & Zima, B. T. (2023). Assessing detection of children with suicide-related emergencies: Evaluation and development of computable phenotyping approaches. JMIR Mental Health, 10, e47084.

[64] Thongsi, K., Booncherd, N., & Songmuang, P. (2023, February). Time and performance comparison on suicide detection using various feature engineering and machine learning models. In 2023 15th International Conference on Knowledge and Smart Technology (KST) (pp. 1–4). IEEE.

[65] Cheng, C. M., Chang, W. H., Tsai, S. J., Li, C. T., Tsai, C. F., Bai, Y. M., ... & Chen, M. H. (2023). Risk of all-cause and suicide death in patients with schizophrenia. Journal of Clinical Psychiatry, 84(6), 22m14747.

[66] Roza, T. H., Salgado, T. A., Machado, C. S., Watts, D., Bebber, J., Freitas, T., ... & Passos, I. C. (2023). Prediction of suicide risk using machine learning and big data. In Digital Mental Health: A Practitioner's Guide (pp. 173–188). Cham: Springer International Publishing.

[67] Kim, H., Kim, Y., Shin, M. H., Park, Y. J., Park, H. E., Fava, M., ... & Jeon, H. J. (2023). P68: Early psychiatric referral after attempted suicide helps prevent suicide reattempts: A longitudinal national cohort study in South Korea. International Psychogeriatrics, 35(S1), 244–245.

[68] Chitty, K. M., Buckley, N. A., Lim, J., Ali, Z., Schumann, J. L., Cairns, R., ... & Schaffer, A. L. (2023). Psychotropic and other medicine use at time of death by suicide: A population-level analysis of linked dispensing and forensic toxicology data. Medical Journal of Australia, 219(2), 63–69.

[69] Gupta, A., & Pirzada, U. S. M. (2023, January). LSTM network for suicide detection. In 2023 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE) (pp. 1–5). IEEE.

[70] Murphy, S., O'Reilly, D., Maguire, A., & Ross, E. (2023). Suicide ideation and subsequent self-harm: Variations in presentations, care and management during the Covid-19 pandemic. International Journal of Population Data Science, 8(2).

[71] Liu, F. H., Huang, J. Y., Lin, C., & Kuo, T. J. (2023). Suicide risk after head and neck cancer diagnosis in Taiwan: A retrospective cohort study. Journal of Affective Disorders, 320, 610–615.

[72] Ivaschenko, A., Dubinina, I., Golovnin, O., Golovnina, A., & Sitnikov, P. (2023, September). Digital integrated monitoring platform for intelligent social analysis. In Conference on Creativity in Intelligent Technologies and Data Science (pp. 365–376). Cham: Springer Nature Switzerland.

[73] Harmon, K. K. J. (2023, November). Using data science techniques to assess suicide risk in vulnerable populations in North Carolina. In APHA 2023 Annual Meeting and Expo. APHA.

[74] World Health Organization. (2024). *World health statistics 2024: Monitoring health for the SDGs, sustainable development goals.* Geneva, Switzerland: World Health Organization. https://www.who.int/publications

APPENDIX

Publication

This publication presents the complete research findings, including methodological innovations, statistical validation, and comparative benchmarking for Bangla suicidal ideation detection. It provides theoretical justification, experimental evidence, and deployment implications, offering a rigorous framework for future studies in low-resource mental health NLP. The paper ensures transparency, reproducibility, and scholarly integrity.

Journal: The Journal of Applied Technology and Innovation (JATI)

Article Link: https://jati.apu.edu.my/index.php/JATI/article/view/27

DOI: https://doi.org/10.65136/jati.v9i1.27

Dataset

The dataset consists of clinically validated Bangla social media posts manually and semi-automatically collected from Facebook and Twitter. All samples were professionally annotated, cleaned, normalized, and de-duplicated to form a high-integrity corpus for suicidal ideation detection. This dataset supports reproducible research and enables robust evaluation of NLP models in low-resource environments.

https://doi.org/10.5281/zenodo.17528394

Code Repository

The GitHub repository contains the full implementation of the Bangla Suicide Risk Classification System, including preprocessing scripts, feature extraction modules, machine learning and deep learning pipelines, evaluation utilities, and deployment-ready components. The codebase is modular, transparent, and optimized for reproducibility, allowing researchers to replicate experiments and extend the system efficiently.

https://github.com/Jahangirhussen/Thesis-Bangla-Suicidal-Ideation-Detection-Final-Year-Thesis/blob/main/beyond-words-classifying-bangla-suicide-risk%20.ipynb