UNSUPERVISED TEXTUAL SARCASM DETECTION USING OPTIMIZATION TECHNIQUES

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

in

Mathematics

by

HIMANI POKHRIYAL
(Roll No. 2K21/PHDAM/07)

Under the Supervision of
Dr. Goonjan Jain
Delhi Technological University

Department of Applied Mathematics
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, New Delhi-110042, India

December, 2025

DELHI TECHNOLOGICAL UNIVERSITY

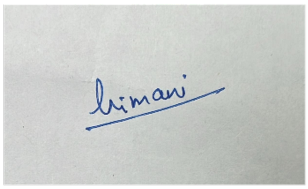
(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I **Himani Pokhriyal** hereby certify that the work which is being presented in the thesis entitled "**Unsupervised Textual Sarcasm Detection Using Optimization Techniques**" in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy, submitted in the Department of Applied Mathematics, Delhi Technological University is an authentic record of my own work carried out during the period from August 2021 to September 2025 under the supervision of Dr. Goonjan Jain.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.



Himani Pokhriyal

Roll No. 2K21/PHDAM/07

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that **Himani Pokhriyal** (Enrollment no: 2K21/PHDAM/07) has carried out her research work presented in this thesis entitled **"Unsupervised Textual Sarcasm Detection Using Optimization Techniques"** for the award of **Doctor of Philosophy** from the Department of Applied Mathematics, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student herself, and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Dr. Goonjan Jain
Assistant Professor
Department of Applied Mathematics,
Delhi Technological University, Delhi

Date:

ABSTRACT

Sarcasm detection is a form of figure of speech that conveys the opposite of its literal meaning, often to express insult, wit, irritation, or ridicule. In text, sarcasm is typically conveyed through positive or intensified positive words to mask negative feelings. With the rise of social media platforms like Twitter, Facebook, and WhatsApp, posting sarcastic messages has become a common way to avoid direct negativity. However, detecting these indirect negativities is crucial as they significantly impact businesses. The challenge in analysing sarcasm lies in the gap between its literal and intended meanings.

Despite extensive research in natural language processing (NLP) and sarcasm detection, there is a notable lack of comparative analysis among different NLP techniques and their ability to correctly classify sarcastic content. Additionally, there is a scarcity of studies on using mathematical optimization techniques for sarcasm detection and a neglect of the intonation and tonal traits of sarcasm.

This thesis addresses these gaps by introducing frameworks that integrate mathematical optimization techniques with NLP models. These frameworks generate robust algorithms for detecting sarcasm and its inherent tonal nature. We utilize sentence scores from sentiment lexicon models and apply mathematical optimization techniques to identify sarcasm in social media comments. The thesis includes binary and tertiary classification of social media comments across various domains. It also presents a model for detecting sarcasm in Hindi comments, demonstrating that these mathematical optimization techniques can be adapted to any language with minor modifications. We have specifically focused on incorporating the tonal traits of sarcasm into sentiment analysis.

The primary objective is to expand knowledge in this area and provide new perspectives on the strengths and weaknesses of the proposed models. This research aims to contribute to both the academic community and companies that develop or use this technology. Our study employs a qualitative approach supported by quantitative data. An extensive literature review was conducted to deepen our understanding of the field. Benchmark datasets were used for analysis, and the results form the basis for evaluating the selected models ability to identify sarcasm based on metrics such as accuracy, precision, recall, and F1 score. The results indicate that the proposed mathematical optimization-based models are effective for classifying and detecting sarcasm. These models offer efficient, scalable, and accurate solutions for analysing written reviews by leveraging mathematical optimization techniques.

In summary, our novel unsupervised sarcasm detection methods provide effective solutions to the challenges posed by large amounts of online data and the resource-intensive nature of conventional machine learning approaches. By utilizing mathematical optimization models, we ensure logical and consistent outcomes, thereby enhancing confidence in the accuracy of sarcasm classifications. These models are designed to be efficient, scalable, and accurate in detecting sarcasm in written contexts.

PUBLICATIONS RELATED TO THE THESIS

Journal Papers Published

- Pokhriyal, Himani, and Jain, Goonjan. "Supposititious Sarcasm Detection and Sentiment Analysis Coping Hindi Language in Social Networks Harnessing Zipf-Mandelbrot Probabilistic Optimisation and Perplexity Entropy Learning." *ACM Transactions on Asian and Low-Resource Language Information Processing*. vol. 24, Issue 2, no. 17, pp. 1-28, 2025, DOI: https://doi.org/10.1145/3712061
- Pokhriyal, Himani, and Jain, Goonjan. "Sarcasm Detection with Induced Sentimental Cues Using Heuristic Search Based on Unconstrained Optimisation Learning Quantifying Callousness on Social Media." *Neurocomputing*. vol. 645, pp. 130499, 2025, DOI: https://doi.org/10.1016/j.neucom.2025.130499

Journal Paper Communicated

- Pokhriyal, Himani and Jain, Goonjan. "SOBH-REB: Acknowledging Sarcasm Detection Using Unified Cognition of Combinatorial Optimisation and Entropy Distribution Mechanism." **(Under Review)**
- Pokhriyal, Himani and Jain, Goonjan. "Examining Social Media Comments through Sarcasm Detection with Intonation Analysis using Unified Semantic Modelling Logistic-Ions Motion Optimization." **(Under Review)**
- Pokhriyal, Himani and Jain, Goonjan. "Sarcasm Detection Deploying Statistical Optimisation Learning of Semantic Analytics in Online Web." **(Under Review)**
- Pokhriyal, Himani and Jain, Goonjan. "SD-GKOT: Sentiment Induced Sarcasm Detection with Humblebrag and Oxymoron via Kuzmin-Logistic Hunter Prey Optimization in Social Web." **(Under Review)**
- Pokhriyal, Himani and Jain, Goonjan. "Unsupervised Sarcasm Detection Using Poisson-KL Divergence Optimization with Zero-Sum Game-Theoretic Pragmatic Modeling in Social Media Discourse." **(Under Review)**

Conference Papers Published

- Pokhriyal, Himani, and Jain, Goonjan. "Sarcasm Detection via sentiment and emotion analysis of News Headlines Using optimized simulation of Weibull entropy distribution." In *2023 Second International Conference on Informatics (ICI)*, pp. 1-6. IEEE, 2023. DOI: 10.1109/ICI60088.2023.10421084

- Pokhriyal, Himani, and Jain, Goonjan. "Implicit and explicit sarcasm detection via target of offensive language and cognition of logistic tsallis entropy distribution in social networks." In *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, vol. 2, pp. 1-6. IEEE, 2024. DOI: 10.1109/IATMSI60426.2024.10503150

Book Chapter Published

- Pokhriyal, Himani, and Jain, Goonjan. "Sarcasm Detection: Acknowledging Misleading Content in Social Media Using Optimised Wilson's Technique and Gumbel Mechanism." In *Harnessing Artificial Emotional Intelligence for Improved Human-Computer Interactions*, pp. 197-221. IGI Global, 2024. DOI: 10.4018/979-8-3693-2794-4.ch012

ACKNOWLEDGMENT

As I present my Ph.D. thesis, I begin by expressing my deepest gratitude to the almighty for the countless blessings and presence in my life, which I experience daily through my family, teachers, and friends.

I would like to extend my sincere thanks to my supervisor, Dr. Goonjan Jain, mam for her patient guidance and unwavering faith in me throughout my doctoral journey. I feel extremely fortunate to have had a guide who genuinely cares about her students and motivates them to achieve their professional goals. Her determination, positive outlook, and confidence have always been a source of inspiration for me.

I also wish to thank the Head of the Department, Prof. Ramesh Srivastava, sir and other faculty members for providing all the necessary facilities and support for conducting my research. Their continuous guidance, kind words of encouragement, and concern have created a conducive working environment that has been invaluable to my research endeavours. Additionally, I am grateful to the department staff for their assistance throughout my Ph.D.

The enduring source of my mental and emotional strength has been my parents and younger sister. Their unconditional love and perpetual support are the primary reasons this thesis has come to fruition. I am also deeply indebted to my friend, Dr. Amarendra Mishra, who has been an integral partner in my research and whose unwavering support motivated me during challenging times.

Completing this work would have been significantly more difficult without the support and friendship of my colleagues, Mr. Kris Gunjan, Ms. Anjali Aggarwal, Dr. Neha Punetha, and Ms. Reetika Singh. I would like to thank each of them individually, not only for their assistance but also for their extraordinary friendship, which made this journey so memorable. Finally, I extend my heartfelt thanks to all those whose names are not mentioned here but have contributed in any way throughout my Ph.D. journey.

(Himani Pokhriyal)

This thesis is dedicated to my parents and sister.

Table of Contents

List of Tables

List of Figures

List of Symbols, Abbreviations, and Nomenclature

NOTATION	DESCRIPTION
X	Total number of sentences $1 \leq i \leq m$
S1 and S2	Set of two sentences taken from the defined dataset
V_{pos}, V_{neg} and V_{comp}	Positive, negative and compound score obtained from VADER sentiment
x	Sum of the valence square of words in VADER sentiment model
S_{pos}, S_{neg} and S_{neu}	Positive, negative and neutral score obtained from SentiWordNet
W	Total number of words in the sentence $1 \leq j \leq k$
$T_{pos}, T_{neg}, T_{pol}$ *and* T_{sub}	Positive, negative, polarity and subjectivity scores obtained from TextBlob model
F_{pos}, F_{neg}, and F_{conf}	Positive, negative and confidence score obtained from FLAIR sentiment model
ϕ_1	Optimal sentiment value in SOBH-REB method
SCM	Sentiment context matrix
ISS_1	Composite individual sentence score in SOBH-REB method
ψ_1	Optimal sarcasm value in SOBH-REB method
τ_1	Sarcasm detection threshold value using Renyi entropy in SOBH-REB method
r	Average of positive and negative senti score $\left(\frac{(S_{pos}+S_{neg})}{2}\right)$ from SentiWordNet
ϕ_2	Optimal sentiment score in SDO-BCRT method
λ_2	Sentiment threshold value using Pearson correlation coefficient in SDO-BCRT method
$\mathrm{S_{pos}}_{\mathrm{mean}}$ and $\mathrm{S_{neg}}_{\mathrm{mean}}$	Mean values of positive and negative sentiment score
k^1_{max}	Highest triggered indicator of sarcasm using YAKE in SDO-BCRT
ψ_2	Optimal sarcasm score in SDO-BCRT method
τ_2	Sarcasm detection threshold value using Renyi entropy in SDO-BCRT method
ϕ_3	Optimal sentiment score in cauchy distribution method

$avg_{T_{pol}}$ and $avg_{T_{sub}}$	Average of polarity and subjectivity score obtained from TextBlob model
λ_3	Sentiment threshold value using polarity and subjectivity score average in cauchy distribution method
k^2_{max}	Highest triggered indicator of sarcasm using RAKE in cauchy distribution method
ψ_3	Optimal sarcasm score in cauchy distribution method
τ_3	Threshold value for sarcasm detection using shannon entropy in cauchy distribution method
$emot_{avg}$	Average emotion score obtained from Tex2emotion
$emot_{embed}$	Distinct emotion scores of five categories, surprise, fear, anger, sadness and happiness
ψ_4	Optimal sarcasm score in LDL-SDTE method
τ_4	Threshold value for sarcasm detection using tsallis entropy in LDL-SDTE method
ψ_5	Optimal sarcasm score in SWD-SEO method
τ_5	Threshold value for sarcasm detection using Shannon entropy in SWD-SEO method
ISS_2	Individual sentence score obtained from TextBlob in SLIOEN
ISS_3	Individual sentence score obtained from SentiWordNet in SLIOEN
k^3_{max}	Triggered sarcasm indicator score using RAKE
ψ_6	Optimal sarcasm score in SLIOEN
τ_6	Sarcasm detection threshold value using shannon entropy in SLIOEN
ϕ_4	Optimal sentiment score in SD-GKOT method
λ_4	Threshold value for sentiment using Kullback Liebler divergence in SD-GKOT method
ISS_4	Individual sentence score in SD-GKOT method
k^4_{max}	Highest key phrase value using RAKE in SD-GKOT method
ψ_7	Optimal sarcasm score in SD-GKOT method
τ_7	Threshold value for sarcasm detection using Tsallis entropy in SD-GKOT method
PQR	Penultimate Quarttete rates $\{\omega_1, \omega_2, \omega_3, \omega_4\}$

ISS_5	Individual sentence score using BOW in UCRT-CGKO method
ISS_6	Individual sentence score using Tf-Idf in UCRT-CGKO method
k_{max}^5	Highest triggered sarcasm indicator score using YAKE in UCRT-CGKO method
A and B	Players chosen in minmax-maximin game in UCRT-CGKO method
ψ_8	Value of game as optimal sarcasm score in UCRT-CGKO method
τ_8	Threshold value for sarcasm detection using Kullback Leibler divergence entropy in UCRT-CGKO method
ISS_7	Individual sentence score using Glove embedding
ISS_8	Individual sentence score using word2vec embedding
ISS_9	Individual sentence score using ElMo embedding
ψ_9	Optimal sarcasm score in SDWT-ELM method
k_{max}^6	Highest triggered sarcasm indicator score using RAKE in SOBH-REB method
α_1	Optimum sarcasm kind value in SOBH-REB method
$Q_{\alpha_1}^1$	Threshold value for types of sarcasm kind 1 in SOBH-REB method
$Q_{\alpha_1}^2$	Threshold value for types of sarcasm kind 2 in SOBH-REB method
$Q_{\alpha_1}^3$	Threshold value for types of sarcasm kind 3 in SOBH-REB method
α_2	Optimum sarcasm kind value in SDO-BCRT method
Q_{α_2}	Threshold value for types of sarcasm using shannon entropy in SDO-BCRT method
α_3	Optimum sarcasm kind value in LDL-SDTE method
Q_{α_3}	Threshold value for types of sarcasm using shannon entropy in LDL-SDTE method
α_4	Optimum sarcasm kind value in SD-GKOT method
Q_{α_4}	Threshold value for types of sarcasm using shannon entropy in SD-GKOT method
α_5	Optimum sarcasm kind value in UCRT-CGKO method
Q_{α_5}	Threshold value for types of sarcasm using shannon entropy in UCRT-CGKO method
k_{max}^7	Highest triggered sarcasm indicator score using YAKE in SDWT-ELM method
α_6	Optimum sarcasm kind value in SDWT-ELM method

Q_{α_6}	Threshold value for types of sarcasm using shannon entropy in SDWT-ELM method
α_7	Optimum sarcasm kind value in SLIOEN method
Q_{α_7}	Threshold value for types of sarcasm using shannon entropy in SLIOEN method
ε_{avg}	Cohen's Kappa score
ϕ_5	Optimal sentiment value in SPOZ-ESD method
λ_5	Threshold value for sentiment using shannon entropy in SPOZ-ESD method
HSWN	HindiSentiWordNet
k_{max}^{8}	Highest triggered sarcasm indicator score using RAKE in SPOZ-ESD method
ψ_{10}	Optimal sarcasm score in SPOZ-ESD method
τ_{10}	Threshold value for sarcasm in SPOZ-ESD method
e^{avg}	Average sentence embedding score in SPOZ-ESD method

Chapter 1

Introduction

Sarcasm is a way of speaking or writing where the intended meaning is different from, or even opposite to the literal words. It is often used to express hidden emotions, criticize someone, or make fun of something in a clever way. On platforms like Twitter, Reddit, or Facebook, people frequently use sarcasm to share thoughts indirectly or humorously. This makes it hard for machines to understand the real meaning of such messages. As a result, sarcasm detection has become an important task in NLP, helping improve systems like sentiment analysis, opinion mining, and online content monitoring.

This chapter gives the background for the work presented in this thesis for sarcasm detection. *Section 1.1* introduces the idea of sarcasm and its role in digital conversations, along with examples from social media. *Section 1.2* explains various types of sarcasm, while *Section 1.3* discusses its importance in areas like marketing, politics, and mental health. *Sections 1.4* to *1.6* review existing detection methods, tools, and ways to measure accuracy. *Section 1.7* outlines current challenges, and *Section 1.8* states the research problem. The chapter ends with the motivations and goals in *Sections 1.9* and *1.10*, followed by an overview of the remaining thesis chapters in *Section 1.11*.

1.1 Background

Sarcasm is a type of figurative language where people say something but mean the opposite [1]. It is often used to make fun of someone or to express criticism in a clever or humorous way. According to the Oxford dictionary, sarcasm is occurred when we use words that say one thing but actually mean another, often to mock or tease someone [2]. People use sarcasm because it adds a unique and powerful touch to how they communicate [3]. One key feature of sarcasm is that the words may sound positive, but the real meaning is negative or the other way around [4]. *Table 1.1* shows real-life examples of sarcastic sentences used in everyday English.

Table 1.1 Examples of sarcastic sentences

S. No.	***Sentences***	***Target***
1.	*"Wow, I like the sauce stains on your shirt."*	*Sarcastic*
2.	*"If that is a human being, then is my kidney a human being too?"*	*Sarcastic*

Fig. 1.1 Illustration of sarcastic sentence in English text

Consider the sarcastic sentence, *"I love working at the office on holidays,"* as shown in *Fig. 1.1.* While the word "*love*" expresses a positive sentiment, the phrase "*working at the office on holidays*" reveals an underlying negative intent. In this sentence, the contrasting word pairs "*love*" and "*office*," as well as "*holidays*" and "*working*" create a contradiction. These opposing associations contribute to the sarcastic tone of the sentence.

In today's digital era, activities such as data collection, reading product reviews, and engaging in social media interactions have become integral to modern life. The rapid growth of platforms like Facebook, Twitter, and Instagram has further amplified the influence of social media. With this surge, the need to identify and understand sarcasm in user-generated content has become increasingly important.

Sarcasm detection is the task of teaching computers to recognize when someone says one thing but actually means something different. These kinds of statements often have a hidden negative tone and are used to mock, tease, or quietly criticize someone or something. The main goal is to help machines understand these subtle expressions in a way that is similar to how humans do. *Figs. 1.2* and *1.3* show real examples of sarcastic comments taken from social media platforms like Twitter and Instagram.

Fig. 1.2 Snapshot of a sarcastic sentence from Twitter

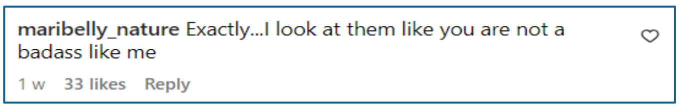


Fig. 1.3 Snapshot of a sarcastic comment from Instagram

Sarcasm detection is important for understanding the real meaning behind a sentence, especially when we rely only on written text. It plays a key role in NLP and sentiment analysis, helping computers understand feelings and opinions more accurately. By spotting the difference between what is said and what is actually meant, sarcasm detection gives useful insights to researchers. It also improves how we study human behaviour and communication in digital spaces. Still, teaching computers to detect sarcasm is not easy. This is because sarcasm often depends on context [5], hidden word clues [6], and tone [7] which are hard to find in text. Also, unlike face-to-face talk, written messages do not include voice or facial expressions, making it harder for machines to fully understand sarcastic remarks.

Traditional sarcasm detection methods [8], [9], [10] often struggle to identify both implicit and explicit sarcastic cues. Even state-of-the-art approaches [11], [12], [13] face limitations, particularly in detecting tonal inflection within sarcastic contexts. Therefore, there is an ongoing need for the application of optimization techniques to improve the performance and reliability of sarcasm detection systems.

Various methods have been proposed for sarcasm detection in English text [14], [15]. However, when it comes to Hindi, significantly fewer resources are available to effectively capture the nuances of this figure of speech [16]. The scarcity of sentiment taggers and annotated datasets [17] contributes to the limited availability of sarcasm detection techniques in Hindi. *Fig. 1.4* and *Fig. 1.5* provide examples of sarcastic expressions in Hindi, extracted from Twitter.

Fig. 1.4 Snapshot of sarcastic Hindi comment

Fig. 1.5 Illustration of a sarcastic Hindi comment

Types of sarcasm refer to the various forms in which sarcastic expressions manifest, each characterized by a distinct tone, purpose, and style. Common types include hyperbole, dropped negation, rhetorical devices, like-prefixed sarcasm, callousness, humour, offensiveness, humblebrag, and oxymoron. Categorizing these types plays a vital role in understanding the speaker's true intent and the subtle impact conveyed through the message. It is especially important in social media analysis, where sarcasm can obscure sentiment and lead to misinterpretation. Without proper classification of sarcastic expressions, the meaning behind user-generated content can often be misunderstood. Therefore, identifying specific types of sarcasm is a crucial step toward building more context-aware and emotionally intelligent language models.

In this thesis, we present a range of methodologies for sarcasm detection in English text using mathematical optimization techniques, addressing the limitations identified in previous studies. Additionally, we propose a novel approach for detecting sarcasm in Hindi text. Furthermore, new methods have been developed for categorizing sarcasm into distinct types in English text. This enabling a more nuanced and structured understanding of sarcastic communication.

1.2 Types of Sarcasm

Sarcasm appears in various forms, each characterized by a unique tone, intention, and linguistic structure. These different types help convey irony, criticism, or mockery in subtle and context-dependent ways. Commonly recognized types of sarcasm include hyperbole, dropped negation, rhetorical devices, like-prefixed sarcasm, callousness, humour-based sarcasm, offensiveness, humblebrag, and oxymoron. Each type reflects a different strategy used by speakers to disguise or contrast the literal and intended meanings.

Identifying and categorizing these types is essential for understanding the true intent behind sarcastic statements. This is particularly important in the context of social media, where sarcasm can distort the perceived sentiment and lead to misinterpretation. Misclassified sarcasm can result in inaccurate sentiment analysis, flawed opinion mining, and misleading insights in user-generated content. Therefore, a well-defined taxonomy of sarcasm types not only deepens linguistic understanding but also enhances the performance of NLP models by enabling more context-aware and emotionally intelligent interpretations.

We present three subcategories under which various types of sarcasm can be grouped. A visual representation of these subcategories is shown in *Fig. 1.6.*

- Meaning Inversion Sarcasm
- Surface-Depth Sarcasm
- Masked Intention Sarcasm

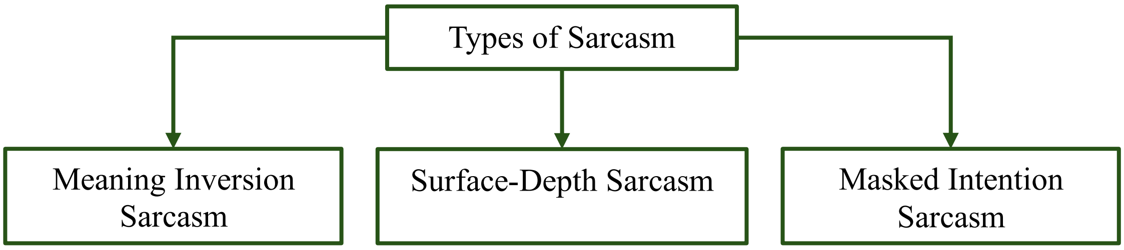


Fig. 1.6 Categorizations of types of sarcasm

1.2.1 Meaning Inversion Sarcasm

Meaning inversion sarcasm involves the use of linguistic manipulation to indirectly express an opposite or inverted meaning. This type of sarcasm is often subtle and relies on the contrast between literal expression and intended sentiment. The category includes hyperbole, like-prefixed sarcasm, rhetorical devices, and dropped negation.

- Hyperbole:

This form of sarcasm uses exaggeration for comedic or satirical effect. Consider an example: *"Oh sure, I just LOVE working 80 hours a week for no recognition at all!"* In this sentence, the speaker exaggerates the phrase "*I just LOVE*," which initially conveys a positive connotation. However, the frustration is emphasized in the phrase "*working 80 hours a week for no recognition,*" leading to a contradiction and revealing the sarcastic intent.

- Like-Prefixed Sarcasm:

This type of sarcasm employs a "like" prefix to introduce a dismissive or ironic tone. For example: *"Like, yeah, because cleaning up everyone else's mess is totally my dream job."* Here, the use of "*Like, yeah*" signals sarcasm and sets a mocking tone, clearly indicating that the speaker does not enjoy the situation.

- Rhetorical Devices:

This form of sarcasm often involves rhetorical questions and elements of irony, wit, or satire. In the example: *"Oh, absolutely! I enjoy sitting in traffic for three hours every day? Really!",* the phrase "*Oh, absolutely!*" appears positive, but the negative intent is expressed through "*sitting in traffic for three hours every day?*" The contrast between the enthusiastic tone and the frustrating situation creates the sarcastic effect.

- Dropped Negation:

In this type, sarcasm is conveyed by omitting a negation word (e.g., "not"), thereby implying the opposite of the literal statement. For example: *"That's just the best movie I've ever seen!"* (said after watching a terrible film). The speaker omits "*not*" to imply dissatisfaction, making the sentence sarcastic through contradiction between literal praise and actual opinion. *Fig. 1.7* provides a diagrammatic representation of meaning inversion, sarcasm, and its subtypes.

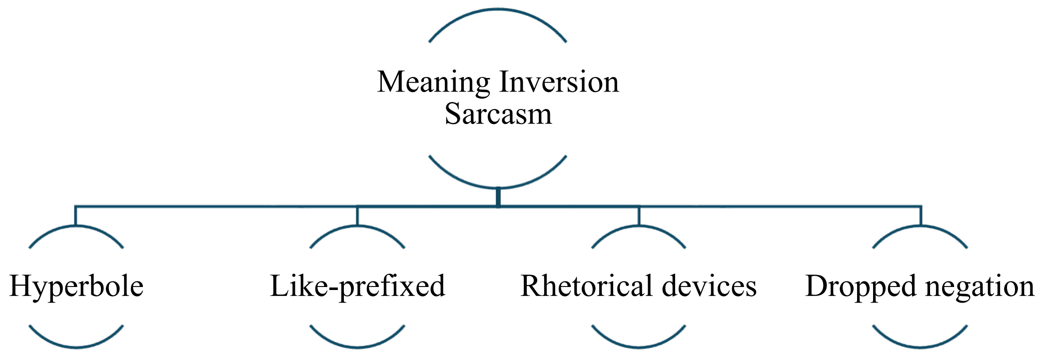


Fig. 1.7 Subtypes of Meaning Inversion Sarcasm

1.2.2 Surface-Depth Sarcasm

This type of sarcasm is based on how clearly the sarcastic meaning is shown in speech or writing. It depends on how easy it is to understand the sarcasm and how much the listener or reader needs to guess the true meaning. Some sarcastic comments are clear and easy to spot, while others need more context or deeper thinking to understand. The sarcasm can be obvious through certain words or tone, or it may be hidden and require background knowledge to catch the joke.

- Implicit Sarcasm:

Implicit sarcasm is not directly stated and depends a lot on the situation and subtle hints. The sarcastic meaning is hidden and must be guessed by the reader or listener using context or past knowledge. For example: "*Wow, you're so thoughtful for eating the last slice of pizza without asking.*" Here, the speaker says "*so thoughtful*" but actually means the opposite. The sarcasm is understood only when we know the situation, making it clear that the speaker is mocking the selfish act.

- Explicit Sarcasm:

Explicit sarcasm clearly communicates the opposite of the intended meaning, often using direct contradiction and a mocking tone. The sarcastic intent is obvious from the wording itself. Let us take an example: *"Oh, you're late? I didn't even notice after waiting for an hour."* Here, the phrase "*I didn't even notice*" is clearly sarcastic, as it contradicts the speaker's frustration from waiting. The sarcasm is directly conveyed without requiring much inference.

1.2.3 Masked Intention Sarcasm

This category of sarcasm involves expressions that aim to either mislead or deliver harsh criticism, often masked under a subtle or ambiguous tone. These types of sarcasm are particularly nuanced, relying on irony or contradiction to conceal negative sentiment beneath seemingly benign language. *Fig. 1.8* illustrates the subtypes of masked intention sarcasm.

- Callousness:

This type of sarcasm reflects indifference toward the feelings or opinions of others. Example: *"You didn't get the promotion? Shocking. I mean, you only worked half as hard as everyone else."* In this sentence, the word "*Shocking*" mocks the person's failure, while the phrase "*you only worked half as hard*" delivers sharp criticism of the individual's effort.

Masked Intention Sarcasm
- Callousness
- Humour
- Offensiveness
- Humblebrag
- Oxymoron

Fig. 1.8 Subtypes of masked intentions sarcasm

- Humour:

Humorous sarcasm is light-hearted and playful. It is typically not meant to offend, but rather to amuse or add a comical twist to a situation. For example: *"Oh great, another Monday. Just what I needed to brighten my life!"* Here, the speaker ironically presents Monday as something positive, using exaggeration to mock the dullness often associated with the start of the week.

- Offensiveness:

This type of sarcasm is used to make fun of, insult, or put someone down. It often has a sharp or harsh tone and can hurt the other person's feelings. For example, the sentence "*Wow, you* *finally showed up only two hours late. Must be your superpower!*" sounds like praise but is actually criticism. The speaker is using exaggerated words to point out that the person was very late, hiding the insult behind a fake compliment.

- Humblebrag:

Humblebragging sarcasm is when someone pretends to complain or act modest but is actually showing off. For example, "*Ugh, it's so annoying when people keep asking for my autograph. Fame is exhausting!*" Here, the speaker seems to be annoyed, but they are really bragging about being famous.

- Oxymoron:

This kind of sarcasm uses two opposite words together to make fun of something or point out how silly it is. For example, "*That's some seriously original plagiarism you've got there!*" The words "*original*" and "*plagiarism*" don't normally go together, and this contrast helps make the sarcastic message stronger and more direct.

In this thesis, we present new methods to automatically sort sarcastic sentences into different types in English. This helps build a more organized and detailed way to understand sarcasm in computer systems.

1.3 Need for Sarcasm Detection

Sarcasm detection is very important in today's digital world, where most communication happens online. When sarcasm is misunderstood, it can cause confusion, arguments, or spread false information. Detecting sarcasm help improve tools like sentiment analysis and chatbots, making online communication more accurate and meaningful. It also plays a key role in understanding what users really mean and helps prevent cyberbullying or wrong interpretations. Therefore, sarcasm detection is an essential part of keeping social media and online networks safe and clear.

- **Understanding Tone and Avoiding Misinformation**: Sarcasm detection helps identify the true tone of a message. It shows whether a comment is misleading or meant as a joke. This reduces confusion and helps avoid conflicts, especially in customer service.

- **Better Sentiment Analysis**: Sarcasm detection improves how we study emotions in text. It helps in tasks like topic detection, question-answering, and opinion analysis. This leads to a better understanding of what people truly feel on social media.
- **Smarter Social Media Analysis**: It helps track and understand online conversations more clearly. Sarcasm detection reveals the tone and intent behind messages, making it easier to follow discussions and online trends.
- **Improving Services**: It is useful in analysing product reviews and customer feedback. By spotting sarcasm, companies can better understand what customers mean and improve their services or products.

In conclusion, sarcasm detection is an important tool for clear communication in online communities. It helps reduce misunderstandings and supports respectful and open digital interactions.

1.4 Techniques for Sarcasm Detection

This section explains the main techniques used in today's systems for detecting sarcasm. These techniques include machine learning, deep learning, rule-based systems, and transformer-based models. They help analyse the meaning, tone, and context of text to find sarcastic expressions.

- **Machine Learning Methods:** Machine learning models use both supervised and unsupervised approaches to study text. In supervised learning, the system is trained with labelled data, where each piece of text is already tagged with its correct category. The model learns from this and then predicts sarcasm in new texts. Unsupervised learning, on the other hand, works with unlabelled data and finds patterns using clustering techniques. However, these models often require a lot of tuning and can be expensive to run due to high computational needs.
- **Rule-based Methods:** These techniques depend on predefined rules or language patterns to identify sarcasm. The rules might be based on grammar, sentence structure, or specific keywords. Although rule-based systems can work well for certain topics or languages, they need a lot of manual work to build and update. Also, they struggle to detect sarcasm that doesn't match the set rules.
- **Deep Learning Methods:** Deep learning has brought major improvements to sarcasm detection. These models use multiple layers of neural networks to understand complex meanings and relationships in text. They are good at capturing deep patterns in

language. However, they often do not explain how different types of sarcasm are learned or identified by the model.

- **Transformer-based Methods:** These models, such as BERT, look at the meaning of a sentence from both directions to understand it better. They often use supervised learning to classify sarcasm. While these models are powerful, they sometimes fail to deal with the hidden or conflicting meanings that sarcasm often contains.

Choosing the right technique for sarcasm detection depends on several factors, such as the size of the dataset, the complexity of the language, the domain of use, and the resources available. In this research, we propose mathematical optimization techniques to detect sarcasm in both English and Hindi texts, along with identifying their specific types.

1.5 Lexicon Datasets and Python Libraries for Sarcasm Detection

Lexicon datasets and Python libraries play a key role in sentiment analysis tasks in NLP. Lexicon resources like SentiWordNet (SWN) and HindiSentiWordNet (HSWN) store words with assigned emotional values. They help in identifying the tone of text as positive, negative, or neutral.

- **SentiWordNet:** SWN is a lexical resource for sarcasm detection that extends WordNet, a collection of English words organized into synonym sets (synset). It assigns sentiment scores to words in WordNet based on their positive, negative, and neutral polarity. *Fig. 1.9* shows an excerpt from SentiWordNet.
- **HindiSentiWordNet:** HSWN is a special version of SentiWordNet created for the Hindi language. It is used to find the emotional tone of words written in Hindi. The model works by using WordNet synsets and gives each word a score that shows whether it is positive, negative, or neutral in meaning. This helps in understanding the true feeling or mood behind Hindi text. *Fig. 1.10* shows a sample taken from the HindiSentiWordNet model.

	sentences	sentences_without_stopwords	After_lemmatization	pos_tags	senti_score	positive_sentiscore	negative_sentiscore
0	A perfect example of why Christian fundamental...	perfect example christian fundamentalismm evol...	perfect example christian fundamentalismm evol...	[(perfect, JJ), (example, NN), (christian, JJ)...	-0.50	1.500	2.000
1	This decision was not solely based on self, bu...	decision solely based self thenreally dongivei...	decision solely based self thenreally dongivei...	[(decision, NN), (solely, RB), (based, VBN), (...	0.25	0.500	0.250
2	I love finding out who your true friends are. ...	love finding true friends itseriously best	love finding true friend itseriously best	[(love, VB), (finding, VBG), (true, JJ), (frie...	1.00	1.375	0.375
3	USER :/ bye	user bye	user bye	[(user, NN), (bye, NN)]	0.00	0.000	0.000
4	USER I absolutely love Kanye's voice USER	userabsolutely love kanyevoice user	userabsolutely love kanyevoice user	[(userabsolutely, RB), (love, VB), (kanyevoice...	0.00	0.000	0.000

Fig. 1.9 Snapshot of SentiWordNet with polarity on dataset

sentences	senti_score	positive_sentiscore	negative_sentiscore
एक "कर की विरासत और उधार लेने की विरासत दो साल पहले विरासत में मिली अर्थव्यवस्था को अपंग कर दिया।"	-0.500	0.250	0.75
एक "हमारे अधिकांश छात्र" अंडरपरफॉर्मिंग स्कूलों में फंस गए हैं।	-0.500	0.250	0.75
विस्कॉन्सिन के लिए $ 300 मिलियन के प्रोत्साहन आवंटन ने "इस राज्य के हर कोने में हजारों नौकरियों का निर्माण किया है।"	0.125	0.125	0.00
2005 के एक गे प्राइड वोट "भेदभाव से कोई लेना -देना नहीं था।"	0.875	1.125	0.25
पेपर बैग पर 5-प्रतिशत का शुल्क एक कर है।	-0.750	0.000	0.75

Fig. 1.10 Snapshot of HindiSentiWordNet with polarity on dataset

Python libraries like TextBlob and VADER (Valence Aware Dictionary and Sentiment Reasoner) use word-based scores to understand the sentiment in a sentence. TextBlob is good for general language tasks, while VADER is better at handling casual and informal text, such as content from social media. When used together, these tools help computers better understand emotions in human language. This section explains the lexical datasets and Python libraries that are useful for detecting sarcasm in text.

- **TextBlob:** TextBlob is a Python library used for working with text in NLP tasks. It is built on top of the NLTK library and makes it easier to handle text data. TextBlob is commonly used in applications like text classification, content analysis, and information retrieval. *Fig. 1.11* shows a sample output from the TextBlob library.
- **VADER sentiment:** VADER Sentiment is a rule-based tool for sentiment analysis. It is specially designed to understand emotions in text, even when sarcasm is involved. VADER is part of the Python NLTK library and is freely available for analysing the emotional context of text. *Fig. 1.12* displays an example output from the VADER sentiment tool.

	sentences	lemmatized	polarity	subjectivity	sentiment	negative	neutral	positive
0	It is forstudy im doing at school what are the...	[forstudy, im, school, social, economic, benef...	0.116667	0.133333	positive	0	0	1
1	Trust the Podcast	[trust, podcast]	0.000000	0.000000	neutral	0	1	0
2	Those ears...	[ears...]	0.000000	0.000000	neutral	0	1	0
3	I started doubting Sunni Islam fromacceptedwas...	[started, doubting, sunni, islam, fromaccepted...	0.100000	0.200000	positive	0	0	1
4	Everytimehear or see the word tripe Rocket Pow...	[everytimehear, see, word, tripe, rocket, powe...	0.000000	0.666667	neutral	0	1	0

Fig. 1.11 Snapshot of TextBlob NLP library with polarity on dataset

	label	neg	neu	pos	compound	sentences
0	0	0.297	0.598	0.105	-0.6921	A perfect example of why Christian fundamental...
1	1	0.145	0.804	0.051	-0.5719	This decision was not solely based on self, bu...
2	2	0.062	0.294	0.643	0.9501	I love finding out who your true friends are. ...
3	3	0.545	0.455	0.000	-0.3400	USER :/ bye
4	4	0.000	0.527	0.473	0.6697	USER I absolutely love Kanye's voice USER

Fig. 1.12 Snapshot of the VADER sentiment NLP library with polarity on the dataset

1.6 Challenges in the Field of Sarcasm Detection

Sarcasm detection presents inherent challenges due to the nuance nature of sarcastic expressions, which often rely on contextual, tonal, and pragmatic cues. The absence of linguistic markers, coupled with limited annotated resources and user-specific variability, further complicates automated detection. The existing state-of-the-art methods reveal that there are several shortcomings in the present works of sarcasm detection, summarized as follows.

- **Contextual awareness:** Detecting sarcasm is difficult without understanding the full context behind a comment. Most models [1], [18] miss out on user intent or background references that carry sarcastic cues. This lack of context often leads to wrong or shallow interpretation of sarcastic text [18].
- **Learning exaggerated and subtle markers:** Sarcasm often hides behind exaggerated words or very subtle hints in tone or style. Identifying subtle markers is challenging, especially when they are culturally nuanced. Machines [19], [11], [10] struggle to catch what humans naturally sense in dramatic or dry remarks.
- **Recognition types of sarcasm:** Sarcasm appears in many forms like irony, mockery, or rhetorical questions. Recognizing which type is present in a sentence is a major challenge for systems. This makes it hard to build models [20], [21], [22] that fully understand varied sarcastic expressions.
- **Semantic and tonal cues:** Understanding sarcasm is not just about words it is also how they are said or meant. Capturing emotional tone or hidden meaning through text alone is quite difficult. Models [22], [17], [23] often miss the mix of tone and intent that defines sarcastic language.
- **Low resource language limitations:** Detecting sarcasm in low-resource languages like Hindi faces serious constraints [24]. There are fewer labelled examples and limited NLP

tools to support deep analysis [25], [26]. This makes it tough to train systems that can handle sarcasm across languages.

To address these challenges, we propose novel sarcasm detection methods leveraging optimization techniques with enhanced accuracy and adaptability. Addressing the specific issue, we aim to provide effective and reliable methods for detecting sarcasm in English and Hindi text.

1.7 Evaluation Metrics

We have evaluated the proposed methods performance with different evaluation metrics. These metrics help in providing the effectiveness of the proposed methods. The evaluation metrics are listed below in *Eqns. (1.1) to (1.16)*.

- Accuracy: Accuracy measures the presence of correct predictions. Here, we compute the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) ratings.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{1.1}$$

- Precision: Precision measures the correctness of actual and predicted classifications, having TP and FP ratings.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{1.2}$$

- Recall: Recall measures the completeness of any classification or information retrieval of a system. It accounts for TP and FN for the classification of the final score.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{1.3}$$

- F1 score: The harmonic mean of precision and recall is the F1 score, which is high when both precision and recall values are high.

$$\text{F1 score} = \frac{2*\text{Precison}*\text{Recall}}{\text{Precision+Recall}} \tag{1.4}$$

- Matthew’s correlation coefficient: The Matthew’s correlation coefficient (MCC) is used to measure how well a binary classification model performs. It looks at all parts of the confusion matrix TP, TN, FP, and FN even when the number of samples in each class is different. A higher MCC value shows better prediction accuracy.

$$\text{MCC} = \frac{(\text{TP}\times\text{TN})-(\text{FP}\times\text{FN})}{\sqrt{(\text{TP+FP})\times(\text{TP+FN})\times(\text{TN+FP})\times(\text{TN+FN})}} \tag{1.5}$$

- Misclassification Rate: The misclassification rate (MR), tells us how often the model makes wrong predictions. It counts the number of incorrect results (FP and FN) and compares them to the total number of predictions made.

$$\mathrm{MR} = \frac{\mathrm{FP+FN}}{\mathrm{TP+FP+TN+FN}} \tag{1.6}$$

- False positive rate: The false positive rate (FPR) shows how often the model wrongly predicts something as positive when it is actually negative. A lower FPR means the model is making fewer such errors.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} \tag{1.7}$$

- False negative rate: The false negative rate (FNR) tells us how often the model misses real positive cases. In other words, it shows the number of times something that should have been detected is not identified by the model.

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP+FN}} \tag{1.8}$$

- False discovery rate: The false discovery rate (FDR) measures how many of the predicted positive results are actually wrong. It helps us understand how often the model is "fooled" into making false positive predictions.

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{TP+FP}} \tag{1.9}$$

- Macro-averaged precision: Macro-averaged precision is used in multi-class classification problems. It calculates precision separately for each class and then finds the average. All classes are treated equally, no matter their size.

$$\text{Macro-averaged precision} = \frac{\sum_{i=1}^{n} \mathrm{Precision}_i}{n} \tag{1.10}$$

- Macro-averaged recall: Macro-averaged recall works the same way as macro precision, but it focuses on recall. It looks at how well each class is recognized and then takes the average without giving weight to any specific class.

$$\text{Macro-averaged recall} = \frac{\sum_{i=1}^{n} \mathrm{Recall}_i}{n} \tag{1.11}$$

- Macro F1 score: Macro F1 Score combines macro precision and macro recall into one single value. It gives a balanced view of how well the model performs across all classes equally.

$$\text{Macro F1 score} = 2 \times \frac{\mathrm{Macro{-}Precision} \times \mathrm{Macro{-}Recall}}{\mathrm{Macro{-}Precision} + \mathrm{Macro{-}Recall}} \tag{1.12}$$

- Micro-averaged precision: Micro-averaged precision calculates precision across all classes by adding up the total TP and FP first, and then applying the precision formula. It gives more weight to larger classes.

$$\text{Micro-averaged precision} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i} \tag{1.13}$$

- Micro-averaged recall: Micro-averaged recall works in a similar way. It adds up all true positives and false negatives across every class before calculating recall. This method also gives more importance to bigger classes.

$$\text{Micro-averaged recall} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \tag{1.14}$$

- Micro-averaged F1-score: Micro-averaged F1 Score brings together micro precision and micro recall to give a single score. It shows the overall balance of the model's performance across all classes, considering both correct and missed predictions.

$$\text{Micro-averaged F1 score} = 2 \times \frac{Micro-Precision \times Micro-Recall}{Micro-Precision + Micro-Recall} \tag{1.15}$$

- Cohen Kappa metric: It is a statistical measure denoted as ε used to evaluate the level of agreement between two annotators when they classify the same set of items. It is here used in annotation tasks such as identifying sarcasm types or discourse labels.

$$\varepsilon = \frac{h_0 - h_e}{1 - h_e} \tag{1.16}$$

Evaluation metrics are very important for checking how well a model works. They give clear numbers that help us measure how accurate, efficient, and reliable the proposed method is. These metrics help us understand whether the model makes good or poor predictions. By using them, we can test how strong the method is in different situations. This also helps in spotting what the model does well, where it may be weak, and how it can be improved further.

1.8 Problem Statement

In this thesis, we aim to leverage optimization techniques for developing unsupervised methods to analyse sarcasm in written texts.

1.9 Motivation

Sarcasm detection helps understand people's true opinions, feelings, and attitudes about products, events, services, or organizations. It is an important part of NLP because it lowers the chances of misunderstanding what someone really means. Finding sarcastic remarks allows for clearer and more accurate communication. It also helps detect when someone may be trying to mislead or hide their true intent. This is especially useful in understanding how people express themselves in online conversations.

1.10 Organization of the Thesis

This thesis presents a structured exploration of sarcasm detection using advanced techniques. Each chapter builds upon the previous to explain methods, models, and evaluations clearly.

Chapter 1 introduces sarcasm detection, its applications, and the motivation behind this thesis. It outlines key challenges, the problem statement, objectives, and evaluation metrics used to compare proposed methods with existing approaches.

Chapter 2 reviews state-of-the-art methods for sarcasm detection, sentiment analysis, and optimization techniques to identify their strengths, limitations, and research gaps. It also discusses the datasets used for experimental validation of the proposed methods.

Chapter 3 presents sarcasm detection methods using statistical distributions and transformation optimization techniques on English datasets. The effectiveness is validated through experiments, with results compared and numerically interpreted in a dedicated section.

Chapter 4 presents sarcasm detection methods using unsupervised hybrid optimization techniques in English. The effectiveness is validated through experiments, with results compared and numerically interpreted in a dedicated section.

Chapter 5 focuses on learning the inherent intonation analysis of sarcasm types using non-linear optimization techniques.

Chapter 6 presents a novel sarcasm detection model using unsupervised optimization techniques for Hindi. Its effectiveness is evaluated through various parameters, with comparative analysis and numerical interpretations provided.

Finally, in **Chapter 7**, we summarize the conclusions from the proposed research work and highlight the potential future work in this area.

Chapter 2

Literature Review

Sarcasm is a subtle and often witty way of expressing meaning by saying the opposite of what one truly intends. While humans can usually pick up on sarcastic tones through the voice of the speaker, recognizing sarcasm in text without any vocal cues remains a difficult task, even for machines. This challenge has sparked growing interest in the field of sarcasm detection, where the goal is to teach computers how to identify sarcastic remarks in text data. This is especially important for applications like sentiment analysis, customer feedback interpretation, and human-computer interaction, where understanding true intent matters.

In recent years, range of techniques have emerged to tackle this task from rule-based systems and machine learning models to more sophisticated deep learning and transformer methods. These approaches attempt to capture not just the words themselves, but also the underlying context, emotional tone, and contradictions that often signal sarcasm.

This chapter presents a comprehensive literature review on sarcasm detection, based on traditional machine learning, deep learning methods, and various optimization techniques.

In this chapter, we have organized the literature review into the following subsections. *Section 2.1* presents the related work of sarcasm detection and its working using various techniques. In *Section 2.2*, we conduct a new study of sarcasm affective indicators alongside their types in *Section 2.3*. *Section 2.4* focuses on the related work concerning various mathematical optimization techniques and distributions. *Sections 2.5* and *2.6* laid down the research gaps from the past studies and outlined the research objectives of the thesis. *Section 2.7* covers the literature that explores the datasets used in the proposed methodology. Finally, *Section 2.8* concludes the key findings of this chapter.

2.1 Sarcasm Detection Techniques

Sarcasm detection has recently acquired a tangible foundation of research in multiple areas, like supervised learning, unsupervised learning, deep learning, and transformer-based techniques. Sarcasm detection research began in 2010 when researchers created a corpus by extracting texts from several social websites. Scientific organizations and businesses are becoming more and more interested in the knowledge of sarcasm detection. The foundation of sarcasm detection is detailed in the following categories.

2.1.1 Machine Learning-Based Methods of Sarcasm Detection

Machine learning (ML) methods shift the paradigm from programming computers to enabling them to learn directly from data. Employing these algorithms that discern inherent patterns, make predictions, classifications, or decisions without rigid, pre-defined rules. This learning process manifests in two primary forms: supervised and unsupervised learning. These machines learning based methods are divided into two categories, elaborated as follows.

Supervised learning-based methods: Supervised learning requires a large corpus of labelled training data and the feature engineering process. The researchers have introduced some algorithms for sarcasm detection based on supervised learning.

Xiong et al. [27] adopted lexical sequence features with semantic imbalance to detect sarcasm. Liu et al. [28] depicted the incongruity of sarcasm in the form of implicit and explicit features. Rajadesingan et al. [29] argued that contextual trait has a crucial role in detecting sarcasm. Rahma et al. [20] displayed a review paper for sarcasm detection using supervised techniques in the Arabic language. Băroiu and Matu et al. [30] demonstrated another review of automatic sarcasm detection using predominant machine learning and transformer techniques.

Wen et al. [31] introduced a long short-term memory (LSTM) technique with sememe knowledge for sarcasm detection. Muaad et al. [32] depicted an artificial intelligence (AI)-driven model of supervised technique with a misogyny form of sarcasm detection. Dave et al. [33] demonstrated product reviews of using semantic knowledge of text for sarcasm detection. Chatterjee et al. [34] had given the bidirectional (BiLSTM) auto-encoder method for sarcasm detection. Vinoth and Prabhavathy [35] conveyed intelligent machine learning-based semantic learning of sarcasm detection using a support vector machine (SVM).

Pan et al. [18] displayed a semi-supervised technique of sarcasm detection methodology using SVM. Poria et al. [1] displayed a convolutional neural network (CNN) and LSTM method in recognizing sarcasm. Huang and Galinsky [36] utilized a combination of LSTM, CNN, and BiLSTM-based techniques for sarcasm detection. Lemmens et al. [24] had provided a sarcasm detection model consisting of an ensemble embedding approach, with CNN-LSTM and LSTM.

Das et al. [19] introduced a method for sarcasm detection using hybrid four-parallel LSTM network known as FLSTM. Li et al. [37] provided an enhanced commonsense knowledge of sarcasm using COMET model, which gave the knowledge of the text integration mechanism. Riloff et al. [38] proposed a bootstrapping method to identify sarcastic tweets by recognizing positive and negative situation phrases. Ptáček and his team [39] compared LSTM and SVM classifiers using different sets of features across both Czech and English datasets. Araque [40] worked on classifying sentences based on sentiment targets using Bi-LSTM and CNN models. Devlin [41] and Ren [42] applied a combination of CNN and LSTM to detect sarcasm in text. Krishna [43] used ensemble classifiers along with LSTM for sentiment analysis of tweets on Twitter. Hiai and Shimada [44] introduced a method that combined interface LSTM with RNN-based knowledge for sarcasm detection.

Mehndiratta and Soni [45] developed a neural language model that used Word2Vec, GloVe, and FastText, along with Bi-LSTM and CNN, to capture sarcasm. Tan and his team [46] proposed two methods implicit and explicit inconsistency analysis to identify sarcastic content using AutoML, DeepConcat, and BiLSTM. Plepi and Flek [47] were the first to use a graph attention network (GAT) to detect sarcasm on social media, by analyzing user conversations and past tweets.

Mou [48] introduced a model called rumor2vec and used a graph convolutional network to detect sarcastic rumors. Li [49] suggested an affection-enhanced graph attention network (ARGAT), which combines emotional and dependency features to better understand sarcasm. Lu [50] created a network that joins fact-sentiment conflicts with LSTM and SVM models. Another approach by Li [51] focused on using attention-based contextual knowledge for sarcasm detection, while Minaee [6] designed a hybrid system using CNN, LSTM, and deep neural networks.

Unsupervised learning-based methods: No predefined labelled datasets are available for these techniques. Clustering is a widely used unsupervised classification technique. In the clustering algorithm, we divide a set of objects into clusters, grouping similar objects together and dissimilar ones into different clusters. Several techniques are given by the researcher for sarcasm detection based on unsupervised techniques.

Lee and his team [52] used a clustering-based topic modeling method to identify sarcasm in text. Kumar and Sarin [53] worked with unsupervised learning and developed a fuzzy logic model that used word embedding techniques like GloVe and Word2Vec, along with a BiLSTM model, to detect sarcasm. Liang and his group [54] explored sarcasm detection using multimodal data by creating in-modal and cross-modal graphs, called InCrossMGs. Some review papers [55], [56], [57] have also studied sarcasm detection using fuzzy logic, clustering analysis, and pattern recognition with ensemble models.

In addition, several earlier studies have focused on multimodal sarcasm detection using well-known datasets such as MUStARD and SemEval from the years 2015, 2016, and 2017 [58], [59]. Mittal et al. [60] introduced a type-2 fuzzy method with a clustering model for sarcasm detection. The multiscale visual kernel used to depict hate speech comments from sarcasm on social networks, produced by Chhabra and Vishwakarma [61].

Meanwhile, these traditional ML models rely on hand-crafted features [46], [47], [48] and shallow linguistic representations [55], [56], [57]. However, they often fall short of capturing the complex, context-dependent nature of sarcasm cues. There are need for more robust feature engineering and adaptive learning techniques that can dynamically interpret subtle sentiment shifts and cultural nuances which help in sarcasm detection.

2.1.2 Deep Learning and Transformer-based Methods for Sarcasm Detection

Deep learning based methods enable classification methods to automatically learn complex and abstract data features and adaptability across diverse applications. Kalchbrenner et al. [62] and Shen et al. [63] introduced a convolutional architecture embedded with dynamic neural network (DCNN) with max-pooling and a self-attention model. Yang et al. [64] combined CNN designs for the task of sarcasm detection. Ghosh and Veale [5] provided a DNN with an LSTM design for sarcasm detection and sentiment analysis.

Minaee et al. [6] introduced sarcasm using CNN and LSTM networks in collaboration with DNN. Sarcasm detection used graph neural networks (GNN) and graph-structured data provided by Wu et al. [7]. Mohan et al. [65] identified sarcastic tone in texts by integrating GCN and bidirectional encoder representations from transformers (BERT) as a BERT-GCN model. Yang et al. [66] provided XLNet with a BERT mechanism for detecting aggressive language patterns with sarcasm. Cai et al. [67] depicted the correlation between humour and sarcastic markers with subjective knowledge.

He and his team [68] developed a sarcasm detection system using the BERT model, which helps understand the context of language more accurately. Tiwari et al. [69] proposed a technique that combines quantum fuzzy neural networks with BERT to detect both sentiment and sarcasm in social media posts. Helal and his group [70] used a RoBERTa-based model, which focuses on deeper context, to improve sarcasm detection. Palivela [71] introduced two important features: one looks at the structure of text in a hierarchical way, and the other focuses on specific phrases that help to indicate sarcasm. Ilievski and his team [72] worked on identifying sarcastic meanings within different conversational settings.

Ren et al. [73] used a CNN with BERT to get better outcomes for sarcasm detection. Tay et al. [74] utilized a word embedding attention-based neural model with semantic information. Akula and Garibay [75] suggested a model consisting of a multi-head attention mechanism to identify crucial cue terms from gated recurrent network. Kolchinski and Potts [76] utilized social network analysis using sentiment features and neural models. Zhang et al. [77] introduced a quantum interactive neural network with sentiments in conversational contexts. Vitman et al. [4] presented a model using a sentiment emotion-based BERT-CNN structure. Kumar et al. [78] depicted word-emoji-based techniques utilizing hybrid networks to detect sarcasm. Keivanlou-Shahrestanaki et al. [79] introduced the multi-head self-attention architecture framework of sarcasm detection. Du et al. [80] utilized a CNN BiLSTM-based framework to detect sarcasm.

Despite the success of these transformer models in NLP tasks, their performance in sarcasm detection is impeded by challenges such as limited domain-specific annotated datasets [76], [77], [4] over-reliance on surface textual patterns [70], [71], and difficulties in capturing long-range contextual dependencies [72], [73]. More focused pre-training and fine-tuning strategies specifically for sarcastic expressions are necessary in these proposed models.

2.1.3 Multi-Modal based Sarcasm Detection

Multimodal classification of sarcasm detection integrates multiple data types (e.g., text, images, audio, video) to improve sarcasm detection by leveraging complementary information. Multimodal sarcasm detection with sentiment has been given by Li et al. [81]. Schifanella et al. [82] developed written and visual traits for recognizing sarcasm and applied them to a multipurpose task learning. Xu et al. [83] developed a linked approach of multimodal odd words using connection networks and disintegration.

In addition, Pan et al. [84] introduced a model built on a BERT that used the significance of cross-modality and in-modality characteristics. Kolhatkar et al. [85] has given various in-modal and cross-modal representations based on BERT. They identified phrases of disparity and incompatible effects across and within texts. Yue et al. [2] suggested a knowledge fusion-based transformer technique, i.e., the KnowleNet method for sarcasm detection. Qiao et al. [86] introduced a multi-enhanced incongruity method for sarcasm detection.

Pandey and Singh [3] conveyed transformer-based fusion of BERT and LSTM in code-mixed language based sarcasm analysation. Zhang et al. [87] showed BERT-based stance-level sarcasm detection using a gated attention transformer. Savini and Caragea [88] introduced transformer learning with the BERT technique in the sentences. Meng et al. [11] used BERT and CNN methods for the sarcasm detection task.

Galal and his team [12] worked on sarcasm detection in Arabic by improving machine learning techniques to increase accuracy. Hassan et al. [13] explored sarcasm in Urdu by using a hybrid model that combines transformer-based methods for better understanding of the text. Alayba and his group [89] designed a model that merges LSTM and CNN to identify sarcasm in Arabic social media posts. Similar research has also been done in other languages like Indonesian [90] and Spanish [91], where supervised learning and deep learning models have been applied to recognize sarcasm in various online texts.

Sarcasm in real-world interactions is frequently conveyed through a mix of modalities, including text, intonation, and pragmatic cues. However, effective fusion of these modalities remains a challenge, particularly when aligning disparate data types and accounting for modality-specific ambiguities [11], [12], [13]. Developing unified, multi-modal models and curated datasets is a promising yet underexplored research direction in sarcasm detection.

2.1.4 Rule and Pattern-Based Methods for Sarcasm Detection

In this class of sarcasm detection, the authors identify a collection of patterns. A rule-based method, primarily focused on the hyperbolic and syntactic features within the content for sarcasm detection. Gong et al. [8] found indicative words and characteristic language using rule based pattern which helped in the task of sarcasm detection. Tsur et al. [9] and Bharti et al. [10] introduced the incompatible emotional states inside statements as a key factor for sarcasm detection. Maynard and Greenwood [92] have used the emotional knowledge contained in labels for sarcasm detection.

Chauhan et al. [93] initiated an emoji-aware multitask learning-based method for sarcasm detection. Hiremath and Patil [94] provided a negation-aware task with a language model for sarcasm detection. Ilić et al. [95] used an ensemble language model, i.e., ELMo, to identify sarcasm in text. Bharti et al. [96] specifically used the value of pattern-based characteristics for sarcastic statements recognition. Chen et al. [97] Introduced a Hadoop-based framework that collected real-time tweets to detect tonal and gestural clues in sarcastic statements.

Nevertheless, these methods offer interpretability and domain expertise, but they often lack flexibility [10] in the case for sarcasm detection. Static rules struggle to keep pace with the evolving nature of language [94] and diverse cultural contexts [92] that influence sarcastic expression. There is a gap in designing dynamic rule systems that can flexibly adapt to new linguistic trends and contextual variations [95], [96] in detecting sarcasm.

2.1.5 Methods for Sarcasm Detection in Hindi Text

Recent studies have started exploring code-mixed content on social media, especially in low-resource languages like Hindi. However, there is still very little research focused entirely on texts written in pure Hindi. One of the earliest works in this area was by Desai and Dave [25], who developed a dataset of sarcastic Hindi sentences. They trained an SVM-based classifier using features like punctuation and polarity. Bharti and his team [26] later used a contextual part-of-speech tagging approach with a hidden markov model to decide if a tweet was sarcastic. They counted positive and negative keywords using a fixed list of Hindi words marked by sentiment values.

Another model by Bharti et al. [98] used patterns in time-related information and inconsistencies in statements to predict sarcasm in Hindi tweets. Katyayan and colleagues [99] analyzed 1,000 Hindi posts from platforms like Facebook, Instagram, and Twitter. They used POS tagging and bag-of-words (BOW) methods to extract features, then applied machine learning models like naive bayes, SVM, and decision trees. However, their results lacked strong interpretation and reliability.

In another work, Bharti's team [10] proposed a parsing-based context technique (PBCT) to count positive and negative words in tweets, using a predefined Hindi lexicon. Later, the TANA model [100] was introduced using LSTM-SVM architecture for sarcasm detection, and Kumar and his team [78] applied a CNN-LSTM system for the same task. Bharti's recent work [101] also introduced a context-based pattern (CBPSD) that used Hindi news articles as contextual references with a lexicon tool. In related multilingual studies, Bousmaha and colleagues [102] worked on sarcasm detection in Algerian using deep learning with the EasyOCR tool. While Marreddy et al. [103] developed a BERT-based approach for sarcasm analysis in language. A Ahmed et al. [104] used a revised lexicon model with embeddings to learn NLP patterns of satire contexts in Hindi text.

Meanwhile, these methods include a scarcity of annotated corpora specially in Hindi. Additionally, linguistic complexities such as dialectal variations and extensive code-switching with English text, make the capturing of culturally nuanced expressions difficult for Hindi sarcastic text. Focusing efforts on dataset creation and model adaptation for sarcasm detection in Hindi are essential measures need to solved.

Following the discussion of the established techniques applied in sarcasm detection, the subsequent section explores approaches that are specifically tailored to different categories of sarcasm.

2.2 Methods Based on Meaning Inversion Sarcasm

This type of sarcasm includes the leaning of those sarcastic utterances, which include the semantic contrast between what is said and what is meant. This type of sarcasm categorization includes four subtypes, such as hyperbole, rhetorical devices, like-prefixed and dropped negation. Over the past decade, sarcasm detection has gained attention in the field of NLP and sentiment analysis. Researchers have explored linguistic cues such as hyperbole, dropped negation, and rhetorical questions are essential to identify sarcastic intent [93].

Expressions starting with "like" have also been studied, as they often reflect subtle irony or mock praise in online conversations. These models typically rely on surface-level features and sentiment shifts to flag sarcastic content [94]. However, many fall short when sarcasm is deeply embedded in cultural or conversational context. They often overlook speaker intention, emotional tone, or the contrast between literal and intended meaning [95]. Such limitations affect performance, especially in detecting nuanced, multi-layered sarcasm across diverse platforms. Improving contextual understanding remains a key challenge for advancing sarcasm detection systems.

2.3 Methods Based on Surface-Depth Sarcasm

This type of sarcasm considered implicit and explicit kinds of sarcasm. This type of sarcasm usually help to get the inherent contextual features of the sarcastic text. Most of the studies have been conducted to detect sarcasm, but these studies still failed to give a correct classification at a better computational cost for implicit and explicit sarcasm.

Explicit sarcasm [105] is shown through sentiment words of opposing polarities. For example, in the sentence "*I love being scolded all the time,*" in this sentence "love" is positive, and "scolded" is negative which give contradictions. Implicit sarcasm [106] is subtly conveyed through phrases with implied sentiment rather than opposing polar words. For example, in "*I love waiting in line for hours, it's the highlight of my day*", the only polar word is *"love*," but the phrase *"waiting in line* for hours" implies drift negative sentiment that contrasts with "*love*".

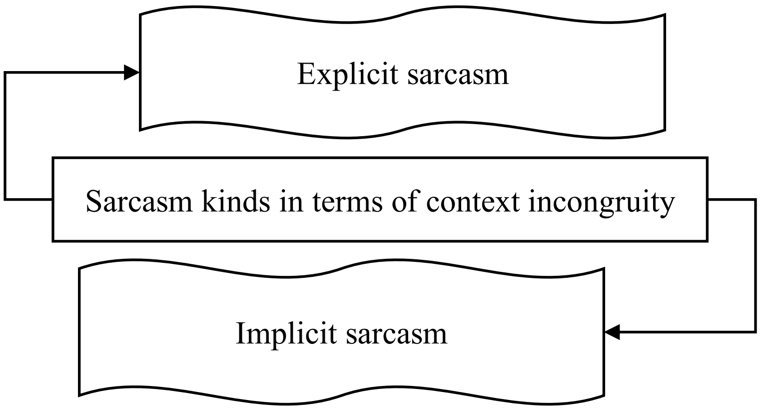


Fig. 2.1 Classification of sarcasm kinds

Fig. 2.1 shows the architectural framework for kinds of sarcasm in a conceptual way. From past studies, Barbieri et al. [107] highlighted the presence of sarcasm in Twitter comments with figurative language, with sentiment analysis. However, its drawback is that it relies heavily on rule-based features, limiting scalability and not distinguishing sarcasm from inherent intents. Ghosh et al. [108] introduced sarcasm magnets as a lexical resource to detect sarcasm using a neural network model with psychological context but lack to learn inherent intent of contexts.

Bedi et al. [109] introduced sarcasm depiction differing from the humour context model MSH-COMICS using attention neural utterance classification. Somewhat, it lacked generalizability beyond small experimental datasets. Majumder et al. [15] provided attention to a neural tensor network-based model for sentiment and sarcasm detection in sentences. But they neglect the presence of affective intonations in a sarcastic context. Meanwhile, high computational costs struggled with balancing modality contributions restricted to only tweets.

Bouazizi et al. [96] introduced a pattern-based approach on Twitter that is sensitive to noisy or irrelevant input features. However, they neglect types, kinds, and indicators of sarcasm. Potamias et al. [110] showcased a hierarchical neural network RCNN-RoBERTa framework to handle sarcasm context. Moreover, noise in external commonsense data and challenges in cross-modal alignment remain unresolved in this method.

Li et al. [51] emphasized the role of inconsistency and developed a sentiment-based approach to sarcasm detection in multimodal attention settings. Nevertheless, the issue of overfitting on small datasets due to the lack of generalized pre-trained models and the overlooked kinds of sarcasm with inherent intonations are some of its challenges.

Rajadesingan et al. [29] introduced a multi-behavioral approach for sarcasm detection across social media like Twitter platforms, utilizing rule-based and pattern-based features. Meanwhile, it struggled with imbalanced datasets and the inability to fully capture sarcasm's contextual dependencies. Gupta et al. [111] investigated how emotion context factor-based sarcasm detection using ANN. This method needs improvement in learning the implicit and explicit intent of expressions. Zhang & Liu [112] introduced a hybrid CNN model using quantum probability distribution with fuzzy logic membership to identify explicit sarcasm in tweets. Despite that, it lacked implicit tonal traits in sarcastic sentences and neglected sarcasm-triggered indicators.

Liu et al. [113] used cross-modal transformers for implicit sarcasm detection in multi-turn dialogues on Reddit. Regardless, this method required extensive computational resources are suffered from overfitting on small datasets, and avoid giving sarcasm nuances that posed a challenge. Chen et al. [114] combined BERT with a sarcasm-aware attention mechanism for better detection of both explicit and implicit sarcasm in social media data. Nonetheless, the existing method is unable to classify types of sarcasm with their indicators. Liang et al. [54] introduced a method for depicting implicit and explicit kinds of sarcasm nature with fewer intonation features. However, it neglected the various features of sarcasm with the presence of implicit intonations.

2.4 Methods Based on Masked Intention Sarcasm

In recent years, scholars have begun to investigate deeper emotional and stylistic elements in sarcasm, including offensiveness and callousness. Offensive sarcasm, often overlapping with toxic or abusive speech, has been studied in the context of online aggression and hate speech classification [115]. Such forms of sarcasm usually mask hostility behind humour or exaggeration, making detection difficult without cultural or pragmatic context.

Callous sarcasm, marked by emotional detachment or insensitivity, which is less overt but can be damaging, especially when it targets vulnerable groups [116]. Another related phenomenon is the humblebrag, where individuals mix modesty with self-praise, often in a sarcastic tone. Though subtle, humblebragging carries ironic undertones and presents a challenge for systems relying only on surface-level sentiment features.

Oxymorons contradictory terms used for stylistic or ironic effect are also common in sarcastic and humorous expressions. While these linguistic devices add depth to language, they often confuse models not trained on semantic paradoxes. Humour, closely tied to sarcasm, has been studied using computational models that focus on incongruity, punchline structure, and context-based embeddings. However, distinguishing light-hearted humour from sarcastic or biting commentary remains a major limitation in current scenarios. These challenges suggest that sarcasm detection requires more than just lexical or syntactic features; and it demands deeper understanding of tone, intent, and social nuance.

2.5 Mathematical Optimization Techniques used for Sarcasm Detection

Mathematical optimization refers to the process of finding the best solution (maximum or minimum) to a problem within a defined set of constraints. The goal is to determine the optimal value for decision variables that satisfy constraints while optimizing a given objective function. In this section, we present the use of mathematical optimization techniques to detect sarcasm and its inherent intonation features using different kinds of optimization techniques. Based on evaluating the optimal value of the optimization, the task of sarcasm detection is performed.

Certainly, we dealt with linear optimization and non-linear optimization techniques. Linear optimization (also known as constrained optimization) involves maximizing an objective function while considering a set of constraints subject to them. In this optimization, the objective function is either minimized or maximized. As opposed to moderate constraints, it includes certain factors that are penalized by the target function. The constraints are applied to specific decision variables. Whereas non-linear optimization is a method of optimizing (maximizing or minimizing) a non-linear objective function. The objective function is accompanied by constraints where either the objective function, the constraints, or both are nonlinear [117].

The following are the optimization techniques and distributions used in the proposed methods formulation for sarcasm detection.

The first one is the Chinese remainder theorem (CRT), which describes the criteria that must be met for several equations to generate a contemporaneous integer solution [118]. CRT has been used in various studies, including Li et al. [163], who's provided the method for CRT using two integer residue sets where the remainders have some errors. Lu et al. [120] suggested the use of CRT with fog computing-enhanced internet of things (IOT) applications, giving a lightweight privacy-preserving data aggregation (LPDA) model. Xiao et al. [121] depicted the collaborative presence of CRT in the geometry property of linear space, giving pseudometrics as a uniform framework. Li et al. [122] provided the vector-based robust framework of CRT (Vec-CRT) with maximum likelihood estimation as MLE-CRT for measuring the distance of radio interferometric.

In mathematical optimization, the Kuhn-Munkres algorithm, commonly referred to as the Hungarian method, is employed to deal with allocation issues [123]. It utilized the best allotment from a cost matrix while maximizing the overall profit of a given objective function. The given problem involved finding the optimal assignment of agents with tasks while minimizing the total cost or maximizing the total profit. Fei et al. [124] provided the aspect-based sentiment triplet extraction framework underlying the Hungarian problem with a non-auto-regressive decoding algorithm.

Liu et al. [125] introduced the berth allocation and quay crane assignment problem (QCAP) with minimizing the negative deviation objective function. It provides the behavior perception-based disruption (BAP) analysis. Zeng et al. [126] introduced the EmotionCues, a visual analytics system to analyse classroom videos from an emotion perspective with an assignment problem mechanism. Another one is the Box-Cox transformation [171], which is a collection of statistical alterations used to reduce variability and bring a dataset closer to a typical distribution. When tackling irregularities and non-constant variance in predictive modelling, it is helpful.

Silveira et al. [127] introduced the topic modelling using an autoencoder with Gumbel SoftMax and logistic normal distributions. Huijben et al. [128] explained a review of Gumble max and discrete stochasticity distributions in patterns of contexts. Lin et al. [129] detailed topic modelling evaluation using the NLP model and Gumbel distribution for short text. Zheng et al. [130] highlighted the hierarchical Gumbel attention network model's working using an optimization heuristic model in a text-based system.

Mardešić et al. [131] introduced the phenomenon of zeta functions of orbit-based parabolic distribution utilizing a linear optimization technique. Shah et al. [132] suggested the comparative analysis of logistic regression with random forest and KNN model distribution for text classification. Xu et al. [133] introduced a BERT-based NLP model for modelling a basic warranty data study with logistic distribution. Hidayat et al. [134] provided insights into the sentiment analysis of Twitter using doc2vec, logistic distribution, and an SVM model.

Akgül et al. [135] detailed the distribution of Weibull for modelling wind speed data. Abdullah et al. [136] focused on the new version of the weighted Weibull distribution in the modeling of COVID-19 textual data. Rajpathak et al. [137] presented the data ontology-driven text mining reliability model, sentiment evaluation. Uglanov et al. [138] explained the framework for the identification of lifecycle issues with reliability analysis. Cao et al. [139] examined the augmented Lagrange point approach with NLP problems on the graphics unit.

The Zipf-Mandelbrot distribution technique has been used in various existing state-of-the-art methods, which are subsequently elaborated as follows. Linders et al. [140] explored the Zipf's distribution phenomenon in spoken dialog linguistic units and parametric effects. Sridevi et al. [141] described a robust modification of Zipf's law for text analysis using a linear regression technique. Linders et al. [142] investigated the human sentiment evaluation phenomenon using Zipf's mechanism. Oliveira et al. [143] analyzed Zipf's law using graph theory operations with a non-linear optimization model.

On the other hand, the Gauss-Kuzmin distribution [144] a discrete probability distribution used to characterize the likelihood of the resulting number in data. The Gauss-Kuzmin distribution technique has been implemented by Siddiqui et al. [145] who demonstrated the detection and classification of lung cancer using an improved novel deep belief network. Kuzmin et al. [146] discussed the low spatial coherence evaluation using the NLP model.

2.6 Research Gaps

In this section, based on the above existing state-of-the-art methods, we laid out the final drawbacks and challenges posed by the mentioned methodologies.

- Traditional ML methods [1], [61] heavily rely on hand-crafted features that often fail to capture sarcasm's complex context and inherent linguistic nuances.
- Although transformer-based approaches [66], [67] are effective in many NLP tasks, but they tend to overlook latent intonation signals and focus on superficial textual patterns.
- Meanwhile, multimodal [86], [3] approaches struggled with fusing heterogeneous signals (text, intent, gestural cues) and rarely integrate mathematical optimization models to fine-tune and align these modalities effectively.

- Typically, rule and pattern-based methods are too rigid and unable to adapt dynamically to the evolving nature [96] and the inherent features of sarcastic expressions [95], [97].
- Whilst methods in the case of sarcasm detection in Hindi encounter unique challenges due to limited annotated datasets [147], dialect variations [148], code-switching [100], and insufficient modelling of sarcasm's intrinsic cues [78].
- Additionally, most of the mathematical optimization methods are underexplored within NLP frameworks. Employing these methods could help in capturing intention signals and the inherent features of sarcasm more effectively.

2.7 Research Objectives

After understanding the given drawbacks and challenges, we formulated certain objectives for sarcasm detection, which are to be tackled by this thesis.

- To detect sarcasm types in terms of polarity, as implicit and explicit sarcasm.
- To propose an optimization-based method for the sarcastic classification of English text with the presence of hyperboles and other figurative features.
- To detect the presence of types of sarcasm in English text and differentiate between them using optimization methods.
- To detect sarcasm using an optimization-based method in Hindi text.

2.8 Datasets

Datasets play a crucial role in sarcasm detection, as they provide the foundation for validating machine learning models to understand and recognize sarcasm across various contexts. *Fig. 2.1* shows the basic datasets used for sarcasm detection by past studies. The textual datasets are categorized in this thesis as resource-abundant and resource-constrained, respectively.

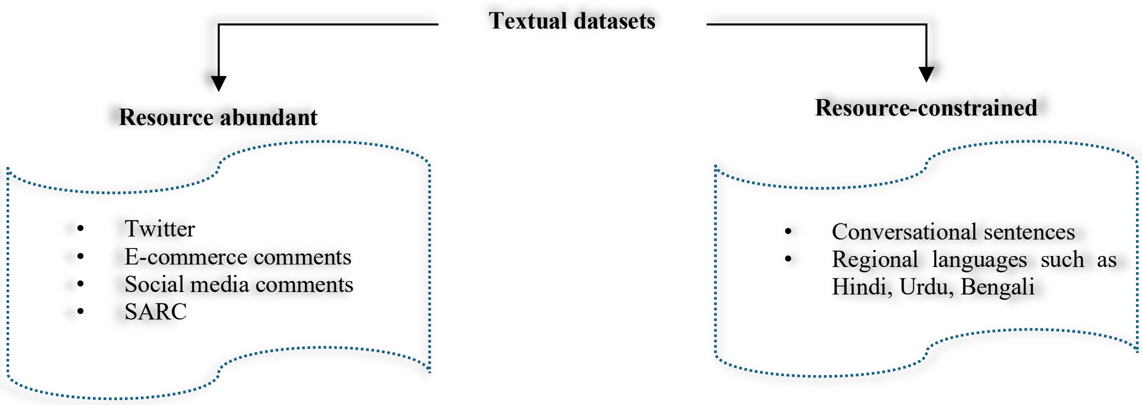


Fig. 2.2 Varied datasets used for sarcasm detection in text

In this thesis, we have undertaken certain benchmark datasets to implement the proposed methodologies. *Table 2.1* shows the statistics of defined datasets used for sarcasm detection by the proposed methods.

Table 2.1 Statistical variations of datasets

S. No.	Datasets	Renamed as	Total number of sentences
1.	*Self-Annotated Reddit Corpus (SARC) [149]*	*Dataset-1*	*12,257*
2.	*Reddit comments [150]*	*Dataset-2*	*7,895*
3.	*Semeval 2018 task [151]*	*Dataset-3*	*11,025*
4.	*News Headlines [152]*	*Dataset-4*	*12,506*

Several benchmark datasets have been developed to facilitate sarcasm detection research. These datasets provide labelled examples of sarcastic and non-sarcastic statements, helping to evaluate machine learning models. Below is a detailed elaboration of the key datasets used for sarcasm detection tasks:

- **Dataset-1** [149]: The SARC is one of the most widely used datasets for sarcasm detection. It consisted of Reddit comments where users have explicitly labelled their comments as sarcastic or non-sarcastic. This dataset is particularly valuable as it captures naturally occurring sarcasm in real-world conversations. It is a large-scale dataset containing millions of comments with conversational context and metadata such as subreddit categories, timestamps, and user interactions. However, one of the challenges is its reliance on explicit user annotations, which may not always be accurate. Some comments labelled as sarcastic might be ambiguous or require external knowledge to interpret correctly.
- **Dataset-2** [150]: The Reddit comments dataset is another widely used resource extracted from various subreddits. Unlike SARC, this dataset may include sarcasm labels obtained through different annotation techniques, such as crowdsourcing or automated heuristics. It provided a diverse set of sarcastic and non-sarcastic comments useful for supervised and unsupervised sarcasm detection. Since, it contained thread-based conversations, it offered a rich context for sarcasm interpretation.
- **Dataset-3** [151]**:** The SemEval-2018 task 3 dataset is designed for sarcasm detection in English tweets. This dataset is part of a shared task where researchers worked on distinguishing sarcastic and non-sarcastic statements. It has been divided into two subtasks: binary classification (sarcasm vs. non-sarcasm) and detecting different types of sarcasm,

such as verbal and situational sarcasm. The dataset consisted of manually labelled tweets, ensuring high-quality annotations. It also captured diverse expressions of sarcasm found on social media. However, the informal nature of tweets and the frequent use of hashtags, abbreviations, and slang make sarcasm detection in this dataset particularly difficult.

- **Dataset-4** [152]**:** The News headlines dataset is composed of sarcastic and non-sarcastic news headlines collected from the Orabley dataset. The dataset is useful for studying sarcasm in news media, where sarcasm is often subtle and relies on exaggeration. It provided a straightforward method to sarcasm detection by analysing textual cues in headlines.

While some datasets focused on social media platforms (e.g., Twitter, Reddit), others analyze sarcasm in news articles and longer texts. The inclusion of contextual information, annotation methods, and linguistic diversity is a critical factor in selecting the appropriate dataset for specific sarcasm detection tasks.

2.9 Summary

This chapter has provided a comprehensive literature review on sarcasm detection and optimization techniques, covering various methods used in previous studies. We explored traditional machine learning methods, deep learning techniques, transformer-based architectures, and rule-based methods for sarcasm detection. Each technique has demonstrated effectiveness in different contexts, as machine learning models excel in feature-based classification, deep learning methods leverage contextual learning, transformers capture intricate patterns, and rule-based systems offer interpretability. However, each method has limitations, such as feature dependency, data-intensive requirements, computational complexity, and rigid rule sets in defined approaches. The advantages and shortcomings of these techniques have helped shape the contributions and identifying gaps for improvement.

In addition to sarcasm detection methods, this chapter also delved into optimization techniques used to enhance model performance. Additionally, we determined different types of sarcasm, including verbal, situational, and textual intonation signals in sarcasm detection. Analysing past research and existing techniques, in this chapter, we establish a strong foundation for the research objectives of this thesis. The next chapter will focus on the proposed methodology, detailing the method adopted in this research to enhance sarcasm detection performance while overcoming the limitations observed in existing works.

Chapter 3

Statistical Distributions and Analytical Transformation-Based Methods for Sarcasm Detection in English Text

Building upon the findings discussed in *Chapter 2*, which reviewed a wide range of existing models for sarcasm detection, this chapter introduces novel methodologies using optimization techniques grounded in statistical theory. While past studies has made significant progress in detecting sarcasm using machine learning and deep learning models, but many of them struggle with interpretability, computational efficiency, and performance across varied datasets and languages. To address these challenges, we propose the use of mathematical optimizations combined with advanced probability distributions as a new direction in sarcasm detection.

This chapter presents four new models that incorporate specific statistical distributions and transformations namely, the Bessel function, Box-Cox transformation, Cauchy distribution, and a Logistic-Weibull distribution as objective functions in optimization frameworks. Each model is designed to extract and optimize features such as sentiment shifts, contextual patterns, and syntactic cues that are associated with sarcasm in textual data. These methods not only enhance accuracy but also provide mathematical justification for the detection process. In addition, statistical validation and comparative analysis are conducted using standard benchmark datasets.

The sections in this chapter are structured as follows. *Section 3.1* provides a brief introduction to sarcasm detection and the role of optimization techniques in enhancing its accuracy and efficiency. *Sections 3.2* to *3.5* describe each of the four proposed methodologies. *Section 3.2* discusses the use of the Bessel function for modelling sentiment based sarcasm shifts. *Section 3.3* focuses on the Box-Cox transformation to normalize and optimize skewed text feature distributions for sarcasm detection. *Section 3.4* explores the application of the Cauchy distribution in capturing extreme sentiment fluctuations, which are key indicators of sarcasm depiction. *Section 3.5* integrates the Logistic and Weibull distributions that accounts for both growth and decay patterns in textual emotion analysis of sarcastic cues. *Section 3.6* evaluates the performance of each proposed methods using various datasets, presenting accuracy, precision, recall, F1-score, and ROC-AUC values. It also includes a comparative study and statistical tests such as z-test, Kolmogorov Smirnov, Mann Whitney U test and Kruskal-Wallis H test to validate model performance. *Section 3.7* highlights the practical challenges and limitations of the proposed models, including computational time complexity, scalability across datasets, and adaptability to code-mixed or low-resource languages. *Section 3.8* concludes the chapter with a summary of the key contributions and outlines potential directions for future research, including the integration of these optimization-based models with neural architectures and sarcasm analysis.

3.1 Introduction

The previous chapter reviewed various techniques developed for sarcasm detection, including machine learning, deep learning, and transformer-based models. While these approaches have shown good results, many still struggle with subtle expressions and contextual ambiguity. The existing methods [153], [2] have not shown the signals of sarcasm using mathematical optimization in their methodologies. To address these challenges, we employ mathematical optimization techniques in this chapter as novel methods for sarcasm detection in English text.

Formulating sarcasm detection as mathematical optimization technique enables optimal calculation, leading to the robustness of the proposed methods. Optimization techniques handle large datasets and adapt to different languages or contexts without creating problems. Additionally, using different distribution functions in optimization help capture various aspects of sarcasm with effective indicators (discussed in *Chapter 5*). Using maximized or minimized distributions as an objective function offers unique insights into the problem for sarcasm detection. Integrating optimization techniques help refine feasible scores with optimal solutions, leading to more accurate sarcasm detection. In this chapter, we have discussed five approaches for sarcasm detection in English text using various optimization techniques.

- **Proposed Method 1:** Sarcasm detection using optimised balanced Hungarian technique based on Rényi entropy Bessel mechanism (SOBH-REB), discussed in *section 3.2*.
- **Proposed Method 2:** Sarcasm detection using unconstrained optimisation problem based on Box-Cox Rényi entropy transformation (SDO-BCRT), discussed in *section 3.3*.
- **Proposed Method 3:** Sarcasm detection using semantical Cauchy distribution Shannon optimization technique (SD-SCOT), discussed in *section 3.4*.

- **Proposed Methods 4 and 5:** Two combined models of sarcasm detection using Logistic and Weibull distribution-based optimization techniques. These methodologies are discussed in *section 3.5*.

The proposed methods discussed above are followed by defined steps for sarcasm detection. It is categorized into two distinct phases as follows.

3.1.1 Fundamental steps

The proposed methods use sentiment detection as the foundation for sarcasm detection. To detect, if a sentence is sarcastic or not, we analyse the sentiment behind the sentence as well as its literal interpretation. The conflict help us to detect the sarcasm in the text. The fundamental step of sentiment evaluation is given in the subsequent *subsection 3.1.1.1*.

3.1.1.1 Sentiment evaluation

In the initial step, we calculate the optimal sentiment value using certain NLP methods. We use TextBlob, SentiWordNet, Flair sentiment, and a valence-aware dictionary for sentiment reasoning (VADER) are used to evaluate the optimal sentiment score. These NLP lexicon methods determine the sentiment of context using a given pre-trained process. The detailed *algorithms 3.1, 3.2, 3.3*, and *3.4* are demonstrated as follows.

ALGORITHM 3.1: Calculate text polarity using VADER Sentiment
Input: Input sentence X, VADER sentiment analyzer lexicon *Output: positive (V_{pos}), negative (V_{neg}), and compound (V_{comp}) scores* 1. *Import → VADER sentiment analyzer* 2. *Initialize analyser = SentimentIntensityAnalyzer() and* $V_{pos} = V_{neg} = 0$ 3. *Get sentiment scores: score = analyser.polarity_scores (X)* 4. *If score* ≥ 0.05: *"The text is Positive"* $V_{pos} = score$ 5. *If score* ≤ -0.05: *"The text is Negative"* $V_{neg} = score$ 6. *Calculate_frame_of_reference_score:* $V_{comp} = \frac{x}{\sqrt{x^2+\propto}}$ *where* x *is the sum of the valence square of words and* $\propto$ *is a normalization constant 0.66* 7. *End*

ALGORITHM 3.2: Calculate text polarity using SentiWordNet
Input: input sentence X, SentiWordNet lexicon *Output: positive (S_{pos}) and negative (S_{neg}) senti scores* 1. *Initialize* $S_{pos} = S_{neg} = 0$ 2. *Let* $W = \{ w_1, w_2, w_3, \dots, w_h \}$ *where* w_i *represents the* i^{th} $1 \leq i \leq k$ *word in the sentence input.* 3. *If* $w_i \in SentiWordNet$ *then* $\frac{a\ positive\ sentiment\ value\ of\ w_i}{k} = S_{pos}$ *"positive sentiment."* *and* $\frac{negative\ sentiment\ value\ of\ w_i}{k} = S_{neg}$ *"negative sentiment."* *Where k is the total number of words in sentence X* 4. *End*

ALGORITHM 3.3: Calculate text polarity using TextBlob
Input: input sentence X, TextBlob lexicon *Output: positive (T_{pos}), negative (T_{neg}), polarity (T_{pol}), and subjectivity (T_{sub}) scores* 1. *Import → TextBlob* 2. *Compute sentiment polarity: sentiment = TextBlob (X).sentiment.polarity* *Compute sentiment subjectivity: sentiment = TextBlob (X).sentiment. subjectivity* 3. *sentiment* $\in [-1,1] \rightarrow T_{pol}$ *and sentiment* $\in [-1,1] \rightarrow T_{sub}$ 4. *Initialize scores:* $T_{pos} = 0$ *and* $T_{neg} = 0$ 5. *Assign scores based on polarity:* 6. *If sentiment* < 0: *Return "Negative"* $\rightarrow T_{neg}$ *Else "Positive"* $\rightarrow T_{pos}$ 7. *End*

ALGORITHM 3.4: Calculate text polarity using the Flair sentiment model
Input: *input sentence X, Flair sentiment lexicon model* **Output:** *positive (F_{pos}), negative (F_{neg}) sentiment scores* 1. *Initialize flair NLP sentiment model: "Flair/english-sentiment."* 2. *Function get_sentiment_scores (X)* 3. *Initialize scores:* $F_{pos} = 0$ *and* $F_{neg} = 0$ 4. *Extract sentiment label and confidence score:* *If sentiment_result.label = "positive," assign a confidence score to* F_{pos} *If sentiment_result.label = "negative," assign a confidence score to* F_{eng} 5. *Sentiment scores:* $\{F_{pos}, F_{neg}\}$ 6. *End*

3.1.1.2 Key phrase extraction technique

In the second step, we focus on finding the most significant signs of sarcasm through keyword extraction methods. The first method used is yet another keyword extractor (YAKE), which works without training data. It looks at features such as how often a word appears, its position in the text, and how it occurs with other words. Each keyword is then given a score, where lower values point to higher importance, and the results are listed as keyword score pairs.

The second method is rapid automatic keyword extraction (RAKE), a rule-based approach. It breaks the text into possible keywords by removing stop words and punctuation, then scores them using frequency and co-occurrence patterns. In this case, higher scores highlight more important keywords, which are also presented as keyword score pairs.

In the following section, we provide details of the proposed methods for sarcasm detection using different optimization techniques.

3.2 Proposed Method 1: Sarcasm Detection based on Bessel Function

In this section, we discuss the proposed (SOBH-REB) method for sarcasm detection using an optimization technique containing the Bessel function.

The process of SOBH-REB is structured into three core steps, each elaborated in detail

below.

Step 1: Sentiment Evaluation and Sentence Score Calculation

For the initial step, we utilize TextBlob and SentiWordNet (SWN) to compute the polarity score (T_{pol}), subjectivity score (T_{sub}), and sentiment score (S_{pos}) using *algorithms 3.3* and *3.2*. Then, the optimal sentiment value (ϕ_1) is calculated based on the average sum of T_{pol}, T_{sub}, and S_{pos} shown in *Eqn. (3.1)*. The ϕ_1 is used as the optimal value for determining the overall sentiment.

$$\phi_1 = \frac{T_{pos}+T_{neg}+S_{pos}}{3} \tag{3.1}$$

Step 2: SCM Matrix Generation

In the second step, we generate a third-order square matrix, i.e., sentiment context matrix (SCM) from T_{pol}, T_{sub}, and S_{pos}. We employ these scores in the formation of SCM. The SCM matrix is formed by adding each term with its index term, as shown in *Eqn. (3.2)*. *Fig. 3.1* shows the overall framework of SOBH-REB for sarcasm detection.

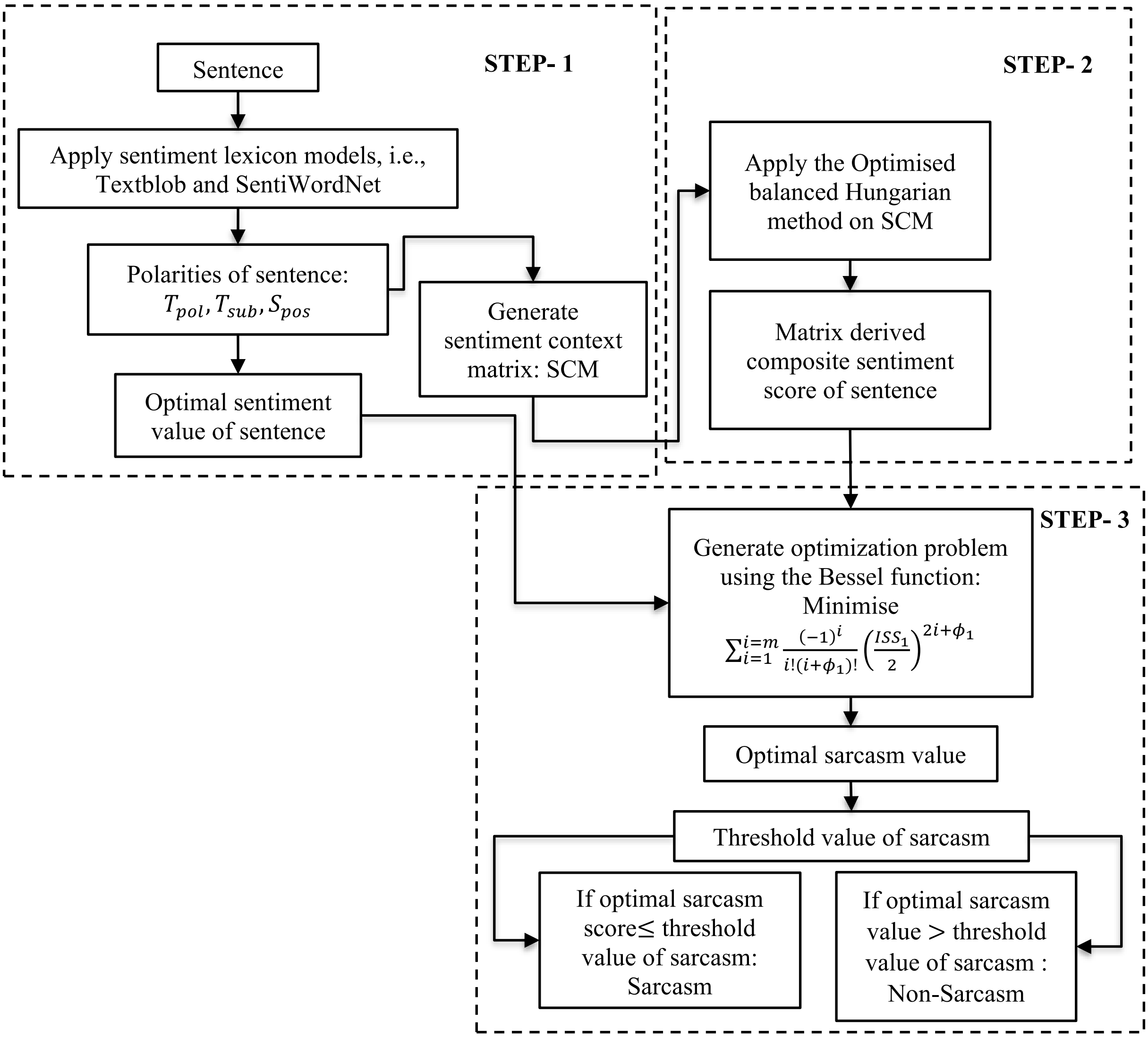


Fig. 3.1 Flowchart of the proposed method SOBH-REB for sarcasm detection

$$\text{SCM} = \begin{bmatrix} \downarrow\rightarrow & T_{pos} & T_{neg} & S_{pos} \\ T_{pos} & T_{pos}+T_{pos} & T_{pos}+T_{neg} & T_{pos}+S_{pos} \\ T_{neg} & T_{neg}+T_{pos} & T_{neg}+T_{neg} & T_{neg}+S_{pos} \\ S_{pos} & S_{pos}+T_{pos} & S_{pos}+T_{neg} & S_{pos}+S_{pos} \end{bmatrix} \tag{3.2}$$

The SCM matrix has been generated using the balanced Hungarian method, resulting in a composite individual sentence score (ISS_1). The procedure is shown in *Algorithm 3.5.*

ALGORITHM 3.5: Calculate CSS using the Balanced Hungarian Method
Input: SCM of a sentence, $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_h}\}$ *total number of sentences in the dataset.* *Output: composite individual sentence score (*$\boldsymbol{ISS_1}$*)* *1. For each sentence,* $\boldsymbol{X}$ *performs steps from steps 1 to 4.* *2. Perform linear sum assignment using the balanced Hungarian method:* *3. Cost matrix [row_indices, col_indices] = linear_sum_assignment(sentiment_cost_matrix)* *4. Calculate the composite sentiment score:* $\boldsymbol{ISS_1}$ *= Cost matrix.sum()* *5. End*

Step 3: Sarcasm detection

In the third step, we detect the presence of sarcasm by formulating a combinatorial non-linear optimization problem with the Bessel function as a minimized objective function. The objective function is subject to ISS_1 (derived in Step 2) and ϕ_1 (derived in Step 1) of the sentence. The main task of the given optimisation problem is to calculate the optimal sarcasm value of the minimised objective function. The Bessel function chosen as an objective function helps to learn the inhibiting traits of a sentence. It makes the algorithm computationally efficient. The defined optimisation problem is formulated in *Eqn. (3.3)*.

$$\left.\begin{array}{c} \psi_1 = Minimise \sum_{i=1}^{i=m} \frac{(-1)^i}{i!(i+\phi_1)!} \left(\frac{ISS_1}{2}\right)^{2i+\phi_1} \\ \text{Subject to}, ISS_1 \geq 0 \; and \; \phi_1 \geq 0 \end{array}\right\} \tag{3.3}$$

The optimal value obtained from *Eqn. (3.3)* is the optimal sarcasm value (ψ_1). The value of ψ_1 is then compared with the defined sarcasm detection threshold value (τ_1). The value of τ_1 is calculated as 2 using Renyi entropy. The Renyi entropy is an essential hyperparameter utilized to calculate the τ_1, as shown in *Eqn. (3.4)*.

$$\tau_1 = \frac{1}{1-t}\log\left(\sum_i \left(P(\psi_{1_i})\right)^t\right) \tag{3.4}$$

Here, $t = 0.44$, $P(\psi_{1_i})$ taken as the probability of α taking on the value ψ_{1_i}, whereas $1 \leq i \leq m$ is the total number of sentences in the dataset. The calculated value of τ_1 set as 2 is shown in *Table 3.1*.

Table 3.1 Computation of threshold value for sarcasm (τ_1)

Parameter	***Values***
Threshold value for sarcasm detection (τ_1)	*(Dataset-1 [149]):* $\tau_1 = \frac{1}{1-0.44} log(17.633) = 2.224 \approx 2$ *(Dataset-2 [150]):* $\tau_1 = \frac{1}{1-0.44} log(9.328) = 1.784 \approx 2$ *(Dataset-3 [151]):* $\tau_1 = \frac{1}{1-0.44} log(13.664) = 2.027 \approx 2$

In comparing the value of ψ_1 and τ_1, we decided whether the sentence is sarcastic or not. If ψ_1 is less than or equal to τ_1, the sentence is tagged as "*sarcastic*"; or else, it is tagged as "*non-sarcastic*." *Algorithm 3.6* defines the procedure of sarcasm detection using the proposed (SOBH-REB) model.

ALGORITHM 3.6: Sarcasm Detection using Proposed Method-SOBH-REB
Input: $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_h}\}$ *total number of sentences in the dataset,* $\boldsymbol{ISS_1}$ *and* $\boldsymbol{\phi_1}$ *Output: optimal sarcasm value (*$\boldsymbol{\psi_1}$*)* 1. *For each sentence,* $\boldsymbol{X}$ *performs steps from steps 1 to 5.* 2. *Generate the optimisation problem using Bessel function:* $\boldsymbol{\psi_1}$ = *Minimise* $\sum_{\boldsymbol{i=1}}^{\boldsymbol{i=m}} \frac{(-\boldsymbol{1})^{\boldsymbol{i}}}{\boldsymbol{i}!(\boldsymbol{i}+\boldsymbol{\phi_1})!}\left(\frac{\boldsymbol{ISS_1}}{\boldsymbol{2}}\right)^{\boldsymbol{2i+\phi_1}}$ *and Subject to,* $\boldsymbol{ISS_1} \geq \boldsymbol{0}$ *and* $\boldsymbol{\phi_1} \geq \boldsymbol{0}$ 3. *Print the output → "optimal sarcasm value (*$\boldsymbol{\psi_1}$*)"* 4. *Based on a comparison between* $\boldsymbol{\psi_1}$ *and* $\boldsymbol{\tau_1}$*:* 5. *If* $\boldsymbol{\psi_1} \leq \boldsymbol{\tau_1}$ *→ "sarcastic"* *Else "non-sarcastic"* 6. *End*

3.2.1 Numerical Illustration of the Proposed Method: SOBH-REB

Let us consider the following raw sentences from our datasets to demonstrate the implementation of the proposed model.

S1: "I mean, if you guys still don't understand what you did to him, then I don't think you'll ever get it."   ***S2: "I'm glad this pretentious piece of shitt didn't do as planned by the Dodge stratus Big Shots... It's gonna help movie makers who aren't in the very restrained "movie business" of Québec…"***   ***S3: "When you see a relative who spotted you with your gf the other day. Praise by my parents DUDE."***

In step 1, of SOBH-REB, we get the ϕ_1, In step 2, we generate the SCM matrix to get individual sentence scores using *Algorithms 3.2*, *3.3*, and *3.5*, respectively. *Tables 3.2, 3.3*, and *3.4* show the computation of ϕ_1, SCM, and ISS_1. In step 3, we perform sarcasm detection using *Algorithm 3.6*. *Table 3.5* shows the computation strategy for detecting sarcasm using the proposed (SOBH-REB) method.

Table 3.2 Numeric values of optimal sentiment score (ϕ_1)

Sentences	T_{pol}	T_{sub}	S_{pos}	ϕ_1
S1	*0.3125*	*0.6875*	*0.125*	*0.375*
S2	*0.1*	*0.525*	*0.375*	*0.333*
S3	*0.125*	*0.375*	*0*	*0.166*

Table 3.3 Numeric values of sentiment context matrix (SCM)

Sentences	T_{pol}	T_{sub}	S_{pos}	*SCM*
S1	*0.3125*	*0.6875*	*0.125*	$\begin{array}{cc} \downarrow\rightarrow & \begin{array}{ccc} PS_1 & SS_1 & ZS_1 \end{array} \\ \begin{array}{c} PS_1 \\ SS_1 \\ ZS_1 \end{array} & \begin{bmatrix} 0.625 & 0.375 & 0.187 \\ 0.375 & 1.375 & 0.812 \\ 0.187 & 0.812 & 0.250 \end{bmatrix} \end{array}$
S2	*0.1*	*0.525*	*0.375*	$\begin{array}{cc} \downarrow\rightarrow & \begin{array}{ccc} PS_2 & SS_2 & ZS_2 \end{array} \\ \begin{array}{c} PS_2 \\ SS_2 \\ ZS_2 \end{array} & \begin{bmatrix} 0.2 & 0.625 & 0.475 \\ 0.625 & 1.05 & 0.9 \\ 0.475 & 0.9 & 0.75 \end{bmatrix} \end{array}$
S3	*0.125*	*0.375*	*0*	$\begin{array}{cc} \downarrow\rightarrow & \begin{array}{ccc} PS_3 & SS_3 & ZS_3 \end{array} \\ \begin{array}{c} PS_3 \\ SS_3 \\ ZS_3 \end{array} & \begin{bmatrix} 0.25 & 0.5 & 0.125 \\ 0.5 & 0.75 & 0.375 \\ 0.125 & 0.375 & 0 \end{bmatrix} \end{array}$

Based on the analysis of *Algorithm 3.6* we deduced that S1 is tagged as "non-sarcasm" having $\psi_1 > \tau_1$ while S2 and S3 are tagged as "sarcasm," having $\psi_1 \leq \tau_1$.

Table 3.4 Numeric values of composite sentiment score (ISS_1)

Sentences	***Applying the Balanced Hungarian method to get*** $\boldsymbol{ISS_1}$
S1	5
S2	2
S3	0.5

Table 3.5 Numeric values of optimal sarcasm score (ψ_1)

Sentences	$\boldsymbol{\phi_1}$	$\boldsymbol{ISS_1}$	$\boldsymbol{\psi_1}$	***Tag***
S1	*0.375*	5	2.239	*Non-Sarcasm*
S2	*0.333*	2	1.279	*Sarcasm*
S3	*0.166*	0.5	1.0634	*Sarcasm*

3.3 Proposed Method 2: Sarcasm Detection Based on Box-Cox Transformation

In this section, we discuss the proposed method, namely SDO-BCRT, using a Box-Cox transformation system.

Step 1: Sentiment Evaluation and Sentence Score Calculation

In the first step, we utilize SWN to obtain S_{pos} and S_{neg} senti scores using *Algorithm 3.2.* Apply the Cauchy distribution function to maximize and derive the global optimal solution for sentiment analysis.

The given optimization problem aims to obtain the sentence's optimal sentiment score, as shown in *Eqn. (3.5)*.

$$\left.\begin{aligned} \phi_2 = Maximise \; \frac{1}{\pi r\left[1+\left(\frac{(S_{pos}-S_{neg})^2}{r}\right)\right]} \\ Subject\ to, (S_{pos}, S_{neg}) \geq 0\ and\ r = \frac{(S_{pos}+S_{neg})}{2} \geq 0 \end{aligned}\right\} \tag{3.5}$$

The sentiment analysis has been done on the basis of the final calculated optimal sentiment score of a sentence, denoted as ϕ_2. Whether the sentence is positive or negative, sentiment is deduced based on ϕ_2 and sentiment threshold value (λ_2). The λ_2 is calculated using the Pearson correlation coefficient paradigm. *Eqn. (3.6)* gave the formula for the λ_2 value as 0.8.

$$\lambda_2 = \phi_{2_{mean}} \times \frac{\Sigma\left|S_{pos_i} - S_{pos_{mean}}\right| \left|S_{neg_i} - S_{neg_{mean}}\right|}{\sqrt{\Sigma\left(S_{pos_i} - S_{pos_{mean}}\right)^2 \Sigma\left(S_{neg_i} - S_{neg_{mean}}\right)^2}} \tag{3.6}$$

The S_{pos_i} and S_{neg_i} are the individual positive and negative sentiment scores, respectively. The $S_{pos_{mean}}$ and $S_{neg_{mean}}$ are the mean values of the positive and negative sentiment scores. $\phi_{2_{mean}}$ denoted the average of the optimal sentiment scores. The λ_2 is obtained as 0.8, as shown in *Table 3.6*. *Fig. 3.2* represents an architectural overview of SDO-BCRT for sarcasm detection using mathematical optimization techniques.

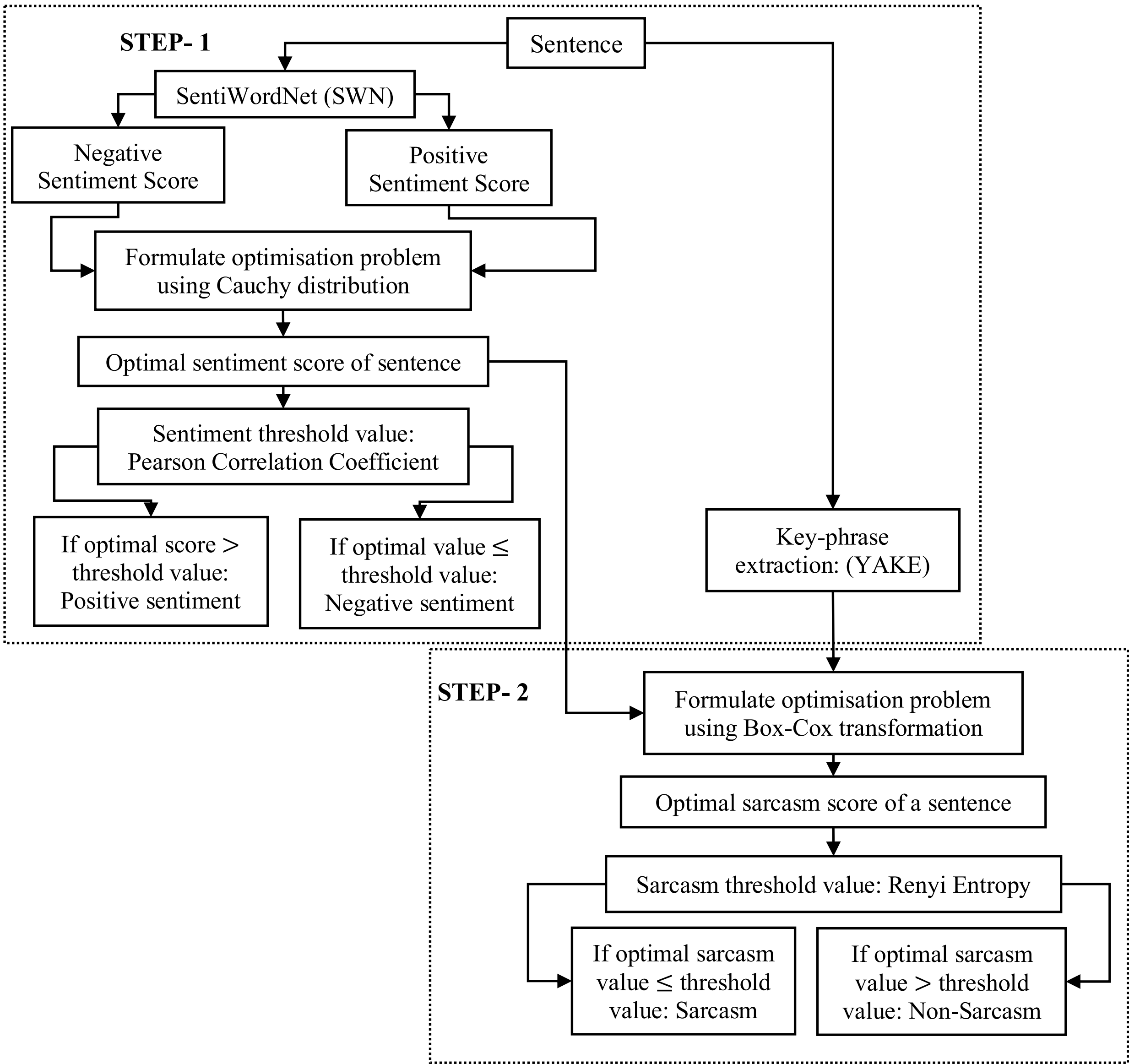


Fig. 3.2 Schematic diagram of SDO-BCRT for sarcasm detection

The λ_2 is then compared with ϕ_2 of the sentence, which helps to determine whether the sentence is a positive or negative sentiment. The decision criteria are to be established, which implies that if $\phi_2 > \lambda_2$, the sentence is tagged as "positive sentiment"; otherwise, it is tagged as "*negative sentiment*". *Algorithm 3.7* defines the procedure of sentiment analysis based on the defined unconstrained optimisation problem using the Cauchy distribution as the objective function.

Table 3.6 Threshold computation for sentiment evaluation using PCC

Parameters	***Values***
Sentiment threshold value ($\boldsymbol{\lambda_2}$)	*Dataset-1 [149]:* $\lambda_2 = 1.378 \times \frac{850.35}{\sqrt{1493.64 \times 1348.77}} = 0.8261 \approx 0.8$ *Dataset-2 [150]:* $\lambda_2 = 0.9005 \times \frac{10892.3}{\sqrt{10892.3 \times 14608.79}} = 0.7976 \approx 0.8$ *Dataset-3 [151]:* $\lambda_2 = 1.156 \times \frac{884.89}{\sqrt{1614.67 \times 1395.79}} = 0.8362 \approx 0.8$

ALGORITHM 3.7: Sentiment Evaluation using Cauchy Distribution in SDO-BCRT Method
Input: X − set of sentences, SWN ∈ SentiWordNet lexicon model *Output: positive and negative sentiment based on optimal sentiment score* (ϕ_2) 1. *Initialize for every sentence S repeat steps from 2 to 5.* 2. *Formulate an unconstrained optimisation problem using the Cauchy distribution:* $\phi_2 = Maximise \frac{1}{\pi r\left[1+\left(\frac{(S_{pos}-S_{neg})}{r}\right)^2\right]}$ *and subject to,* $(S_{pos}, S_{neg}) \geq 0$ *and* $r = \frac{(S_{pos}+S_{neg})}{2} \geq 0$ 3. *Output→ "optimal sentiment score* (ϕ_2)*"* 4. *If* $\phi_2 > \lambda_2 \rightarrow$ *"positive sentiment"* 5. *Else "negative sentiment"*

Step 2: Keyword Extraction using YAKE

In the second step, we use a key-phrase model to get the sarcasm indicator score with the highest value. We uproot the keyword with the highest value, denoted as k_{max}^1 in the proposed SDO-BCRT method using the YAKE model.

Step 3: Sarcasm Detection

In the third step, we detect sarcasm by maximizing the Box-Cox transformation function with the derived values obtained from the previous steps, resulting in an optimal sarcasm score (ψ_2). The objective function is to guide the constraints for learning patterns in order to get the optimal solution. The given maximised objective function with respect to specified constraints gives the optimal score of the sentence, as shown in *Eqn. (3.7)*.

$$\left.\begin{aligned} \psi_2 &= Maximise \quad r\left(\frac{\left|\phi_2{}^{k_{max}^1}-1\right|}{k_{max}^1}\right), \phi_2 \neq 0 \\ &= Maximise \ \log(\phi_2),\ \phi_2 = 0 \\ Subject\ to, \phi_2 &= \frac{1}{\pi r\left[1+\left(\frac{(S_{pos}-S_{neg})}{r}^2\right)\right]} \geq 0, k_{max}^1 \geq 0\ and\ r = \frac{(S_{pos}+S_{neg})}{2} \geq 0 \end{aligned}\right\} \tag{3.7}$$

Algorithm 3.8 defines the task of sarcasm detection utilising an unconstrained optimisation problem containing the Box-Cox transformation function.

ALGORITHM 3.8: Sarcasm Detection using the Proposed Method SDO-BCRT
Input: $\boldsymbol{\phi_2}$, $\boldsymbol{k_{max}^1}$, *average of* $\boldsymbol{S_{pos}}$ *and* $\boldsymbol{S_{neg}}$ *(*$\boldsymbol{r}$*)* *Output: sarcasm and non-sarcasm detection based on optimal sarcasm score (*$\boldsymbol{\psi_2}$*)* 1. *For each sentence, X repeats steps 2 to 5.* 2. *Formulate an unconstrained optimisation problem using the Box-Cox transformation:* $\boldsymbol{\psi_2} = Maximise \begin{cases} \boldsymbol{r\left(\frac{\left\|\phi_2{}^{k_{max}^1}-1\right\|}{k_{max}^1}\right),\ if\ \phi_2 \neq 0} \\ \boldsymbol{log(\phi_2),\ if\ \phi_2 = 0} \end{cases}$ *subject to,* $\boldsymbol{\phi_2} = \frac{\mathbf{1}}{\boldsymbol{\pi r}\left[\mathbf{1}+\left(\frac{(S_{pos}-S_{neg})}{r}^2\right)\right]} \geq \mathbf{0}$*,* $\boldsymbol{k_{max}^1} \geq \mathbf{0}$ *and* $\boldsymbol{r} = \frac{(S_{pos}+S_{neg})}{2} \geq \mathbf{0}$ 3. *Output → "optimal sarcasm score (*$\boldsymbol{\psi_2}$*)."* 4. *If* $\boldsymbol{\psi_2} \leq \boldsymbol{\tau_2}$ *→ "sarcasm"* *Else "non-Sarcasm"*

The final evaluation has been done based on a comparison between ψ_2 and the choice of sarcasm detection threshold value (τ_2) computed using Rényi entropy as 0.78. The calculation of τ_2 is shown in *Eqn. (3.8).*

$$\tau_2 = \frac{\frac{1}{1-t}\log\left(\Sigma_i\left(P\left((\psi_{2_i})\right)\right)^t\right)}{10} \tag{3.8}$$

Where $t = 0.77$, $P(\psi_{2_i})$ considered as the probability of taking on the value ψ_{2_i}. The calculated value of τ_2 is computed as 0.78. The calculated sarcasm detection threshold value $\tau_2 = 0.78$ using the rényi entropy mechanism, as shown in *Table 3.7.*

Table 3.7 Threshold computation using Renyi entropy

Parameters	***Values***
Sarcasm detection threshold value ($\boldsymbol{\tau_2}$)	*Dataset-1 [149]:* $\tau_2 = \frac{\frac{1}{1-0.7}log(1716.83)}{10} = 0.778 \approx 0.78$ *Dataset-2 [150]:* $\tau_2 = \frac{\frac{1}{1-0.7}log(1780.786)}{10} = 0.773 \approx 0.78$ *Dataset-3 [151]:* $\tau_2 = \frac{\frac{1}{1-0.7}log(1842.11)}{10} = 0.748 \approx 0.78$

The above comparison helps to determine whether the sentence is sarcastic or non-sarcastic. Based on the established decision criteria, it stipulates that if the ψ_2 is less than or equal to τ_2, the sentence is tagged as "*sarcasm*"; otherwise it is tagged as "*non-sarcasm".*

3.3.1 Numerical Illustration of Proposed Method: SDO-BCRT

This subsection discusses numerical examples to demonstrate the SDO-BCRT model. We consider two English comments, S1 and S2. The examples show the step-by-step depiction of the proposed method working for sarcasm detection and its inhibited intent learning from it.

S1: "Well done, Monty. Mark that up as your first ever 100% honest and accurate post here." ***S2: "Oh fantabulousss, another Sundays morning meeting just what I needed to make my weekend perfect."***

First, we calculate the optimal sentiment score of a sentence, denoted as ϕ_2 and then in step 2, we deduce the keyword with the highest value, denoted as k_{max}^1 using *Algorithm 3.7* and the

YAKE model, respectively. *Tables 3.8* and *3.9* show the computation of the given scores utilized in steps 1 and 2.

Table 3.8 Optimal sentiment score computation based on an unconstrained optimization technique

Sentences	S_{pos}	S_{neg}	ϕ_2	*Tag*
S1	*3.25*	*2.125*	0.0805	*Positive sentiment*
S2	*0.75*	*0.125*	0.8	*Positive sentiment*

Table 3.9 Computation of the highest key-phrase value

Sentences	*Key-phrase extraction using YAKE (k^1_{max})*	*Triggered context*
S1	*0.7807*	*"honest and accurate post"*
S2	*0.2973*	*"Sundays morning meeting just what I needed "*

Next, in step 3, we calculate the optimal sarcasm score (ψ_2) for sarcasm detection employing an unconstrained optimisation problem using the Box-Cox transformation. *Table 3.10* shows the computation of this score using *Algorithm 3.8.* In step 4, we deduce the inherent sarcasm kind in terms of showing the sarcastic comment either as callous or non-callous in nature shown in *Chapter 5*.

Table 3.10 Optimal sarcasm score computation based on unconstrained optimization technique using Box-Cox transformation

Sentences	ϕ_2	k^1_{max}	r	ψ_2	*Tag*
S1	*0.0805*	*0.7807*	*2.6875*	0.0960	*Non-sarcasm*
S2	*0.8*	*0.2973*	*0.4375*	0.03639	*Sarcasm*

Next, in step 3, we calculate the optimal sarcasm score (ψ_2) for sarcasm detection employing an unconstrained optimisation problem using the Box-Cox transformation. *Table 3.10* shows the computation of this score using *Algorithm 3.8.* In step 4, we deduce the inherent sarcasm kind in terms of showing the sarcastic comment either as callous or non-callous in nature shown in *Chapter 5*.

3.4 Proposed Method 3: Sarcasm Detection Based on Cauchy Distribution

In this section, we discussed the proposed methodology, (SD-SCOT) of sarcasm detection using the Cauchy distribution in detail. *Fig. 3.3* demonstrates the flowchart of the proposed (SD-SCOT) method.

Sarcasm detection involves three key steps in the proposed method, as follows.

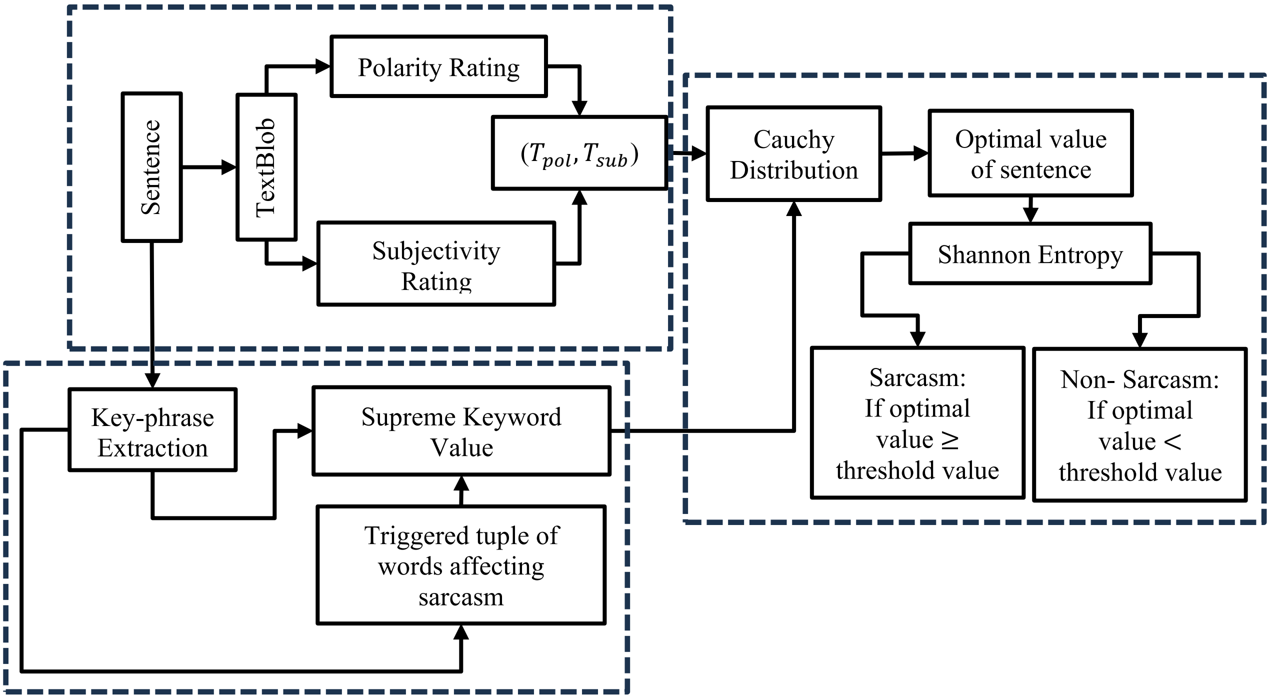


Fig. 3.1 Flowchart of the proposed method, SD-SCOT for sarcasm detection

Step 1: Sentiment Evaluation and Sentence Score Calculation

First, TextBlob has been used to get T_{pol} and T_{sub} scores using *Algorithm 1.1.* Derive the sentiment value (ϕ_3) by calculating the absolute sum of T_{pol} and T_{sub}. If $\phi_3 = |T_{pol} + T_{sub}| > \lambda_3$, the sentence is tagged as "positive"; or else it is a "negative" sentiment. The threshold value (λ_3) is calculated by taking the average values of T_{pol} ($avg_{T_{pol}}$) and T_{sub} ($avg_{T_{sub}}$) of all sentences. This is computed as 0.4. *Table 3.11* gives the value of λ_3 obtained over the four datasets.

Step 2: Keyword Extraction using RAKE

In the case of the proposed method, the calculation of the affected sentence marker gives the trigger tuple of words that affect sarcasm in a sentence. The system utilized the RAKE model and extracted the keyword with the highest score. The keyword with the highest value is the sentence's supreme keyword value (k_{max}^2).

Table 3.1 Parameter used in sentiment evaluation

Datasets	***Sentiment score threshold value***
Dataset-1 [149]	$\lambda_3 = \frac{avg_{T_{pol}}+avg_{T_{sub}}}{2} = \frac{0.3994+0.4388}{2} = 0.4191 \approx 0.4$
Dataset-2 [150]	$\lambda_3 = \frac{avg_{T_{pol}}+avg_{T_{sub}}}{2} = \frac{0.4776+0.3220}{2} = 0.3998 \approx 0.4$
Dataset-3 [151]	$\lambda_3 = \frac{avg_{T_{pol}}+avg_{T_{sub}}}{2} = \frac{0.5543+0.2511}{2} = 0.4027 \approx 0.4$

Step 3: Sarcasm Detection

In the third step, we detect sarcasm by maximizing the Cauchy distribution function to get the optimal sarcasm score (ψ_3). In the proposed method, the Cauchy distribution is used as a maximised objective function. *Eqn. (3.9)* demonstrates the Cauchy distribution formula used in the proposed method.

$$\left.\begin{array}{c} \psi_3 = \text{ Maximise } \frac{1}{\pi k_{max}^2\left(1+\left(\frac{|T_{pol}-T_{sub}|^2}{k_{max}^2}\right)\right)} \\ Subject\ to, (T_{pol}, T_{sub}) \geq 0\ and\ k_{max}^2 \geq 0 \end{array}\right\} \tag{3.9}$$

The ψ_3 is the required optimal sarcasm score of the optimized Cauchy distribution. *Algorithm 3.9* defines the step-by-step implementation of getting the ψ_3 of each sentence obtained from the specified optimization function.

These optimal values obtained for each sentence of the dataset are used to evaluate the threshold value (τ_3) for sarcasm detection as shown in *Eqn. (3.10).*

$$\tau_3 = -\sum_{i=1}^{i=m} \psi_{3_i} * log(\psi_{3_i}) \tag{3.10}$$

The proposed method establishes the threshold value for sarcasm detection τ_2, using Shannon Entropy, yielding a value of 0.05, shown in *Table 3.12.*

ALGORITHM 3.9: Sarcasm detection using the SD-SCOT
Input: Total number of sentences $\boldsymbol{X_i}$ $\boldsymbol{1 \le i \le m}$, $\boldsymbol{T_{pol}}$, $\boldsymbol{T_{sub}}$, $\boldsymbol{k^2_{max}}$ *Output: sarcastic and non-sarcastic based on the optimal value of a sentence (*$\boldsymbol{\psi_3}$*)* 1. *Compute the Cauchy distribution as a maximised objective function:* $\boldsymbol{\psi_3}$ = *Maximise* $\frac{\mathbf{1}}{\boldsymbol{\pi SKV}\left(\mathbf{1}+\left(\frac{\left\lvert \boldsymbol{T_{pol}-T_{sub}}\right\rvert^{2}}{\boldsymbol{k^2_{max}}}\right)\right)}$ *and Subject to: (*$\boldsymbol{T_{pol}}, \boldsymbol{T_{sub}}$*)* $\ge \mathbf{0}$ *and* $\boldsymbol{k^2_{max}} \ge \mathbf{0}$ 2. *If* $\boldsymbol{\psi_3} \ge \boldsymbol{\tau_3} \rightarrow$ *"sarcastic"* 3. *Else "non-sarcastic"*

Table 3.12 Parameter used in SD-SCOT for sarcasm detection

Datasets	***Sarcasm detection threshold value***
Dataset-1 [149]	$\tau_3 = -\sum \psi_3 * log(\psi_3) = 0.05046 \approx 0.05$
Dataset-2 [150]	$\tau_3 = -\sum \psi_3 * log(\psi_3) = 0.04967 \approx 0.05$
Dataset-3 [151]	$\tau_3 = -\sum \psi_3 * log(\psi_3) = 0.05180 \approx 0.05$

The computed τ_3 is then compared with the ψ_3 of the sentence. If the optimal value of the sentence $\psi_3 \ge \tau_3$, the sentence is tagged as "sarcastic"; otherwise, it is "non-sarcastic".

3.4.1 Numerical Illustration of the Proposed Method: SD-SCOT

We have taken two sample sentences, i.e., S1 and S2, from our dataset to illustrate the implementation of the proposed method.

S1: I know right? It's literally impossible to record a stream!, Lol, I can feel the pain mann.. ***S2: Well there are apps that can delete personal info and even send you location of the person using / send you their. URL***

In step 1, we first calculate the T_{pol} and T_{sub} of each sentence using TextBlob. *Table 3.13* shows the computed PR and SR scores, respectively, with their respective sentiment tag.

Table 3.13 Numeric scores of polarities and subjectivity rating

Sentences	*Parameters*	*Values*	ϕ_3	*Sentiment Tag*
S1	PR_1, SR_1	*0.15, 0.316*	*0.466*	*Positive*
S2	PR_2, SR_2	*0.02,0.130*	*0.150*	*Negative*

Based on the sentiment classification criteria in step 1, we deduced the sentence's sentiment classification as either positive or negative. Subsequently, in step 2, the calculation of the supreme keyword value (k_{max}^2) using RAKE, which gives the trigger tuple of words that affect sarcasm, is performed, as shown in *Table 3.14.* Since we are using it as a pre-defined model, the computed score of $\mathrm{k}_{\mathrm{max}}^2$ is shown in *Table 3.14*.

Table 3.14 Numeric score computation using RAKE

Sentences	k_{max}^2	*Triggered tuple of words*
S1	*4*	*"record a stream Lol, I can feel the pain"*
S2	*9*	*"apps that can delete personal info"*

In the final step, the task of sarcasm detection, has been performed using SD-SCOT a probabilistic optimization technique. *Table 3.15* shows the formulation of the optimization problem and getting the optimal solution (ψ_3) through it. Based on the computed threshold value of sarcasm detection (τ_3), we compared it with the sentence's optimal value (ψ_3).
For S1, the optimal value of sentence $\psi_3 \geq \tau_3$, the sentence is tagged as "sarcastic"; and for S2, $\psi_3 < \tau_3$, it is tagged as "non-sarcastic." Therefore, in this way, we perform sarcasm detection alongside sentiment analysis using mathematical optimization techniques.

Table 3.15 Computation of the optimal value of a sentence using the SD-SCOT technique

Sentences	T_{pol}, T_{sub}	k_{max}^2	ψ_3	*Tag*
S1	*0.15, 0.316*	*4*	0.07959	*Sarcasm*
S2	*0.02, 0.130*	*9*	0.03543	*Non-sarcasm*

3.5 Proposed Methods 4 and 5: Sarcasm Detection Based on Logistic and Weibull Distributions

In this section, we discuss LDL-SDTE and SWD-SEO proposed methods for sarcasm detection in detail using logistic distribution and Weibull distribution. The proposed methods follows two steps to perform sarcasm detection.

Step 1: Sentiment Evaluation and Emotion Analysis

In the first step, we use VADER sentiment analysis, Text2Emotion, and Textacy to obtain sentence sentiment, emotion value, and target word value. The fundamental component of the proposed methods, LDL-SDTE and SWD-SEO, is used to capture the emotions from texts using the "Tex2emotion" Python library. This library identifies emotions expressed in emotive score analysis statements, denoted as $\mathrm{Emot}_{\mathrm{avg}}$. For every distinct emotion, we establish an emotion embedding score, referred to as $emot_{embed}$. Five emotions are present in our library: surprise, fear, anger, sadness, and happiness. We derive two emotions: positive emotion, which involves an average score $\frac{Surprise+Happy}{2}$, and negative emotion, which involves a mean score $\frac{Angry+Fear}{2}$, $\frac{Sad+Fear}{2}$, and $\frac{Sad+Angry}{2}$. The schematic representation of LDL-SDTE is depicted in *Fig. 3.4.* The schematic diagram of the SWD-SEO is depicted in *Fig. 3.5.*

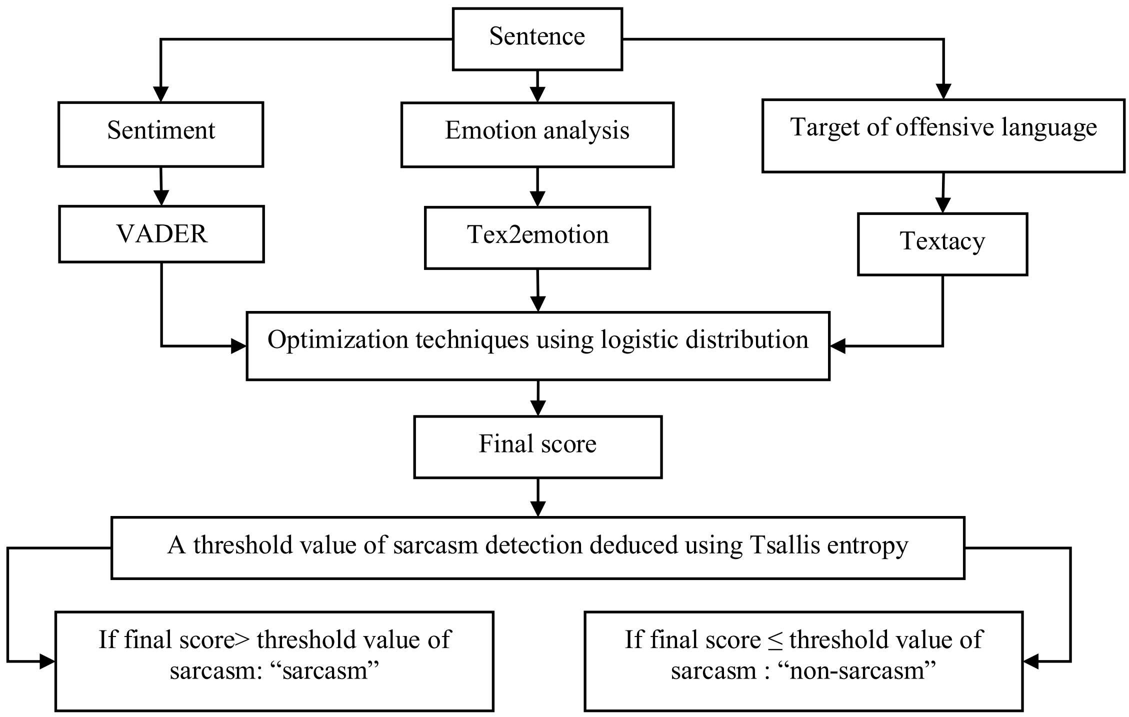


Fig. 3.4 Flowchart of the proposed LDL-SDTE method for sarcasm detection

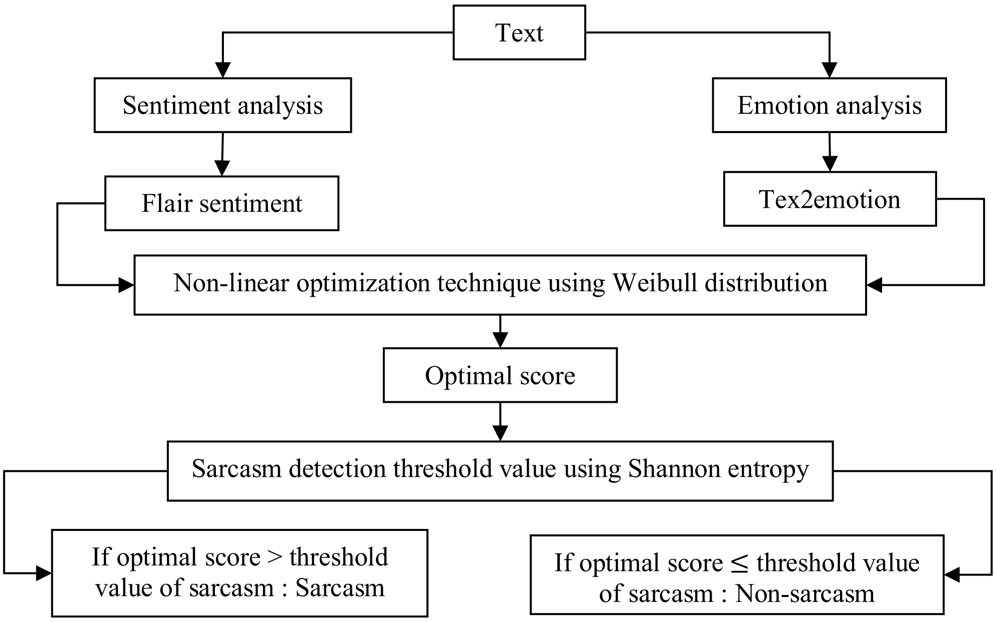


Fig. 3.5 Flowchart of the SWD-SEO method for sarcasm detection

The proposed LDL-SDTE method for sarcasm detection uses the standard deviation of the sentiment compound score, denoted as $\mathrm{senti\_conf_{SD}}$, which is obtained from VADER. The target of offensive language has been obtained using a keyword extraction method, i.e., Textacy, a Python library. The targeted offensive word is marked by the highest score given to the proposed sarcasm identification framework. Through the aid of flair sentiment, depending on the sentiment vocabulary tool, the feeling of the specified text is detected in SWD-SEO. We subsequently import the sentence sentiment confidence score, denoted as F_{conf}, onto the SWD-SEO method for sarcasm detection.

Step 2: Sarcasm Detection using Proposed Methods: LDL-SDTE and SWD-SEO

In the second step, the input of these values into a maximized logistic distribution function is done to derive the optimal sarcasm score for sarcasm detection. The logistic distribution used in the LDL-SDTE method for sarcasm detection is shown in *Eqn. (3.11)* to get the optimal sarcasm score (ψ_4).

$$\psi_4 = \frac{e^{-\frac{(V_{comp}-Emot_{avg})}{F_{conf}}}}{F_{conf_i}\left(1+e^{-\frac{(V_{comp}-Emot_{avg})}{F_{conf}}}\right)^2} \tag{3.11}$$

The sarcasm detection threshold value (τ_4) is calculated using the Tsallis entropy distribution, which comes up with a value of 0.5, a fixed score. The τ_4 in *Eqn. (3.12)* is calculated using the Tsallis entropy structure, such as with $e_1 = 3.7$ as a fixed value metric.

$$\tau_4 = \frac{1}{1-e_1}\left(1 - \sum_i \left(P(\psi_{4_i})\right)^{e_1}\right) = 0.5 \tag{3.12}$$

The statement is deduced to be a sarcastic label if ψ_4 is greater than τ_4 and non-sarcastic if the ψ_4 is less than or equal to τ_4.

Meanwhile, Weibull distribution is used as a probabilistic distributional optimization in the proposed SWD-SEO method for sarcasm detection. It is applied to describe data in terms of dependability and the existence of variables. Weibull distribution's general formula with three parameters is shown in *Eqn. (3.13)* for sarcasm detection.

$$\psi_5 = \frac{emot_{embed}}{F_{conf}{}^{emot_{embed}}}\left(F_{conf}{}^{emot_{embed}-1}\, e^{-\left(\frac{emot_{embed}}{F_{conf}}\right)^{emot_{embed}}}\right) \tag{3.13}$$

The ψ_5, termed as the optimal sarcasm score, is the defined distribution obtained from the Weibull distribution. The threshold value of sarcasm (τ_5) is defined as a critical value where the defined function changes its trait and depicts the point at which a decision is taken. The $\tau_5 = 0.5$, taken as a parameter calculated using the Shannon entropy structure in *Eqn. (3.14)*.

$$\tau_5 = -\sum_{i=1}^{i=m} \rho_i(c_i) \times \ln(\rho_i\,(c_i)) = 0.5 \tag{3.14}$$

The statement is deduced to be a sarcastic label if the ψ_5 is greater than τ_5 and non-sarcastic if ψ_5 is less than or equal to τ_5. *Algorithms 3.10* and *3.11* give the perfect visualisation of the proposed methodologies of LDL-SDTE and SWD-SEO for sarcasm detection in English text.

3.5.1 Numerical Illustration of Proposed Method: SWD-SEO

The exemplar sentences have been taken for the proposition of the proposed method, the SWD-SEO system. The sentences taken here are from the datasets utilised in this study.

S1: "The first professional baseball team was from an Ohio city."
S2: "speak up and give back if you want the economy to improve."

ALGORITHM 3.10: Sarcasm detection using LDL-SDTE
Input: $\mathrm{Emot}_{\mathrm{avg}}$, F_{conf}, V_{comp}, τ_4 *Output: sarcasm and non-sarcasm (ψ_4)* 1. *Calculate the compound score of sentences V_{comp} using the VADER sentiment library* 2. *Using the Tex2emotion library to obtain the emotion analysis score* 3. *Calculate the Logistic distribution based on the given parameters: L1 =* $e^{-\frac{(V_{comp}-Emot_{avg})}{F_{conf}}}$, *L2 =* $\left(1+e^{-\frac{(V_{comp}-Emot_{avg})}{F_{conf}}}\right)^2$ *and L3 =* $senti_{conf_{SD_i}}$ 4. *Define:* $\psi_4 = \frac{L1}{L2 \times L3}$ 5. *Calculate the Logistic distribution score = calculate_logistic_distribution* $(\psi_4;\ Emot_{avg}, F_{conf})$ 6. *Compare the calculated score with the threshold value to determine sarcasm* *if $\psi_4 > \tau_4$: return "sarcasm"* *Else "non-sarcasm"* 7. *End*

ALGORITHM 3.11: Sarcasm detection using SWD-SEO
Input: $emot_{embed}$, F_{conf} *Output: sarcasm and non-sarcasm based on notch-up score (ψ_5)* 1. *Calculate the Weibull distribution based on the given parameters* *para1 =* $\frac{emot_{embed}}{{F_{conf}}^{emot_{embed}}}$, *para2 =* ${F_{conf}}^{emot_{embed}-1}$ *and para3 =* $e^{-\left(\frac{emot_{embed}}{senti_{conf_{SD}}}\right)^{emot_{embed}}}$ 2. *Define ρ:ρ = para1 * (para2 * para3)* 3. *Define detect sarcasm ($emot_{embed}$, F_{conf}): Calculate the Weibull distribution score; ψ_5 = calculate_weibull_distribution ($emot_{embed}$, F_{conf})* 4. *Compare the calculated score with the threshold to determine sarcasm* *if $\psi_5 > \tau_5$: return "Sarcasm"* *Else "Non-sarcasm"* 5. *End*

In step 1, we use the "Tex2emotion" Python library to extract emotions from sentences. These embeddings are termed emotion embedding scores, referred to as $\mathrm{emot_{embed}}$. This process ensures that the scores are aligned within a consistent range. *Table 3.16* depicts the presence of emotion using the Tex2emotion mechanism.

Table 3.16 Emotion embedding score computation

Sentences	$\boldsymbol{emot_{embed}}$	***Emotion tag***
S1	*0.9032*	*Surprise*
S2	*0.9710*	*Happy*

Table 3.17 Sentiment evaluation of sentences

Sentences	$\boldsymbol{F_{conf}}$	***Sentiment tag***
S1	*0.5689*	*Negative*
S2	*0.9993*	*Positive*

In step 2, through the aid of flair sentiment, the feeling of the specified text is detected using both positive and negative factors, along with the sentence confidence score. Subsequently, we import the sentence sentiment confidence score, denoted as F_{conf}, shown in *Table 3.17* for sarcasm identification.

In step 3, the optimised Weibull distribution is applied to get the notch-up score (ψ_5). This defined distribution is utilized for sarcasm detection, as shown in *Table 3.18.*

Table 3.18 Sarcasm detection via SWD-SEO

Sentences	ψ_5	***Context nature***
S1	*0.7168*	*Sarcasm*
S2	*0.1055*	*Non-Sarcasm*

The statement S1 is deduced to be a sarcastic label where ψ_5 is greater than τ_5, whereas S2 is labelled as non-sarcastic as it is less than or equal to τ_5.

3.5.2 Numerical Illustration of Proposed Method: LDL-SDTE

Two sentences are considered for working with the proposed LDL-SDTE method for the task of sarcasm detection. We have taken examples from the datasets utilised in the proposed method.

S1: "No, no, man, I found a magic way to grind 4200 mobs worth five reps a piece in a single afternoon; it's a joke, Loll!"   ***S2: "Man, the refs are surely giving the Flames a good ass fucking tonight."***

In step 1, the emotional analysis of the text has been done as shown in *Table 3.19*. We evaluate the emotional scores termed emotive score, denoted as $\mathrm{Emot}_{\mathrm{avg}}$, for each unique feeling, using Tex2emotion.

Table 3.19 Emotive scores representation

Sentences	$\boldsymbol{Emot_{avg}}$	*Emotion Tag*
S1	*0.2*	*Positive*
S2	*0.198*	*Negative*

In step 2, we analyse the sentiment using the VADER analyzer model. The sentiment compound score is computed, where the suggested framework uses its standard deviation, $\mathrm{senti\_conf}_{\mathrm{SD}}$. *Table 3.20* gives the sentiment evaluation based on $\mathrm{senti}_{\mathrm{conf}_{\mathrm{SD}}}$.

Table 3.20 Sentiment evaluation via VADER analysis

Sentences	$\boldsymbol{V_{neg}}$	$\boldsymbol{V_{pos}}$	$\boldsymbol{V_{comp}}$	$\boldsymbol{F_{conf}}$	*Sentiment Tag*
S1	*0.302*	*0.349*	*0.1511*	*0.3133*	*Positive*
S2	*0.398*	*0.183*	*0.2134*	*0.1223*	*Negative*

In step 3, the target of offensive language is performed using the Python library textacy to get the triggered context. *Table 3.21* depicts the offensive language presence with the help of the keyword extractor utilised.

In step 4, the sarcasm detection task is performed using the Logistic Distribution, which represents the likelihood distribution model. Based on this model, we deduce the task of sarcasm detection. *Table 3.22* shows the numerical representation.

Table 3.21 Emotive scores representation

Sentences	*Triggered context*	*Target*
S1	*"magic way to grind 4200 mobs."*	*Non- Offensive*
S2	*"good as fucking"*	*Offensive*

Table 3.22 Sarcasm Detection Task Using Logistic Distribution

Sentences	ψ_4	*Tag*
S1	*1.345*	*Sarcasm*
S2	*1.062*	*Sarcasm*

3.6 Experiment and Result Analysis

In this section, we present the analysis of the different datasets utilized in the proposed methods. The performance of proposed methods over these datasets are compared with existing state-of-the-art methods, as illustrated in *subsections 3.6.2* to *3.6.9*, respectively. Several metrics are analysed, and their efficacies are inspected in this subsection. We applied all proposed methods to various English language datasets and analysed the encouraging validation and ablation results.

3.6.1 Dataset used to evaluate the by Proposed Methods

Table 3.23 represents the statistics of the datasets used by the five proposed methods. This involves the utilization of three benchmark datasets for the task of sarcasm detection. The details of each of these datasets are given in *Chapter 2,* "*Literature review*," in *section 2.5*.

Table 3.23 Statistical variations of datasets

S. No.	*Datasets*	*Renamed as*	*Total number of sentences*
1.	*SARC [149]*	*Dataset-1*	*12,257*
2.	*Reddit comments [152]*	*Dataset-2*	*7,895*
3.	*Semeval 2018 task [151]*	*Dataset-3*	*11,025*

3.6.2 Performance Evaluation of Proposed Methods on Dataset-1

We evaluated five proposed methods using accuracy (A), precision (P), recall (R), and F1-score (F1) metrics. The proposed methods, LDL-SDTE, SWD-SEO, SOBH-REB, SDO-

BCRT, and SD-SCOT, demonstrated significantly superior performance compared to existing sarcasm detection methods across these evaluation metrics. Among them, LDL-SDTE and SWD-SEO achieved the highest scores, each exceeding 0.95 across all metrics. This reflects an average improvement of approximately 9.8% in accuracy, 10.2% in precision, 11.4% in recall, and 10.6% in F1 score over existing methods such as CNN-BiLSTM [18], SVM [95], and BERT-CNN [1]. The SOBH-REB method also outperformed most of the BERT-based baselines (Res-BERT+Att-BERT [11], SAWS [94], BERT-CNN [1]) by an average margin of 7.5% in F1 score. Here, proposed methods SDO-BCRT and SD-SCOT offered balanced gains across all metrics, maintaining performance levels above 0.90.

In contrast, methods like CBCP [54], ELMo [7], and CNN-LSTM-DNN [5] exhibited the weakest results, with scores below 0.70 in most metrics. Intermediate methods such as SVM [95], CNN-BiLSTM [18], and even BERT-based techniques like BERT-CNN [1], BERT-SGCAT [35] showed decent performance but still notably lower than the proposed methods. These findings confirm that the proposed methods are more accurate, consistent, and reliable, with a marked reduction in false predictions, making them highly effective for real-world sarcasm detection tasks. *Fig. 3.6* represents the evaluation metric of the proposed methods and the existing state-of-the-art methods on dataset-1.

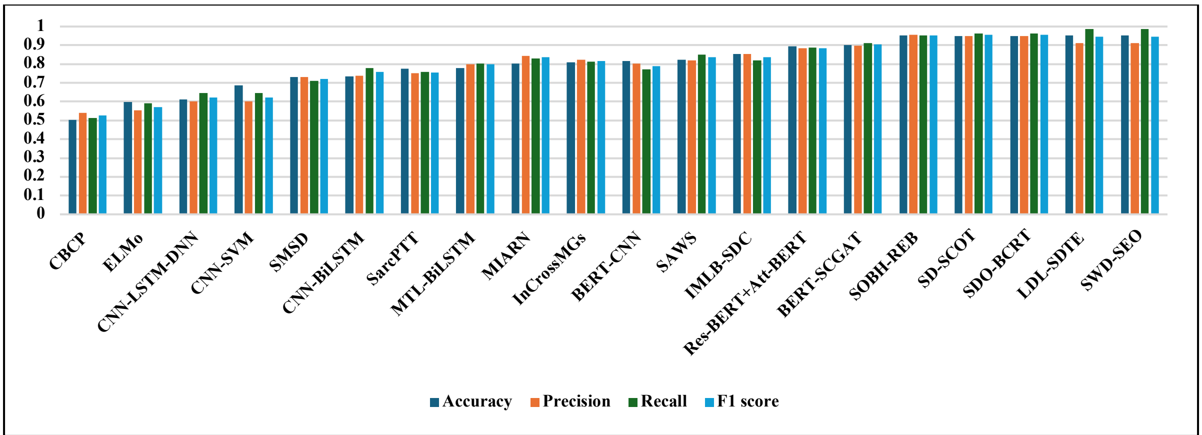


Fig. 3.6 Comparison plot of proposed methods, SDO-BCRT, SOBH-REB, SD-SCOT, LDL-SDTE, and SWD-SEO with existing state-of-the-art methods on the dataset-1 [149]

Overall, the proposed methods offered higher accuracy and precision with sustained high recall and F1 scores, making them more reliable for real-world sarcasm detection methods.

3.6.3 Performance Evaluation of Proposed Methods on Dataset-2

The proposed methods (SDO-BCRT, LDL-SDTE, SWD-SEO, and SD-SCOT) consistently outperformed all existing state-of-the-art methods in terms of accuracy, precision, recall, and

F1 score. The proposed SDO-BCRT and SD-SCOT methods maintained performance scores above 0.92, outperforming models such as EfficientNet [154] and ADGCN-BERT [155] by 8.5-10.3% across metrics. Among them, proposed SWD-SEO and LDL-SDTE methods achieved the highest metrics, each recording values above 0.95 across all evaluation criteria. The proposed methods marked an average improvement of 10.4% in accuracy, 11.2% in precision, 12.3% in recall, and 11.7% in F1 score over traditional methods like Bi-LSTM [27], CNN [156], CNN-LSTM-DNN [5], and even advanced BERT-based variants such as SCGAT-BERT [35] and RCNN-BERT [110].

SD-SCOT showed significant gains with an average increase of 9.1% in F1 score compared to methods like M2Seq2Seq [157] and Att-BERT-CNN [11]. In contrast, state-of-the-art methods like Bi-LSTM [158] and CNN [156] scored below 0.70, indicating their limitations in handling the complexity of sarcastic expressions. These results validated that the proposed methods are not only more accurate but also deliver a better balance between false positives and false negatives, making them effective for sarcasm detection. *Fig. 3.7* represents the evaluation metric of the proposed methods SDO-BCRT, SD-SCOT, LDL-SDTE, SWD-SEO, and existing state-of-the-art methods on dataset-2.

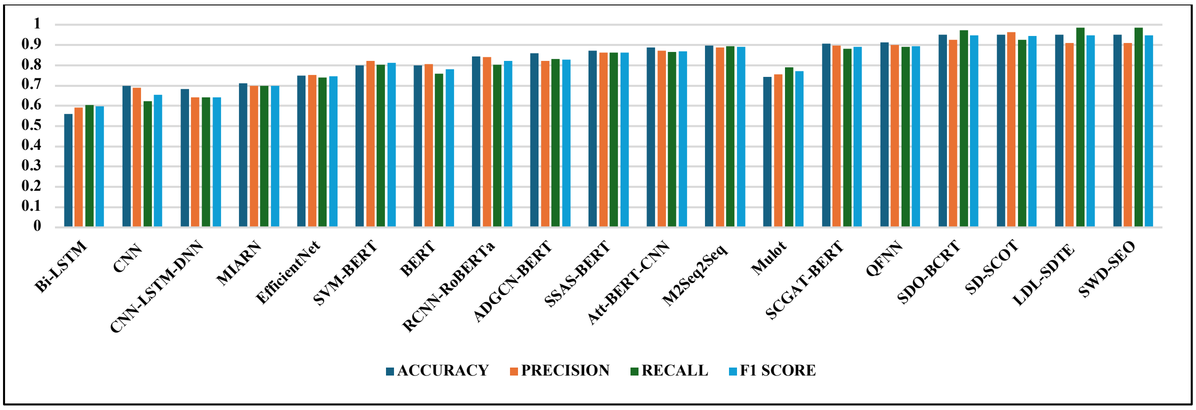


Fig. 3.7 Comparison plot of proposed methods, SDO-BCRT, SOBH-REB, SD-SCOT, LDL-SDTE, and SWD-SEO with existing state-of-the-art methods on the dataset-2 [150]

3.6.4 Performance Evaluation of Proposed Methods on Dataset-3

For dataset-3, we can see that the proposed methods SDO-BCRT, SOBH-REB, LDL-SDTE, SWD-SEO, and SD-SCOT achieved significantly better performance compared to all state-of-the-art methods. The proposed SOBH-REB and SDO-BCRT methods demonstrated superior performance with values exceeding 0.93, reflecting gains of 9.2%-10.6% across metrics compared to models like RCNN-BERTa [110] and ADGCN-BERT [155]. Specifically, LDL-SDTE and SWD-SEO scored the highest, with all metrics consistently above 0.95, indicating an average improvement of 10.8% in accuracy, 11.5% in precision, 12.1% in recall, and 11.6% in F1 score over methods like MIARN [74], CNN-LSTM-DNN [5], Bi-LSTM [158], and even more advanced architectures such as Att-BERT-CNN [11] and SCGAT-BERT [87]. SD-SCOT performed equally well, maintaining high metric scores with minimal variance. In contrast, existing methods such as MIARN [74] and CNN-LSTM-DNN [5] showed poor performance, with F1 scores dropping below 0.65, underscoring their limitations in capturing the nuanced patterns of sarcasm. This analysis shows that consistent improvements in terms of accuracy, precision, recall and F1 score across all proposed methods with confirmed robustness and effectiveness in sarcasm detection tasks. *Fig. 3.8* represents the evaluation metric for the proposed methods (SDO-BCRT, SOBH-REB, LDL-SDTE, SWDSEO, and SD-SCOT) compared to existing state-of-the-art methods on dataset-3.

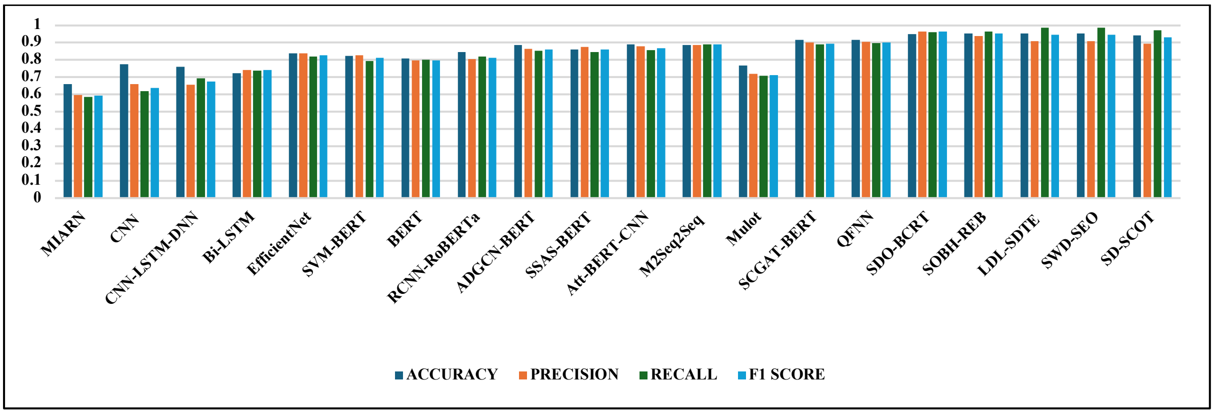


Fig. 3.8 Comparison plot of proposed methods, SDO-BCRT, SOBH-REB, SD-SCOT, LDL-SDTE, and SWD-SEO with existing state-of-the-art methods on the dataset-3 [151]

In conclusion, we see that all five proposed methods outperform other state-of-the-art methods in terms of evaluation metrics and successfully show beating performances compared to well-known techniques. Notably, these improvements help enhance results and lead to a positive rise in processing time.

3.7 Discussion

This section discussed the five proposed methods with certain suitable findings. In this section, we presented the statistical validation tests and their effectiveness for all the proposed methods. Additionally, we discuss the computation cost of each proposed method. In contrast to this, we present the challenges and ablation studies posed by the proposed methods for sarcasm detection.

3.7.1 Statistical Validation Tests Performed on Five Proposed Methods

3.7.1.1 Statistical validation using Z-test

A Z-test has been performed to compare the proportions of two populations, one from the SARC dataset and the other from Twitter comments. Two distinct samples were extracted from these datasets. The test analyzed the proportions (c_1 and c_2) under the null hypothesis (H_0) and alternative hypothesis (H_a), with the results summarized in *Tables 3.24, 3.25,* and *3.26*.

Table 3.24 Two Proportion Z-Test statistics across datasets

	SOBH-REB		***SDO-BCRT***	
Parameters	*Sample 1*	*Sample 2*	*Sample 1*	*Sample 2*
Sample size (N)	*1045*	*767*	*2557*	*1277*
Sample proportion (c)	*0.8296*	*0.8226*	*0.8384*	*0.8324*
Favorable cases (X)	*867*	*631*	*2144*	*1063*
P	*0.8260*		*0.8364*	
z	*0.3883*		*0.4735*	
Hypothesis status	H_0 *is not rejected*		H_0 *is not rejected*	

For two population proportions (c_1 and c_2), null hypotheses ($\mathrm{H_0}$) and alternative hypotheses ($\mathrm{H_a}$) are defined as follows:

H_0: $c_1 = c_2$, i.e., the accuracy score of sample 1 = accuracy score of sample 2

H_a: $c_1 \neq c_2$, i.e., the accuracy score of sample 1 ≠ , the accuracy score of sample 2

In the proposed method, the condition of the null hypothesis is accepted when the critical range z-value at $\alpha = 0.05$ is meet shown in *Eqn. (3.15).*

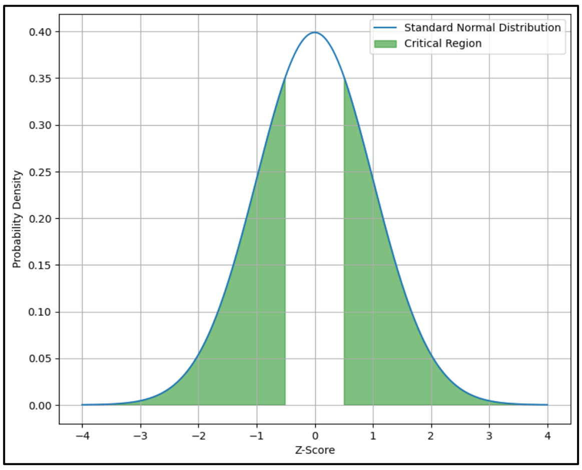
$$P = \frac{X_1+X_2}{N_1+N_2} \tag{3.15}$$

Table 3.25 Two Proportion Z-Test statistics across datasets

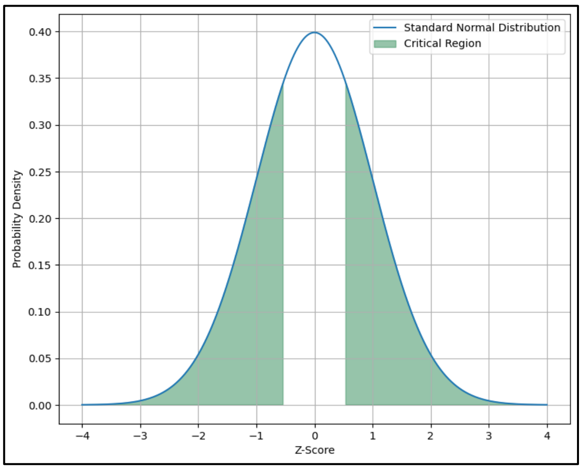
	SD-SCOT	
Parameters	*Sample 1*	*Sample 2*
Sample size (N)	*976*	*468*
Sample proportion (c)	*0.9262*	*0.8482*
Favorable cases (X)	*904*	*397*
P	*0.9010*	
z	*4.640*	
Hypothesis status	H_0 *is not rejected*	

Table 3.26 Two Proportion Z-Test statistics across datasets

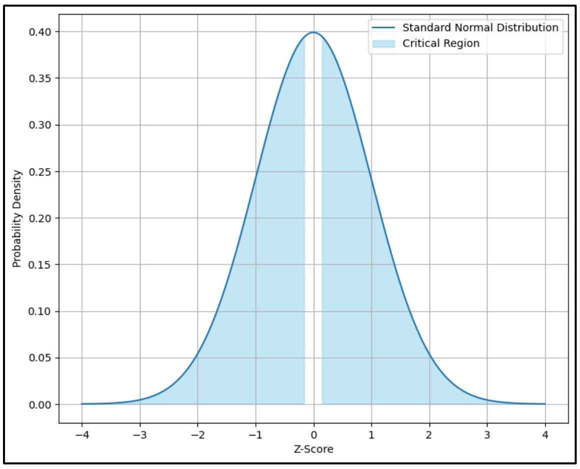
	LDL-SDTE		***SWD-SEO***	
Parameters	*Sample 1*	*Sample 2*	*Sample 1*	*Sample 2*
Sample size (N)	*1013*	*735*	*1100*	*666*
Sample proportion (c)	*0.9536*	*0.9224*	*0.9154*	*0.8843*
Favorable cases (X)	*966*	*678*	*1007*	*589*
P	*0.9405*		*0.9037*	
z	*2.718*		*2.145*	
Hypothesis status	H_0 *is not rejected*		H_0 *is not rejected*	



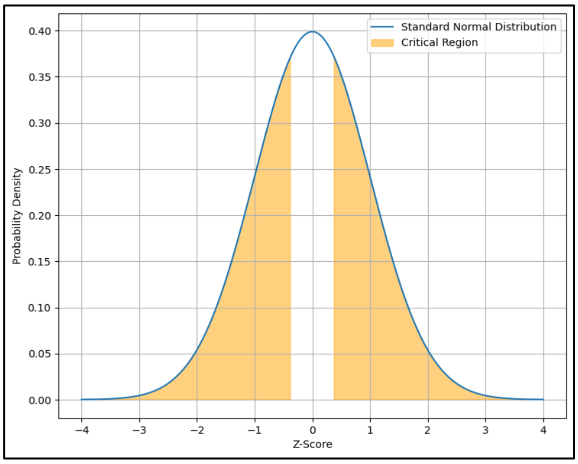
(a)



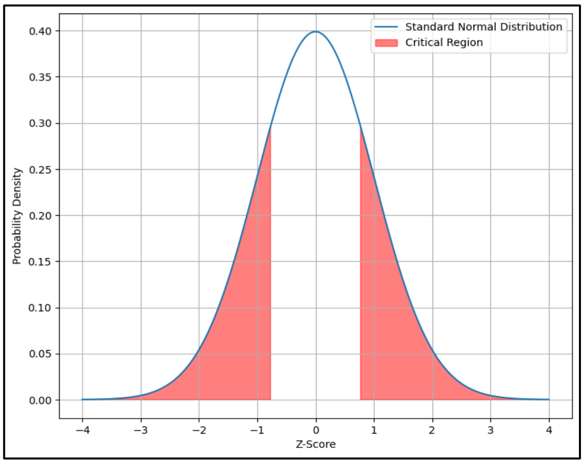
(b)



(c)



(d)



(e)

Fig. 3.9 Statistical validation of the proposed (a) SOBH-REB, (b) SDO-BCRT, (c) SD-SCOT, (d) LDL-SDTE, and (e) SWD-SEO methods using the z-test

It gives the value of the two sum proportions is less than the computed z score, also called z-statistics in *Eqn. (3.16)*.

$$z = \frac{|c_1 - c_2|}{\sqrt{P(1-P)(\frac{1}{N_1}+\frac{1}{N_2})}} \tag{3.16}$$

Figs. 3.9 (a), (b), (c), (d), and *(e)* are the critical regions of the hypothesis depicted graphically. This finding suggested that the consistency of the proposed method's accuracy score holds across diverse sample sizes derived from various datasets.

3.7.1.2 Statistical Validations using Kolmogorov-Smirnov (KS) Test

In the Kolmogorov-Smirnov (KS) test, we compared two independent samples taken from the two datasets. The aim is to determine whether the two samples have similar distributions without assuming any change in the particular distribution of the data based on the empirical cumulative distribution function (ECDF). *Table 3.27* shows how this test works.

Table 3.27 Two Proportion KS-Test statistics across datasets

	SOBH-REB		***SDO-BCRT***	
Parameters	*Sample 1*	*Sample 2*	*Sample 1*	*Sample 2*
Sample size (N)	*1120*	*998*	*1147*	*828*
Sample proportion (c, ECDF)	*0.9419*	*0.9428*	*0.9546*	*0.9553*
Favorable cases (X)	*1055*	*941*	*1095*	*791*
D- value (D_{N_1,N_2})	*0.9428*		*0.9553*	
D_α	*0.5988*		*0.3101*	
Hypothesis status	H_0 *is not rejected*		H_0 *is not rejected*	

For two population proportions (c_1 and c_2), null hypotheses ($\mathrm{H_0}$) and alternative hypotheses ($\mathrm{H_a}$) are defined as follows:

$$H_0: c_1 = c_2, \text{ i.e., accuracy of sample 1} = \text{accuracy of sample 2}$$

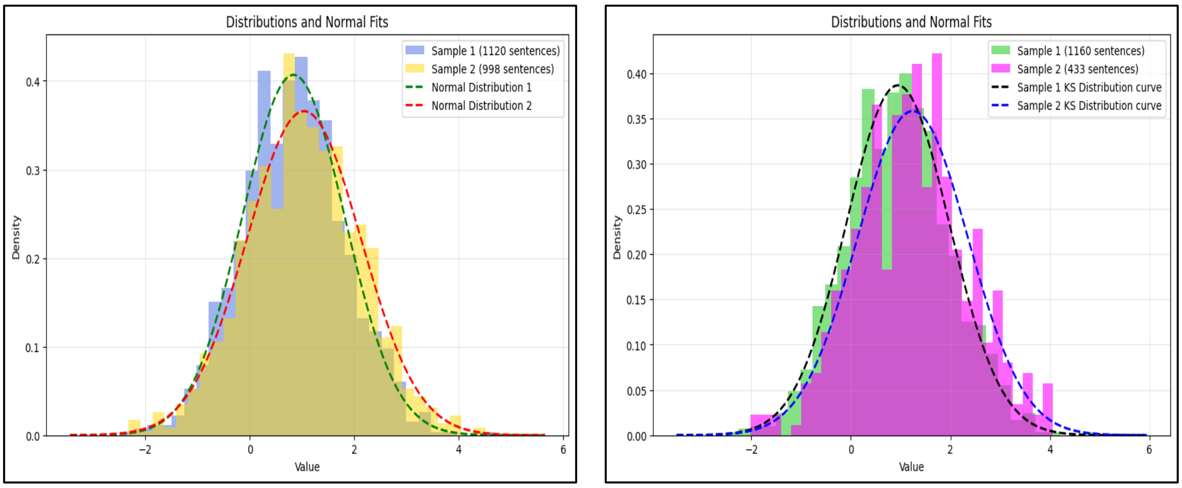
$$H_a: c_1 \neq c_2, \text{ i.e., the accuracy of sample 1} \neq \text{, accuracy of sample 2}$$

According to the Kolmogorov-Smirnov test for two samples, the condition to be satisfied, that D-value (D_{N_1,N_2}) (called KS- statistics) must be less than D_α, i.e., $D_{N_1,N_2} < D_\alpha$. Here, α is the significance level. If it follows this condition, the two samples accept the null hypothesis. In the case of the proposed method, if it follows the condition of the KS test, then the given null hypothesis is accepted or rejected. We compute the given KS-statistics D_{N_1,N_2}, as shown in *Eqn. (3.17).*

$$D_{N_1,N_2} = \max\left(|c_1 - c_2|\right) \tag{3.17}$$

$$D_\alpha = \frac{D(\alpha)\times\left(\sqrt{\frac{1}{N_1}+\frac{1}{N_2}}\right)\times D_{N_1,N_2}\times(1-D_{N_1,N_2})}{10} \tag{3.18}$$

Thus, computing the given values into *Eqn. (3.18)* D_α showed that the given null hypothesis had been accepted without failure.



(a) (b)

Fig. 3.10 Statistical validation of (a). SOBH-REB and (b). SDO-BCRT using the KS Test

Figs. 3.10 (a) and *(b)* depict the given D-value calculated as 0.8845, greater than the significance level $\alpha = 0.05$ where the two curve's peaks meet. This implied that the null hypothesis has been accepted, signifying that the given two samples belong to the same distribution and have a constant F1 score for any sample set.

3.7.1.3 Statistical Validation using Mann-Whitney U Test

The Mann-Whitney U-test determines whether or not two distinct samples originating from different populations have the same distribution. Based on the provided sample data, we constructed a distribution by employing the mean and standard deviation (shown in *Eqn. (3.19)- (3.22))* coming from the number of appropriately categorized sentences in each of the samples shown in *Table 3.28.*

For the two population proportions (p_1 and p_2), null hypotheses (H_0) and alternative hypotheses (H_a) are defined as follows:

H_0: $p_1 = p_2$, i.e., Accuracy distribution of sample 1 = Accuracy distribution of sample 2.

H_a: $p_1 \neq p_2$, i.e., Accuracy distribution of sample 1 ≠ Accuracy distribution of sample 2.

$$U = \min(U_1, U_2) \tag{3.19}$$

$$\mu_{m_1,m_2} = \left(\frac{U_1 \times U_2}{2}\right) \tag{3.20}$$

$$\sigma_{m_1,m_2} = \sqrt{\frac{U_1 \times U_2 \times (U_1 + U_2 + 1)}{12}} \tag{3.21}$$

$$z = \frac{U - \mu_{N_1,N_2}}{\sigma_{N_1,N_2}} \tag{3.22}$$

Table 3.28 Computation of Mann-Whitney U test results for SDO-BCRT and SD-GKOT

	SDO-BCRT	
Parameters	*Sample 1*	*Sample 2*
Sample size (N)	*1660*	*513*
Sample proportion (ECDF)	*0.8686*	*0.8674*
Favorable cases (X)	*1442*	*445*
U- value	*0.8686*	
$\boldsymbol{mean_{N_1,N_2}}$	*0.3767*	
$\boldsymbol{Std_{N_1,N_2}}$	*0.4144*	
$\boldsymbol{z_B}$	*0.8701*	
Hypothesis status	H_0 *is not rejected*	

Fig. 3.11 shows the critical region of the hypothesis z_B value as depicted graphically. This finding suggested that the consistency of the proposed method's accuracy score holds across diverse sample sizes from various datasets. Therefore, the null hypothesis has been accepted, signifying that the two samples belong to the same distribution. This concludes that it has a constant accuracy score for any sample set.

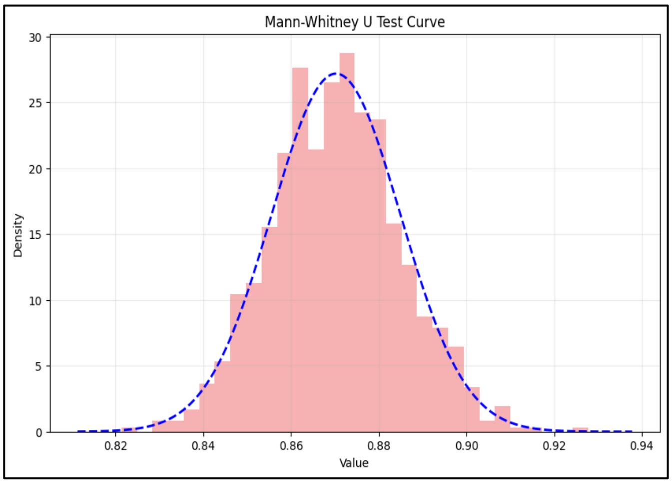


Fig. 3.11 Statistical validation of SDO-BCRT using the Mann-Whitney U Test

3.7.2 Ablation Study of SOBH-REB and SDO-BCRT

A set of ablation tests has been carried out to see how the value of τ_1 affected the performance of the proposed method. The study has proceeded by taking values $\tau_1 > 2$ and $\tau_1 < 2$ respectively. *Table 3.29* shows the use of the actual threshold value $\tau_1 = 2$ which performed better than the other values.

Table 3.29 Ablation study of the proposed method SOBH-REB

Dataset	***Removed module***	***Accuracy***	***Precision***	***Recall***	***F1- score***
Dataset-1 [149]	$\boldsymbol{\tau_1} < 2$	*0.8860*	*0.9484*	*0.8127*	*0.8724*
	$\boldsymbol{\tau_1} > 2$	*0.8675*	*0.8055*	*0.8733*	*0.8380*
	SOBH-REB with $\boldsymbol{\tau_1 = 2}$	***0.9541***	***0.9555***	***0.9516***	***0.9536***
	$\tau_2 < 0.78$	*0.8866*	*0.9572*	*0.7993*	*0.8711*
	$\tau_2 > 0.78$	*0.7845*	*0.8078*	*0.9574*	*0.8762*
	SDO-BCRT with $\tau_2 = \mathbf{0.78}$	***0.9497***	***0.9261***	***0.9722***	***0.9486***
Dataset-2 [150]	$\boldsymbol{\tau_1} < 2$	*0.9079*	*0.9533*	*0.8539*	*0.9008*
	$\boldsymbol{\tau_1} > 2$	*0.8050*	*0.7385*	*0.8890*	*0.8067*
	SOBH-REB with $\boldsymbol{\tau_1 = 2}$	***0.9505***	***0.9538***	***0.9444***	***0.9491***
	$\tau_2 < 0.78$	*0.8760*	*0.9407*	*0.7525*	*0.8361*
	$\tau_2 > 0.78$	*0.7626*	*0.7842*	*0.9430*	*0.8562*
	SDO-BCRT with $\tau_2 = \mathbf{0.78}$	***0.9494***	***0.9502***	***0.9625***	***0.9563***
Dataset-3 [151]	$\boldsymbol{\tau_1} < 2$	*0.9231*	*0.9367*	*0.9000*	*0.9172*
	$\boldsymbol{\tau_1} > 2$	*0.8276*	*0.7532*	*0.8682*	*0.8066*
	SOBH-REB with $\boldsymbol{\tau_1 = 2}$	***0.9543***	***0.9403***	***0.9657***	***0.9529***
	$\tau_2 < 0.78$	*0.7027*	*0.8839*	*0.7271*	*0.7978*
	$\tau_2 > 0.78$	*0.8793*	*0.7191*	*0.9029*	*0.8005*
	SDO-BCRT with $\tau_2 = \mathbf{0.78}$	***0.9488***	***0.9476***	***0.9533***	***0.9505***

The performance of the proposed methods (SOBH-REB and SDO-BCRT) varied significantly with changes in the thresholds τ_1 and τ_2. When τ_1 is set to less than 2, the SOBH-REB showed lower accuracy and recall due to an increased number of false-negative sentences that are actually sarcastic but classified as non-sarcastic. In contrast, setting τ_1 greater than 2 led to lower precision and accuracy across all datasets due to a rise in false positives, where non-sarcastic sentences are misclassified as sarcastic. The optimal value $\tau_1 = 2$, resulting in fewer misclassifications, enhancing the SOBH-REB model's ability to comprehend sarcasm effectively. Similarly, when the sarcasm threshold τ_2 in SDO-BCRT is below 0.78, accuracy and recall decrease due to more false negatives. When τ_2 exceeded 0.78, the method SDO-BCRT suffered lower precision and accuracy because of increased false positives. Setting τ_2 at 0.78 proved ideal, as it improved semantic interpretation and reduced classification errors, thereby enhancing the overall performance of the sarcasm detection method in SDO-BCRT.

3.7.2 Computation Costs of Five Proposed Methods

The time complexity (T) measures the number of computational steps needed to accomplish a given task on a specific input dataset.

Table 3.30 Computational time complexities of the five proposed methods

S. no.	*Proposed methods*	*Time complexity*
1.	*SOBH-REB*	$O(n)$
2.	*SDO-BCRT*	$O(mn)$
3.	*SD-SCOT*	$O\big(n \times (m+n)\big)$
4.	*LDL-SDTE*	$O(n)$
5.	*SWD-SEO*	$O(n^2)$

We evaluate the time complexity ($T(n, m)$) of the algorithms under different scenarios, where n denotes the number of alternative operations and m represents the number of sentences. The values of $T(n, m)$ indicate the performance of the three defined algorithms in *Table 3.30*.

3.7.3 Challenges of the Proposed Methods

There are some challenges undertaken by the five proposed methods based on optimization techniques using different probabilistic distributions. The first is the lack of ethnic uniqueness. Due to the lack of such contextual and background knowledge, the proposed models need some improvement. The second is a lack of veiled over polite sarcasm. Sometimes, being respectful is a pretext for sarcasm and highly complex language unsuitable for casual conversation.

Giving someone a highly polite compliment is a common indication of sarcasm, which the proposed method did not capture. *Table 3.31* shows the text examples incorrectly classified by the proposed methods. The text samples are given in a table with actual labels for every dataset, even though proposed algorithms estimated the opposite.

Table 3.31 Misclassified sentences utilizing the proposed methods

S. no.	*Sentences*	*Misclassified by five proposed methods*	*Actual label*
1.	*"Do you think I should attempt to stop my ex-lover's promotion?"*	*Non-sarcasm*	*Sarcasm*
2.	*"I just want her gone in my dreams, giving love hatred screams of desires."*	*Non-sarcasm*	*Sarcasm*

3.8 Summary

This chapter presents five new methods: SOBH-REB, SDO-BCRT, SD-SCOT, LDL-SDTE, and SWD-SEO for sarcasm detection in social media using various optimization techniques. Each method uses techniques, such as optimization based on the Bessel function, Box-Cox transformation, and distributional techniques, to identify sarcasm by analysing feature relationships and sentiments. These proposed methods effectively capture complex patterns in language, making sarcasm detection more robust and reliable. The proposed methods outperform existing methods in terms of accuracy, precision, recall, and F1 score, as confirmed by statistical tests across benchmark datasets. Their main goal is to help the public recognize sarcastic content online, thus promoting awareness in social media.

This chapter highlights that leveraging unsupervised optimizations significantly improve sarcasm detection performance effectively. The proposed methods also demonstrate strong and consistent results, showing their reliability and adaptability to different data sources with ablation results. In conclusion, the chapter demonstrates the advanced optimization and distributional approaches for sarcasm detection further strengthen the field of NLP.

Chapter 4
Hybrid Optimization-Based Methods for Sarcasm Detection in English

This chapter focuses on techniques for sarcasm detection that involve optimizing the continuous statistical distributions, combining them with varied NLP methodologies. In this chapter, we present novel approaches for sarcasm detection in English, leveraging hybrid optimization techniques to enhance computational efficiency.

The chapter is structured as follows: *Section 4.1* begins with a detailed background and introduction to sarcasm, and exploring its linguistic complexities. *Sections 4.2–4.5* focuses on various proposed methodologies employed to achieve improved detection accuracy. *Section 4.6* provides description of the datasets, followed by the performance analysis and statistical validations of the proposed methods. Moving forward, *section 4.7* highlights the challenges faced by proposed methods, with measure of their time complexities and computational efficiencies. Finally, *section 4.8* concludes the chapter by summarizing the findings and outlining the contributions of the proposed methodologies.

4.1 Introduction

Sarcasm detection has been used to a great extent in today's time. It is defined as use of words which make the actual literal meaning in an opposite intent. There are varied methods for sarcasm detection, including machine learning, deep learning, and transformer-based models. However, some of these methods still lack or neglect detecting sarcastic cues with proper intent without using optimization techniques.

Certain studies [31], [156] defined the sarcasm detection model but, unfortunately, lagged to give better performance metrics with high computational costs. The problem with the prior studies [61], [66], [67] are that most of them have a dearth of context-dependent based techniques. Multiple feature ensemble, behavioral, supervised, semi-supervised, deep learning, and multimodal models do not correctly incorporate the modalities of all features of sarcasm. Sarcasm detection often involves subtle contextual and tonal cues, which can be difficult to model using deterministic methods [86], [3]. Despite these methods [31], [11], [159] there are very few unsupervised techniques that utilize mathematical algorithms to detect sarcasm in context. Many studies [157], [160], [87] not able to detect sarcasm with the presence of positive sentiment showing implied negative intent that is commonly shared by users of social sites. All these issues need to be addressed by formalizing the problem of text classification and introducing novel mathematically optimized models for sarcasm detection.

To address these challenges, this chapter provides varied hybrid optimization models as novel approaches to sarcasm detection in English text. Hybrid optimization techniques are ideal for sarcasm detection due to their ability to navigate complex, high-dimensional data spaces efficiently. These techniques help to identify optimal solutions by handling uncertainties and variabilities in linguistic patterns. They are computationally efficient, capable of processing large datasets while minimizing overfitting, and adaptable to diverse sarcasm expressions. Sarcasm detection models achieve better precision and robustness by leveraging stochastic methods, enhancing their overall performance and applicability.

The use of different hybrid optimization techniques in this chapter is motivated by the complexity of sarcasm in social media text. Sarcasm is nonlinear, context-dependent, and often difficult to capture with a single method. Each hybrid technique improves a specific part of the model, such as stability during convergence, diversity in the search space, or stronger interaction among features.

Combining these mathematical frameworks, the proposed hybrid methods identify subtle and hidden sarcasm patterns without using labelled data. This also improves the model's generalization and makes the unsupervised sarcasm detection process easier to interpret.

Formulating sarcasm detection as mathematical optimization techniques enables to give more feasible solutions leading to the robustness of proposed methods. Despite using different distribution functions, it helps to capture various aspects of sarcasm with its effective indicators which is shown in detailed as in *Chapter 5*. The use of maximized or minimized distribution functions for sarcasm detection offers unique insights into the problem. Integrating optimization techniques helps refine sentiment scores, leading to more accurate sarcasm detection with ideal optimal value. This chapter discusses four proposed approaches for sarcasm detection of English text using varied hybrid optimization techniques.

- **Proposed method 6:** Sarcasm detection using logistic ions motion optimization technique with entropy correlations network (SLIOEN), which is discussed in *section 4.2*.

- **Proposed method 7:** Sarcasm detection method using unified non-linear Gauss-Kuzmin optimization with the Tsallis mechanism (SD-GKOT), which is discussed in *section 4.3*.
- **Proposed method 8:** Unified Chinese Remainder Theorem cooperative game Kullback optimization (UCRT-CGKO), which is discussed in *section 4.4*.
- **Proposed method 9:** Sarcasm detection using Wilson's theorem utilizing ensemble language models (SDWT-ELM), which is discussed in *section 4.5*.

4.2 Proposed Method 6: Sarcasm Detection Based on Logistic Ions Motion Optimization

In this section, we discuss the proposed methodology of sarcasm detection using logistic ions motion optimization - SLIOEN. *Fig. 4.1* demonstrates the flowchart of the proposed SLIOEN method using logistic ions motion optimization for sarcasm detection.

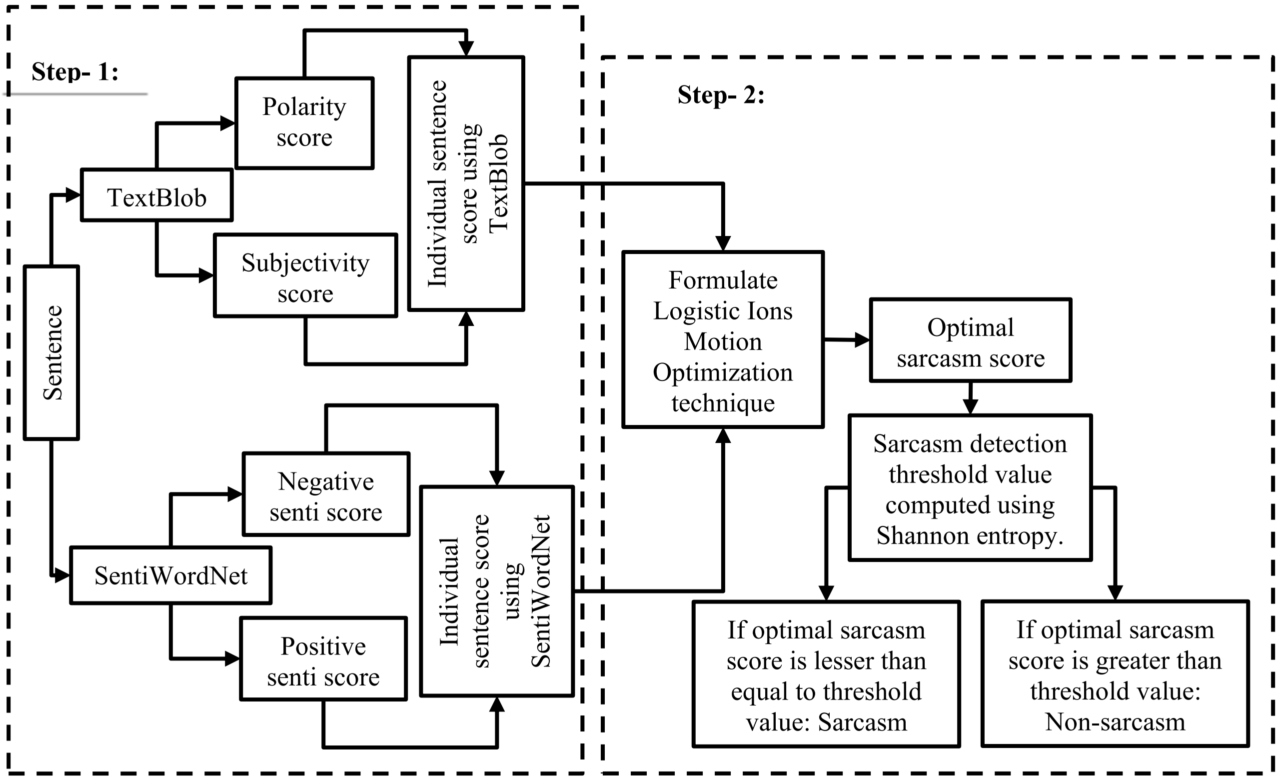


Fig. 4.1 Architectural framework of proposed method SLIOEN for sarcasm detection

The detailed core three steps used to implement the proposed SLIOEN method are shown as follows.

Step 1: Sentiment Evaluation and Sentence Score Calculation

The first step is to calculate two individual sentence scores using lexicon sentiment models viz, TextBlob and SentiWordNet. Their computed scores are T_{pol}, T_{sub}, S_{pos}, and S_{neg} using *algorithms 3.2* and *3.3* in *Chapter 3*. The individual sentence score obtained from TextBlob (ISS_2) is the average of T_{pol} and T_{sub}, as shown in *Eqn. (4.1).*Similarly, the individual sentence score obtained from SentiWordNet (ISS_3) is the average of S_{pos} and S_{neg}, shown in *Eqn. (4.2)*.

$$ISS_2 = \frac{(T_{sub}+T_{pol})}{2} \tag{4.1}$$

$$ISS_3 = \frac{(S_{pos}+S_{neg})}{2} \tag{4.2}$$

The goal of averaging is to create a balanced, consistent, and comparable sentence representation, mitigating bias from skewed positive or negative word distributions.

Step 2: Key Phrase Extraction using RAKE

After getting optimal sarcasm scores, we apply the key-phrase extraction procedure using the RAKE model, which is explained in *Chapter 3*. It calculates the triggered context score of the sarcastic sentence called the highest triggered sarcasm indicator score *(k^3_{max})*.

Step 3: Sarcasm Detection

In the third step, we formulate a logistic distribution as a minimized objective function solved by ion motion optimization for sarcasm detection. The ISS_2 and ISS_3 are the coefficients of the function, which are directly calculated from step 1. Using logistic distribution as an objective function helps model systems with binary outcomes, growth saturation, and robustness optimal solutions. Using this objective function efficiently helps to adapt new patterns and data, capturing more complex sarcastic cues. This enhances exploration capability, providing better optimal conditions and faster convergence. The *Eqn. (4.3)* shows the non-linear optimization technique using the logistic distribution function as the absolute minimized objective function.

$$\psi_6 = f(\psi_6^*) = \left| \text{Minimise} \left(\frac{ISS_2}{ISS_3} \times \frac{e^{-\psi_6^*}}{\left(1+e^{-\psi_6^*}\right)^2} \right) \right| \times 100 \tag{4.3}$$

The aim of the minimized non-linear logistic distribution function using the ions motion optimization technique is to get the optimal sarcasm score (ψ_6). The ψ_6^* is the sentence's optimal solution, which helps in to get the final optimal sarcasm score of sentences to detect sarcasm. The value of ψ_6^* in *Eqn.(4.3)* is obtained using the ion motion optimization technique. The inspiration for the IMO algorithm came from the motion law of anions and cations in nature [161], where the anions and cations act as the candidate solutions in solution space. They move continuously toward the optimal ions. The primary steps of IMO are as follows. The proposed method has taken the search space within the bounds, i.e., $\psi_6^* \epsilon [l_i = -10, u_i = 10]$ with maximum iterations (T_{MAX}) taken as hundred. The ions work in two different phases. The first step is the initialization phase, where the population is initialized using *Eqn. (4.4)*.

$$\psi^*_{6_i} = rand \times (u_i - l_i) + l_i \tag{4.4}$$

In the initialization phase, the population is evenly split into two groups: anions and cations, where $\psi^*_{6_i}$ denotes the value for the ith ion. While $rand$ is a random number with uniform distribution within the range of 0 to 1, l_i is the lower bound of the ith variable, and u_i is the upper bound. The second phase is the liquid phase, where the resultant attraction force of anions and cations is calculated with the help of the distance of anion and cation, as shown in *Eqn. (4.5)* and *(4.6)*.

$$\zeta F_i = \frac{1}{\left(1+e^{-\frac{0.1}{\zeta D_i}}\right)} \tag{4.5}$$

$$\mu F_i = \frac{1}{\left(1+e^{-\frac{0.1}{\mu D_i}}\right)} \tag{4.6}$$

Where $\zeta D_i = |\zeta_i - \mu best|$ and $\mu D_i = |\mu_i - \zeta best|$ such that ζD_i is the distance of the ith anion from the best cation, whereas μD_i calculates the distance of the ith cation from the best anion. At the same time, ζF_i and μF_i are the resultant attraction forces of anions and cations, respectively. This shows that forces between ions are inversely proportional to the distance. As a result, based on this resultant force of attractions, the position updates are derived in *Eqns. (4.7)* and *(4.8)*.

$$\zeta F_i = \zeta_i + \zeta F_i \times (\mu best - \zeta_i) \tag{4.7}$$

$$\mu F_i = \mu_i + \mu F_i \times (\zeta best - \mu_i) \tag{4.8}$$

The $\mu best$ and $\zeta best$ give the best cation and anion, respectively. The third phase is the crystal phase, where ions in the liquid become progressively stabilized by gravitational forces and solidify from liquid into a crystal. Under specific conditions, they switch to the crystal phase search mode to escape local optima. Consequently, the positions are updated using *Eqn. (4.9)* and *(4.10)*.

$$\left.\begin{array}{ll} \zeta_i = \zeta_i + rand_1 \times (\mu best - 1) & rand_1 > 0.5 \\ \zeta_i = \zeta_i + rand_1 \times (\mu best) & rand_1 \leq 0.5 \\ \zeta_i = rand_1 \times (u_i - l_i) + l_i & rand_1 < 0.05 \end{array}\right\} \quad (4.9)$$

$$\left.\begin{array}{ll} \mu_i = \mu_i + rand_2 \times (\zeta best - 1) & rand_2 > 0.5 \\ \mu_i = \mu_i + rand_2 \times (\zeta best) & rand_2 \leq 0.5 \\ \mu_i = rand_2 \times (u_i - l_i) + l_i & rand_2 < 0.05 \end{array}\right\} \quad (4.10)$$

The $rand_1$ and $rand_2$ are random numbers between $[-1,1]$. Using the position updating *Eqns. (4.9)* and *(4.10)*, we compare the fitness of anions and cations to obtain the current optimal solution ψ_6^*. Therefore, this optimization stage between the liquid and crystal phases is repeated in the final step. In the IMO, ions alternate movement between the liquid and crystal phases goes on until the termination condition is met, while updating into the fitness function. The optimal ion's output gives the desired optimal solution ψ_6^*. For detailed procedural steps of the IMO algorithm, refer to [161]. The ions with the lowest score, representing the best fitness, is considered the optimal sarcasm score (ψ_6). The ψ_6 is then compared with the computed sarcasm detection threshold value (τ_6). It is calculated as 0.5 using the Shannon entropy [162] structure shown in *Table 4.1,* shown in *Eqn. (4.11)*. The $p(\psi_6)$ is the probability distribution of ψ_6, i.e., optimal sarcasm score.

$$\tau_6 = -\sum_{i=1}^{i=f} p(\psi_{6_i}) \log p(\psi_{6_i}) \quad (4.11)$$

Table 4.1 Parameters employed in the proposed method SLIOEN

Parameters	***Values***
Sarcasm detection threshold value ($\boldsymbol{\tau_6}$)	*(Dataset-1):* $\tau_6 = -\sum_{i=1}^{i=f} p(\psi_{6_i})\, log\, p(\psi_{6_i}) = 0.4730 \approx 0.5$ *(Dataset-2):* $\tau_6 = -\sum_{i=1}^{i=f} p(\psi_{6_i})\, log\, p(\psi_{6_i}) = 0.4411 \approx 0.5$ *(Dataset-3):* $\tau_6 = -\sum_{i=1}^{i=f} p(\psi_{6_i})\, log\, p(\psi_{6_i}) = 0.4289 \approx 0.5$

The sarcasm detection is deduced if $\psi_6 \leq \tau_6$, then the sentence is tagged as "*sarcastic*," else, it is tagged as "*non-sarcastic*." *Algorithm 4.1* defines the procedure of sarcasm detection based on the proposed method SLIOEN.

Furthermore, a non-linear Poisson distribution optimization technique is applied for categorizing types of sarcasm in SLIOEN as a step 4 procedure detailed in *Chapter 5*.

ALGORITHM 4.1: Sarcasm detection using the proposed method- SLIOEN
Input: $X_i \in$ *sentences where* $1 \leq i \leq m$ *(total number of sentences),* ISS_2*,* ISS_3*,* τ_6*.* *Output: Detect sarcasm or non-sarcasm based on optimal sarcasm score (*ψ_6*)* 1. *For each sentence X from the dataset, repeat steps 2 to 11.* 2. *Define objective function:* $\psi_6 = f(\psi_6^*) =$ *Minimise* $\left(\frac{ISS_1}{ISS_2} \times \frac{e^{-\psi_6^*}}{\left(1+e^{-\psi_6^*}\right)^2}\right) \times 100$ 3. *Compute* ψ_6^* *using ions motion optimization technique: Initialize ions initial position within the bounds* $[l, u]$*:* $\psi_6^* = rand \times (u - l) + l$ 4. *While* $T < T_{MAX} = 100$*: Calculate fitness of the objective function; according to fitness, determine the best global solution.* 5. *Update the ions position based on liquid phase:* $\zeta F = \zeta + \zeta F \times (\mu best - \zeta)$ *and* $\mu F = \mu + \mu F \times (\zeta best - \mu)$*.* 6. *Update ions position based on crystal phase:* $\begin{cases} \zeta = \zeta + rand_1 \times (\mu best - 1) & rand_1 > 0.5 \\ \zeta = \zeta + rand_1 \times (\mu best) & rand_1 \leq 0.5 \text{ and} \\ \zeta = rand_1 \times (u - l) + l & rand_1 < 0.05 \end{cases}$ $\begin{cases} \mu = \mu + rand_2 \times (\zeta best - 1) & rand_2 > 0.5 \\ \mu = \mu + rand_2 \times (\zeta best) & rand_2 \leq 0.5 \\ \mu = rand_2 \times (u - l) + l & rand_2 < 0.05 \end{cases}$ 7. *Evaluate fitness:* $f\left(\psi_6^*(T+1)\right) =$ *objective function at* $\psi_6^*(T+1)$*. Stop the condition where the minimized value of* $f\left(\psi_6^*(T+1)\right)$ *is obtained in total iterations.* 8. *Output "optimal sarcasm score (*ψ_6*)."* 9. *Classification based on* ψ_6 *and* τ_6*: if* $\psi_6 \leq \tau_6$ *return "sarcasm"* *else return "non-sarcasm"* 10. *End*

4.2.1 Numerical Illustration of Proposed Method: SLIOEN

We have taken two sample sentences, i.e., S1 and S2, from our dataset to illustrate the implementation of the proposed SLIOEN method on it.

In step 1, we determine the sentence score using TextBlob (using *algorithm 3.3 in Chapter 3*) and SentiWordNet (using *algorithm 3.2* in *Chapter 3*). *Tables 4.2* and *4.3* show the interpreted values using *Eqn. (4.1)* and *(4.2)* respectively. In step 2, we defined the triggered sarcasm indicator score using the keyword extraction model RAKE. This sarcasm indicator score is used to learn the sarcastic intent of the text.

S1: "I'm always amazed at how child can ruin the toys and then pamper that toy with kindness Loll..amziinggg"   ***S2: "In my years in the State Legislature, I have never voted for a tax increase."***

Table 4.4 shows the computed score using step 2 from the proposed methodology section.

Table 4.2 Individual sentence score obtained from TextBlob

Sentences	*Textblob*		ISS_2
	T_{pos}	T_{neg}	
S1	*1*	*0.25*	*0.62*
S2	*0*	*-0.05*	*0.025*

Table 4.3 Individual sentence score obtained from SentiWordNet

Sentences	*SentiWordNet*		ISS_3
	S_{pos}	S_{neg}	
S1	*0.02272*	*0.01136*	*0.01704*
S2	*0.00961*	*0.1057*	*0.0576*

Table 4.4 Sarcasm indicator score computed using RAKE model

Sentences	k^3_{max}
S1	*9*
S2	*8*

Table 4.5 Sarcasm detection using the proposed SLIOEN method

Sentences	ISS_1	ISS_2	ψ_6	*Tag*
S1	*0.62*	*0.01704*	0.19602	*Sarcasm*
S2	*0.025*	*0.0576*	1.0833	*Non-sarcasm*

While in step 3, we detect sarcasm using Algorithm *4.1*. The proposed method employs minimised logistic distribution function using the ions motion optimization technique. *Table 4.5* shows the computed optimal sarcasm score obtained from SLIOEN using *algorithm 4.1*.

The decision of sarcasm detection is made based on the comparison between ψ_6 and τ_6. The $\psi_6 \leq \tau_6$, for the sentence S1, is tagged as "*sarcastic*," and the sentence S2 is tagged as "*non-sarcastic*". Furthermore, we deduce the inherent sarcasm kind in terms of showing the sarcastic comment either as humour or non-humour in nature, in *Chapter 5*.

4.3 Proposed Method 7: Sarcasm Detection Based on Exponential Gauss-Kuzmin Optimization

In this section we discussed in detail about the proposed methodology of sarcasm detection method SD-GKOT. *Fig. 4.2* demonstrates the flowchart of the proposed method SD-GKOT for sarcasm detection using optimization technique.

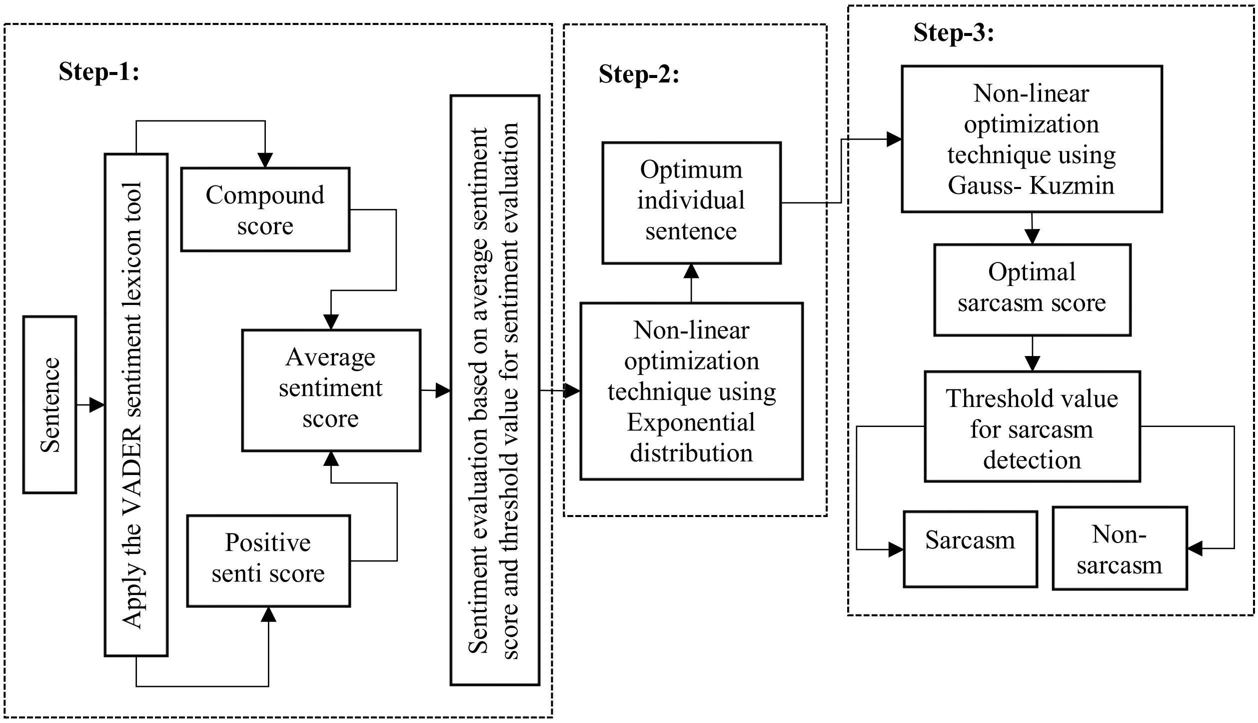


Fig. 4.2 Architectural framework of proposed method SD-GKOT for sarcasm detection

SD-GKOT involves three key steps in the process of sarcasm detection detailed as follows.

Step 1: Sentiment Evaluation and Sentence Score Calculation

In the first step, the lexicon sentiment model, i.e., VADER is used to compute the average sentiment score of a sentence (ϕ_4). The compound score (V_{comp}) and the positive senti score

(V_{pos}) are obtained from VADER using *algorithm 3.1* in *Chapter 3*. The computed average sentiment score of the sentence is as shown in *Eqn. (4.12)*.

$$\phi_4 = \frac{|V_{pos} + V_{comp}|}{2} \tag{4.12}$$

With the help of the ϕ_4 and sentiment threshold value (λ_4), we reduce the polarity of the text either as positive sentiment or negative sentiment. The λ_4 is computed using Kullback Liebler divergence shown in *Table 4.6* computed as 0.2. The general formula of computed λ_4 using Kullback Leibler divergence in *Eqn. (4.13)*.

$$\lambda_4 = \sum Y(i) \log\left(\frac{Y(i)}{\phi_4(i)}\right) \tag{4.13}$$

The λ_4 is the computed threshold value for sentiment evaluation, $\phi_4(i)$ and $Y(i)$ are the average sentiment score for i^{th} value and product value of positive and compound score of each value.

Table 4.6 Parameters used in the proposed method for sentiment evaluation

Parameters	***Values***
The threshold value for sentiment evaluation ($\boldsymbol{\lambda_4}$)	*(Dataset 1 [149]):* $\lambda_4 = \sum Y(i) \log\left(\frac{Y(i)}{\phi_4(i)}\right) \approx 0.2$ *(Dataset 2 [150]):* $\lambda_4 = \sum Y(i) \log\left(\frac{Y(i)}{\phi_4(i)}\right) \approx 0.2$ *(Dataset 3 [151]):* $\lambda_4 = \sum Y(i) \log\left(\frac{Y(i)}{\phi_4(i)}\right) \approx 0.2$ *(Dataset 4 [152]):* $\lambda_4 = \sum Y(i) \log\left(\frac{Y(i)}{\phi_4(i)}\right) \approx 0.2$

The process of sentiment evaluation is done if ϕ_4 is greater than λ_4, then the value is tagged as "*positive sentiment*" otherwise, it is tagged as "*negative sentiment.*"

Additionally, we formulate a non-linear optimization technique using exponential distribution to calculate the optimum individual sentence score (ISS_4). This use of exponential distribution as a minimized objective function is taken because it tends to be smooth and convex in nature for useful for optimization. This helps in achieving the optimal score efficiently, where smooth convex functions ensure better convergence properties. The generalized non-linear optimization technique using exponential distribution is shown in *Eqn. (4.14)*.

$$\left.\begin{array}{c} ISS_4 = Minimise\ \phi_4\ e^{-\phi_4} \\ Subject\ to, \phi_4 \geq 0 \end{array}\right\} \tag{4.14}$$

The aim of this minimized non-linear optimization technique using exponential distribution as an objective function is to acquire the ISS_4. *Algorithm 4.2* demonstrates the calculation of the given score based on a defined optimization technique.

Step 2: Key Phrase Extraction using RAKE

In this step, the key-phrase extraction RAKE model (discussed in *Chapter 3*) is performed on all sarcastic sentences to acquire the highest key-phrase value (k^4_{max}).

ALGORITHM 4.2: Calculate the optimum individual sentence score using a non-linear optimization technique with exponential distribution
Input: X− total number of sentences in datasets, $\boldsymbol{\phi_4}$ *Output: Optimum individual sentence score (*$\boldsymbol{ISS_4}$*)* 1. *For each sentence,* $\boldsymbol{X}$ *repeat steps from 2 to 3.* 2. *Compute* $\boldsymbol{ISS_4}$ *using the formulated non-linear optimization technique with exponential distribution:*$\boldsymbol{ISS_4}$ $=$ *Minimize* $\boldsymbol{\phi_4}\ \boldsymbol{e^{-\phi_4}}$*, Subject to, (*$\boldsymbol{\phi_4}$*)* $\geq \mathbf{0}$ 3. *End*

Step 3: Sarcasm Detection

In the third step, a non-linear optimization technique is formulated based on the Gauss-Kuzmin distribution for sarcasm detection. The ISS_4, which is derived in step 1, is inserted into this defined objective function. The choice of taking the Gauss-Kuzmin distribution as a minimized objective function helps to capture skewed distributions with better patterns. The formulated non-linear Gauss-Kuzmin optimization technique is shown in *Eqn. (4.15)*.

$$\left.\begin{array}{c}\psi_7 = Minimise\ \frac{\log_{10}\left(1+\frac{1}{ISS_4}\right)}{\exp(1+ISS_4)} \\ Subject\ to, ISS_4 \geq 0\end{array}\right\} \tag{4.15}$$

The goal of this optimization technique is to derive the optimal sarcasm score (ψ_7) of a sentence from the minimized objective function for sarcasm detection. The ψ_7 is then compared with the computed threshold value for sarcasm detection (τ_7). The calculated value of τ_7 is 0.4, obtained from Tsallis entropy [164], which is shown in *Eqn. (4.16)* and elaborated in *Table 4.7*.

Table 4.7 Parameters used in the proposed method

Parameters	***Values***
The threshold value for sarcasm detection $(\boldsymbol{\tau_7})$	*(Dataset 1 [149]):* $\tau_7 = \frac{0.068}{1-0.33} log(\lvert 8212.57\rvert) = 0.3972 \approx 0.4$ *(Dataset 2 [150]):* $\tau_7 = \frac{0.068}{1-0.33} log(\lvert 7362.111\rvert) = 0.3924 \approx 0.4$ *(Dataset 3 [151]):* $\tau_7 = \frac{0.068}{1-0.33} log(\lvert 6687.508\rvert) = 0.3882 \approx 0.4$

$$\tau_7 = j \times \frac{1}{1-q} \log\left(\left|1 - \sum_i \left(P(\psi_{7_i})\right)^q\right|\right) \tag{4.16}$$

The q is the normalized set taken as 0.33, $P(\psi_{7_i})$ is the probability of optimal sarcasm score, where $1 \leq i \leq m$, m is the total number of sentences in the dataset. j is the normalization constant taken as the variance of optimal sarcasm score as 0.068.

Based on a comparison between ψ_7 and τ_7, the decision for sarcasm detection is inculcated. If $\psi_7 < \tau_7$, then the sentence is tagged as "*sarcasm*," and if $\psi_7 \geq \tau_7$, then the sentence is tagged as "*non-sarcasm*." *Algorithm 4.3* defines the procedure for sarcasm detection based on the proposed method SD-GKOT.

ALGORITHM 4.3: Sarcasm detection using the proposed method SD-GKOT
Input: X− total number of sentences in the dataset, $\boldsymbol{ISS_4}$, $\boldsymbol{\tau_7}$ *Output: Detect sarcasm and non-sarcasm based on optimal sarcasm score (*$\boldsymbol{\psi_7}$*)* *1. For each sentence,* $\boldsymbol{X}$ *repeat steps from 2 to 4.* *2. Compute* $\boldsymbol{\psi_7}$ *using the formulated non-linear optimization technique of Gauss-Kuzmin distribution:* $\boldsymbol{\psi_7}$ *= Minimize* $\boldsymbol{log_{10}\left(1 + \frac{1}{ISS_4}\right)}$*, Subject to* $\boldsymbol{ISS_4 \geq 0}$ *3. Decision-based on comparison between* $\boldsymbol{\psi_7}$ *and* $\boldsymbol{\tau_9}$*: If* $\boldsymbol{\psi_7 < \tau_7} \rightarrow$ *"sarcasm"* *Else "non-sarcasm"* *End*

4.3.1 Numerical Illustration of Proposed Method: SD-GKOT

In this section, we present the working methodology of the proposed method, SD-GKOT, on two sentences, S1 and S2, taken from the datasets.

S1: "A study of where teachers apply to work found that "less than 5 percent wanted to work in rural Wisconsin." ***S2: "I can stare if I want to. I will stop if anyone glares at me..LoLL, this shows how I am unwanted romanticizing bug"***

In the first step, we deduce the sentiment and polarity of the given context using the *Eqn. (4.12)* shown in the above steps. *Table 4.8* shows the depiction of ϕ_4 and the obtained sentiment polarity from it. Based on a comparison between ϕ_4 and λ_4 we deduce that S1 is tagged as "*negative sentiment*" and S2 tagged as "*positive sentiment*".

Table 4.8 Sentiment evaluation and average sentiment score computation

Sentences	***Vader sentiment model***		$\boldsymbol{\phi_4}$	***Sentiment tag***
	$\boldsymbol{V_{pos}}$	$\boldsymbol{V_{comp}}$		
S1	*0*	*0.128*	*0.064*	*Negative*
S2	*0.266*	*0.4404*	*0.3532*	*Positive*

After getting the sentiment tag with ϕ_4 value, we then obtained the sentence score value using a minimized exponential distribution-based non-linear optimization technique. The optimum sentence score (ISS_4) is deduced using *algorithm 4.2*, depicted in *Table 4.9*.

Table 4.9 Optimum sentence score computation using a non-linear optimization technique

Sentences	$\boldsymbol{\phi_4}$	$\boldsymbol{ISS_4}$
S1	*0.064*	*0.060032*
S2	*0.3532*	*0.2481*

In the second step, we obtain the maximised k^4_{max} value using the RAKE model to get a triggered indicator of sarcasm shown in *Table 4.10*.

Table 4.10 Triggered sarcasm indicator highest score using RAKE

Sentences	***Triggered context***	$\boldsymbol{k^4_{max}}$
S1	*"5 percent wanted to work in rural"*	*5*
S2	*"LoLL, unwanted romanticizing bug"*	*7*

In the third step, we obtain the optimal sarcasm score (ψ_7) based on utilizing a non-linear optimization technique using a fusion of Gauss-Kuzmin distribution as shown in *algorithm 4.3*. *Table 4.11* shows the obtained value of ψ_7 using an *algorithm 4.3*.

Table 4.11 Optimal sarcasm score using SD-GKOT for sarcasm detection

Sentences	ϕ_4	ISS_4	ψ_7	*Tag*
S1	*0.064*	*0.060032*	*0.4311*	*Non-sarcasm*
S2	*0.3532*	*0.2481*	*0.2014*	*Sarcasm*

The sarcasm detection is done based on the decision between ψ_7 and the threshold value for sarcasm (τ_7). On the basis of the decision, we deduce that sentence S1 is tagged as "*non-sarcasm*," and S2 is tagged as "*sarcasm*," respectively.

4.4 Proposed Method 8: Sarcasm Detection based on Chinese Remainder Theorem and Cooperative Kullback-Divergence Optimization

In this section, we discussed in detail the UCRT-CGKO method for the sarcasm detection task. This work aims to decipher a sarcastic and non-sarcastic sentence from social media platforms using optimization technique. We propose a sarcasm detection model named UCRT-CGKO using an optimization technique. In this model, we first apply the Chinese Remainder Theorem (CRT) to the word embedding scores to obtain four basic feasible solutions under different modulo congruences. Next, we use these four values in a zero-sum game strategy, applying the maximin-minimax approach to determine the game's optimal score. Based on this optimal value, the model performs the task of sarcasm detection effectively. The optimized CRT system has been formulated as shown in *Eqn. (4.17)*:

$$\text{Max } (g(\psi_k)_i, f(\psi_k)_i)$$

Subject to constraints,

$$\left.\begin{aligned} X_i &\equiv ISS_{4_i}(mod\ \mu_4) \\ X_i &\equiv ISS_{5_i}(mod\ \mu_1') \\ X_i &\equiv ISS_{4_i}(mod\ \mu_2) \\ X_i &\equiv ISS_{5_i}(mod\ \mu_4') \\ X_i &\equiv ISS_{4_i}(mod\ \mu_2) \\ X_i &\equiv ISS_{5_i}(mod\ \mu_3') \end{aligned}\right\} \tag{4.17}$$

where $1 \leq k \leq 4$ and $1 \leq i \leq m$ for total m sentences. The four values obtained through the CRT methodology are *Penultimate Quartette Rate (PQR)* values, representing the system's optimal solutions for the variable X_i. These values are subsequently utilized in the MaxiMin-MiniMax game to create a 2×2 square matrix. *Fig. 4.3* shows the flowchart for implementing the proposed method.

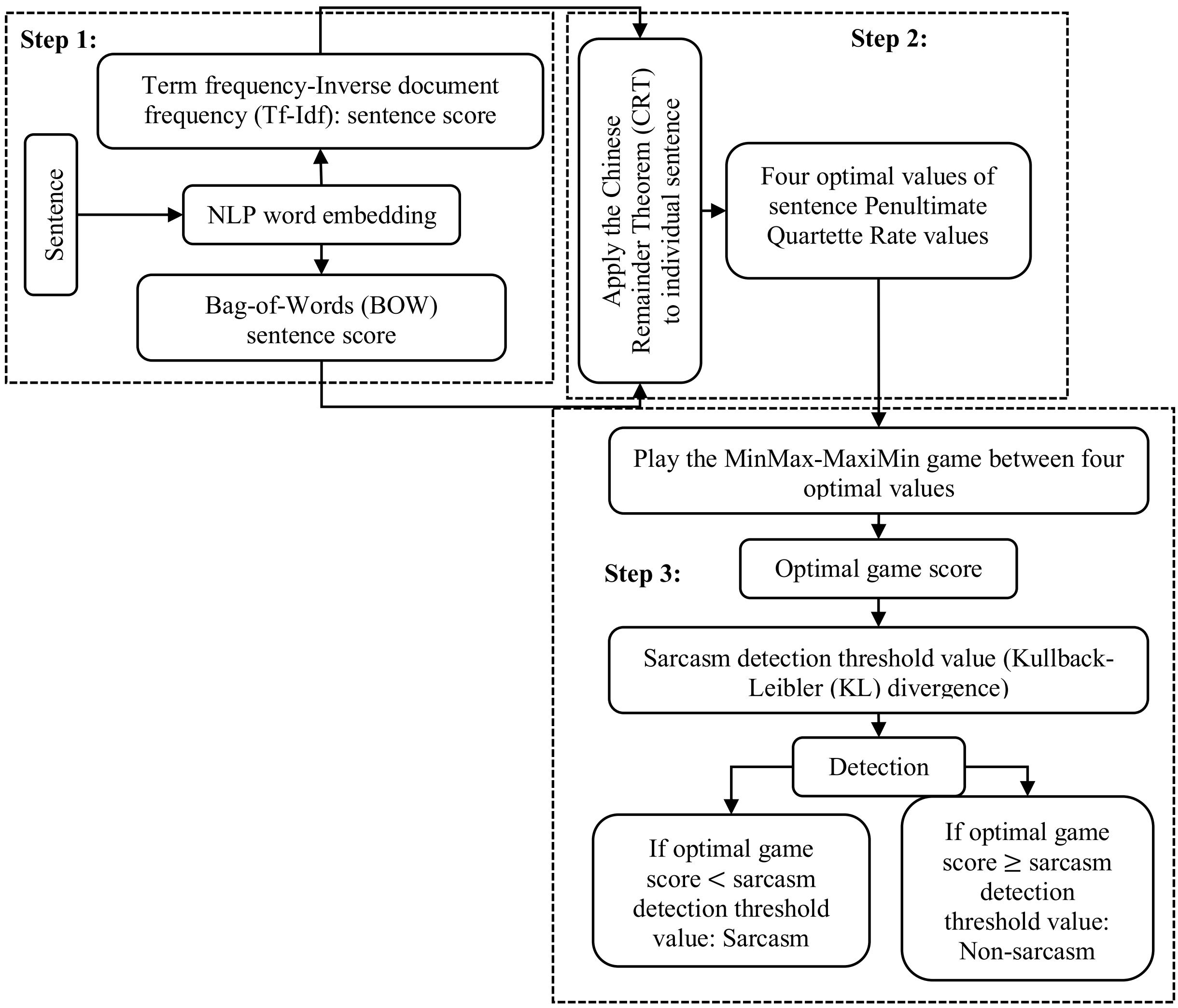


Fig. 4.3 Flowchart depicting steps to implement the proposed UCRT-CGKO method for sarcasm detection

Underneath follows the procedure of sarcasm detection in the text of the proposed methodology:

Step 1: Sentiment Evaluation and Sentence Score Calculation

In this step, we determine individual sentence scores ($\mathrm{ISS}_5, \mathrm{ISS}_6$) for each sentence using Bag of Words (BOW) and term frequency in reverse document frequency (Tf-Idf) word embedding.

We consider a sentence 'X' having $\mathrm{W} = (\mathrm{w}_1, \ . \ . \ . \ , \mathrm{w}_\mathrm{j})$ words. We generate two values - the sum of BOW (e^{BOW}) and Tf-Idf (e^{TIF}). We compute ISS_5 and ISS_6 according to *algorithm 4.4* and *algorithm 4.5*. *Eqn. (4.18)* depicts the representation:

$ISS_5 = \sum_{i=1}^{i=j} e^{BOW}{}_{w_i}$ (Let's say) total summation value from BOW

$ISS_6 = \sum_{i=1}^{i=j} e^{TFD}{}_{w_i}$ (Let's say) total summation value from Tf-Idf (4.18)

ALGORITHM 4.4: Calculate the Sentence Score using BOW
Input: $\boldsymbol{X_i}$ *is the* $\boldsymbol{i^{th}}$ *a sentence from the dataset where i ranges from* $\mathbf{1} \le \boldsymbol{i} \le \boldsymbol{m}$.   *Output: Individual Sentence Score using BOW (*$\boldsymbol{ISS_5}$*)*   1. *Repeat steps 2 to 5 for each sentence in the dataset.*   2. *Take* $\boldsymbol{W} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}, \ldots, \boldsymbol{w_{j,}})$ *where* $\boldsymbol{w_j}$ *represents the* $\boldsymbol{j^{th}}$ $(\mathbf{1} \le \boldsymbol{j} \le \boldsymbol{n})$ *word in* $\boldsymbol{X_i}$ 3. *For each unique* $\boldsymbol{w_j}$*, count its occurrences* $(\boldsymbol{f_j})$ *in the sentence* $\boldsymbol{X_i}$.   4. *Initialize* $\boldsymbol{e^{BOW}} = 0$ *for each* $\boldsymbol{w_j}$*,* $\boldsymbol{e^{BOW}} = \boldsymbol{e^{BOW}} + \boldsymbol{f_j}$ 5. *Compute:* $\sum_{i=1}^{i=j} \boldsymbol{e^{BOW}} \rightarrow \boldsymbol{ISS_5}$

ALGORITHM 4.5: Calculate the Sentence Score using Tf-Idf
Input: $\boldsymbol{X_i}$ *is the* $\boldsymbol{i^{th}}$ *a sentence from the dataset where i ranges from* $\mathbf{1} \le \boldsymbol{i} \le \boldsymbol{m}$.   *Output: Individual Sentence Score using Tf-Idf (*$\boldsymbol{ISS_6}$*)*   1. *Repeat steps 2 to 5 for each sentence in the dataset.*   2. *Take* $\boldsymbol{W} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}, \ldots, \boldsymbol{w_{j,}})$ *where* $\boldsymbol{w_j}$ *represents the* $\boldsymbol{j^{th}}$ $(\mathbf{1} \le \boldsymbol{j} \le \boldsymbol{n})$ *word in* $\boldsymbol{X_i}$ 3. *For each unique* $\boldsymbol{w_j}$*, Count the no. of occurrences* $(\boldsymbol{l_j})$ *in the sentence* $\boldsymbol{X_i}$.   4. *Initialize* $\boldsymbol{e^{TIF}} = 0$ *for each* $\boldsymbol{w_j}$*,* $\boldsymbol{e^{TIF}} = \boldsymbol{e^{TIF}} + \boldsymbol{l_j}$ 5. *Compute:* $\sum_{i=1}^{i=j} \boldsymbol{e^{TFD}} \rightarrow \boldsymbol{ISS_6}$

Step 2: Key Phrase Extraction using YAKE

While detecting sarcasm in text, we also seek terms associated with vulgar language or misleading comments. We extract keywords from the text using YAKE model. The keywords with the largest score are considered as triggered sarcasm markers, which helps to get offensive content. The highest keyword value is fetched out to further interpolate it in the formulated optimization technique for the classification of sarcasm signals shown in *Chapter 5*. The highest keyword value is termed the highest triggered sarcasm signal score (k^5_{max}).

Step 3: Sarcasm Detection

In this step, we express each sentence, *X,* in terms of a system of congruence as in *Eqn. (4.1)* using the sentence scores as the positive integers. CRT is applied to calculate the modulo of different factors of a large number independently and then combine the results to obtain the final solution. This improves parallelism and reduces computational load. We chose four prime numbers viz. $\mu_1 = 3, \mu_2 = 5, \mu_3 = 7 \; and \; \mu_4 = 11$. With four combinations of primes, i.e., $P_\emptyset = ((\mu_4, \mu_1), (\mu_2, \mu_4), (\mu_2, \mu_3), (\mu_3, \mu_4))$, we obtain unique four solutions termed as *PQR* (ω_k) shown in *Eqn. (4.19)* and *(4.20)*.

$$\begin{pmatrix} X \equiv ISS_5(mod\ \mu_k) \\ X \equiv ISS_6(mod\ \mu'_k) \end{pmatrix} \tag{4.19}$$

$$X = \left((ISS_5 \times \mu_k \times {\mu_k}^{-1}) + \left(ISS_6 \times \mu'_k \times {\mu'_k}^{-1}\right)\right) mod(\mu_k \times \mu'_k) = (\omega_k) \tag{4.20}$$

The $(\mu_k, \mu'_k) \in P_\emptyset = ((\mu_4, \mu_1), (\mu_2, \mu_4), (\mu_2, \mu_3), (\mu_3, \mu_4))$ are co-primes and $1 \leq k \leq 4$, $1 \leq i \leq m$. The *PQR* obtained from *Algorithm 4.6* is denoted as $(\omega_k), 1 \leq k \leq 4$, represented in *Eqn. (4.19)* and *Eqn. (4.20)*.

ALGORITHM 4.6: Calculate the PQR values using CRT
Input: $\{\boldsymbol{ISS_5}\}$ *and* $\{\boldsymbol{ISS_6}\}$, *primes* $\boldsymbol{\mu_k}$ $\boldsymbol{1 \leq k \leq 4}$.   *Output: PQR values as A= {four values from CRT* $(\boldsymbol{\omega_k})$*}.*   1. *Initialize X=0, A= {}*   2. *Repeat steps 1 to 5 for all combinations of* $\boldsymbol{\mu}$ 3. $\boldsymbol{M = \mu_k * \mu'_k}$ *and* $\boldsymbol{M_1 = \frac{M}{\mu_k}}$ *and* $\boldsymbol{M_2 = \frac{M}{\mu'_k}}$; $\boldsymbol{M_1\, M_1^{-1} \equiv 1(mod\mu_k)\, and\, M_2\, M_2^{-1} \equiv 1(mod\ \mu'_k)}$.   4. $\boldsymbol{X = (ISS_5 M_1^{-1} \mu'_k + ISS_6 M_2^{-1} \mu_k) mod\ M}$ 5. *A = A. append(X)*

The *PQR* values are fed as payoffs to the MaxiMin-MiniMax game, which focuses on particular textual aspects to distinguish between sarcasm and non-sarcasm. The game is played between two players, Player A and Player B, maximizing Player 'A' and minimizing Player 'B.' The respective strategies of players A and B in the game are $\mathrm{A}_1, \mathrm{A}_2, \mathrm{B}_1$ and B_2 respectively. *Table 4.12* shows the MaxiMin-MiniMax game between two players. We obtain the normalized overall value of the game (ψ_8), which is termed the optimal game score of the sentence.

Table 4.12 MaxiMin-MiniMax game played with PQR values

<table>
<tr><td rowspan="4">Player A</td><td colspan="3">Player B</td><td colspan="2"></td></tr>
<tr><td></td><td>B_1</td><td>B_2</td><td colspan="2">MaxiMin</td></tr>
<tr><td>A_1</td><td>ω_1</td><td>ω_2</td><td>$Min\,\{\omega_1, \omega_2\}$</td><td rowspan="2">$Max\,\{Min\,\{\omega_1, \omega_2\},$
$Min\,\{\omega_3, \omega_4\}\}$
$= f(\omega_1, \omega_2, \omega_3, \omega_4)$</td></tr>
<tr><td>A_2</td><td>ω_3</td><td>ω_4</td><td>$Min\,\{\omega_3, \omega_4\}$</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">MiniMax</td><td>Max
$\{\omega_1, \omega_3\}$</td><td>$Max\,\{\omega_2, \omega_4\}$</td><td colspan="2" rowspan="2">Value of game $\psi_8 =$
$max\begin{Bmatrix} f(\omega_1, \omega_2, \omega_3, \omega_4), \\ g(\omega_1, \omega_2, \omega_3, \omega_4) \end{Bmatrix}$</td></tr>
<tr><td colspan="2">$Min\{Max\,\{\omega_1, \omega_3\}, Max\{\omega_2, \omega_4\}\}$
$= g(\omega_1, \omega_2, \omega_3, \omega_4)$</td></tr>
</table>

To detect the sarcasm in a text, we compare the value of the game (obtained in Step 3) with the threshold value of sarcasm. *Eqn. (4.21)* encapsulates the formula used to calculate the sarcasm detection threshold value (τ_8) using Kullback-Leibler (KL) divergence. It is a relative entropy measure that is used to calculate the value of τ_8.

$$\tau_8 = \sum p(\psi_8) \log \frac{p(\psi_8)}{\mu_{\psi_8}} \tag{4.21}$$

The $p(\psi_8)$ is the probability distribution of optimal game score ψ_8, and μ_{ψ_8} is the mean of all values of all game scores. The computed τ_8 for Dataset-1 is obtained as $\tau_8 = 19.714 \approx 20$. Similarly, for Dataset-2, the $= 19.80 \approx 20$. The value of the game $\psi_8 \geq \tau_8$ signifies the "*non-sarcastic*", and $\psi_8 < \tau_8$ tells that the text is "*sarcastic.*" *Algorithm 4.7* demonstrates the proposed methodology of sarcasm detection using UCRT-CGKO.

4.4.1 Numerical Illustration of Proposed Method: UCRT-CGKO

Two sentences are considered for the working of the UCRT-CGKO model for the task of sarcasm detection. We have taken sample examples from the datasets utilized in the proposed study.

In step 1, the calculation of individual sentence scores using *algorithms 4.1* and *4.2* is done based on BOW and Tf-Idf, as shown in *Table 4.13*.

ALGORITHM 4.7: Detection of sarcasm using UCRT-CGKO
Input: four PQR values $\mathbf{1 \leq k \leq 4}$, *sarcasm detection threshold value (*$\boldsymbol{\tau_8}$*)* *Output: sarcasm and non-sarcasm (*$\boldsymbol{\psi_8}$*)* 1. *Play the MaxMini-MiniMax game with PQR values obtained from Algorithm 4.6: MiniMax = min (*$\boldsymbol{\omega_i}$ *values) and MaxMini = max (*$\boldsymbol{\omega_i}$ *values)* 2. $\boldsymbol{\psi_8}$ *= max (MiniMax, MaxMini)* 3. *Classify the sentence as sarcastic or non-sarcastic based on optimal game score* $\boldsymbol{\psi_8}$ *and* $\boldsymbol{\tau_8}$*: if* $\boldsymbol{\psi_8} < \boldsymbol{\tau_8}$*: classification = "sarcasm"* *else: classification = "non-sarcasm"* 4. *Exit*

S2: "New Study finds reading comprehension down amongst dumb bullshitt perusing this headline" ***S3: "Oh please, don't care too much."***

In step 2, the target of sarcasm indicator is depicted using the Python library YAKE to get the triggered context (k_{max}^5). *Table 4.14* depicts the sarcastic triggered indicator presence with the help of the keyword extractor it utilized.

Table 4.13 Sentence scores representation using Tf-Idf and BOW

Sentences	$\boldsymbol{ISS_5}$	$\boldsymbol{ISS_6}$
S1	*12*	*3.46*
S2	*6*	*2.4*

Table 4.14 Triggered scores of sarcasm indicator representation

Sentences	***Triggered signal of sarcasm (***$\boldsymbol{k_{max}^5}$***)***	***Target***
S1	0.2973	*"dumb bullshitt perusing this headline "*
S2	0.896	*"don't care too much"*

In step 3, we perform the CRT model implementation to get the PQRs values (ω) using *algorithm 4.6*. *Table 4.15* gives the PQR values of the sentences S1 and S2, respectively.

Table 4.15 PQR values using CRT analysis

Sentences	$\boldsymbol{\omega}$
S1	$\{8.569, 28.10, 7.746, 55.21\}$
S2	$\{29.39, 21.94, 1.439, 4.888\}$

Additionally, the sarcasm detection task is to be performed using the cooperative gameplay of the maximin-minimax procedure, which represents the likelihood distribution model. Based on the obtained value of the game (ψ_8) using *algorithm 4.4*, we perform sarcasm detection on it. *Table 4.16* shows the numerical representation.

Table 4.16 Sarcasm detection task using UCRT-CGKO

Sentences	$\boldsymbol{\psi_8}$	***Tag***
S1	8.569	*Sarcasm*
S2	21.94	*Non-sarcasm*

Therefore, on the basis of decision analysis, we deduce that $\psi_8 < \tau_8$ implies that sentence S1 is tagged as "*sarcasm*" and for sentence S2 $\psi_8 \geq \tau_8$ is tagged as "*non-sarcasm*," respectively.

4.5 Proposed Method 9: Sarcasm Detection Based on Gumble Distributional Wilson's Optimization

In this section, we presented the proposed method of sarcasm detection using Wilson's theorem, utilizing ensemble language models (SDWT-ELM). It is a mathematical optimization-based model that employs three content-based techniques, i.e., Glove, Word2Vec, and ElMo, with Wilson's theorem, to ascertain whether the text is sarcastic. The proposed method works by taking different ensemble approaches and combining them to achieve better outcomes. The result of all three components is a vector space of embeddings, which is then concatenated and gives a summed value. This value is sent to optimize Wilson's theorem, which detects sarcasm and non-sarcasm with the help of a summed value in the given text. *Fig. 4.4* suggests the proposed method flowchart.

The construction of the given proposed approach utilizes word-embedding strategies where words are converted into vectors using GloVe, Elmo, and Word2Vec, which are then added to give a combined numeric value we termed as epitome score. This epitome score is then fetched into Wilson's theorem, which generates the entire sentence value, either prime or non-prime. The value in terms of the prime score tagged the sentence as sarcastic, and the non-prime score tagged the sentence as non-sarcastic.

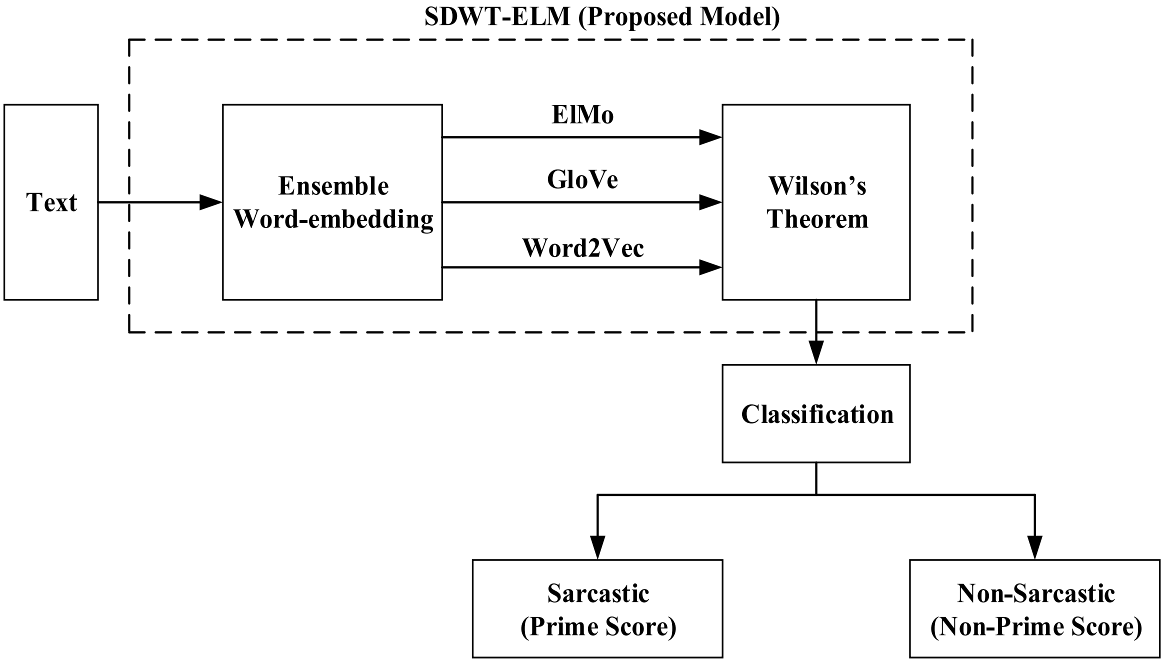


Fig. 4.4 Architectural framework of SDWT-ELM for sarcasm detection

Given dataset X in *Eqn. (4.22)* with sentence x_i, $1 \leq i \leq m$, and m is the total number of sentences.

$$X = \{x_1, x_2, x_3, \ldots., x_i\} \tag{4.22}$$

We initialize here each text of the dataset X and extract predefined word vectors numeric value from embeddings, i.e., GloVe denoted as e_i^{GLV}, ElMo denoted as e_i^{ELM} and Word2vec denoted as e_i^{W2V} where, e is the embedding vector. *Algorithms 4.8*, *4.9*, and *4.10* depict the computation of each of these word embeddings to get the defined sentence value ISS_7, ISS_8 and ISS_9 for glove, word2vec, and Elmo, respectively, using them.

After getting the sentence scores, the epitome score is computed, which is defined as the summation of all three-word embedding values, denoted as ψ_9 obtained in *Eqn. (4.23)*.

$$\psi_9 = \sum_{i=1}^{m} (ISS_{7_i} + ISS_{8_i} + ISS_{9_i}) \tag{4.23}$$

This ψ_9 of each text is then passed to Wilson's algorithm to obtain the Wilson score. Wilson's score is defined as an optimal value utilizing Wilson's Theorem (defined in *Eqn. (4.22)*) in *Eqn. (4.24)* dealing with the given ψ_{9_i} formulated as,

$$(\psi_{9_i} - 1)! \equiv -1 \bmod \psi_{9_i} \tag{4.24}$$

ALGORITHM 4.8: Calculate the sentence value using the glove
Input: $\boldsymbol{W} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}, \ldots, \boldsymbol{w_{j,}})$ *where* $\boldsymbol{w_j}$ *represents the* $\boldsymbol{j^{th}}$ $(\boldsymbol{1} \leq \boldsymbol{j} \leq \boldsymbol{m})$ *word in X.* *Output: sentence value* $\boldsymbol{ISS_7}$ *using a glove.* *1. Load pre-trained glove embedding vector → glove_model.* *2. Tokenize and retrieve the vectors for the sentence.* *3. Sum the vectors to create a sentence vector embedding* $\boldsymbol{vector_{embed_j}}$*:* *Sum vector of sentence* $\boldsymbol{F} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{j=m}} \boldsymbol{vector_{embed_j}}$*.* *4. Initialize* $\boldsymbol{e^{GLV}} = 0$ *for each* $\boldsymbol{w}$ $\boldsymbol{e^{GLV}} = \boldsymbol{e^{GLV}} + \boldsymbol{F_j} \rightarrow \boldsymbol{ISS_7}$ *(output)* *5. End*

ALGORITHM 4.9: Calculate the sentence value using word2vec
Input: $\boldsymbol{W} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}, \ldots, \boldsymbol{w_{j,}})$ *where* $\boldsymbol{w_j}$ *represents the* $\boldsymbol{j^{th}}$ $(\boldsymbol{1} \leq \boldsymbol{j} \leq \boldsymbol{m})$ *word in X.* *Output: sentence value* $\boldsymbol{ISS_8}$ *using word2vec.* *1. From genism. models import KeyedVectors* *2. Calculate sentence score (sentence, word2vec):* *embedding = word2vec.embed_sentence(tokens)* *sentence_embedding = mean(embedding)* *sentence_vector = sum(sentence_embedding)* *3. For each embedding of the set of words* $\boldsymbol{w_j}$*,* *Sentence embeddings,* $\boldsymbol{u_e} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{j=m}} (\boldsymbol{sentence\_vector_j})$*.* *4. Initialize* $\boldsymbol{e^{W2V}} = \boldsymbol{0}$ *for each* $\boldsymbol{w_j}$ $\boldsymbol{e^{W2V}} = \boldsymbol{e^{W2V}} + \boldsymbol{u_{e_j}} \rightarrow \boldsymbol{ISS_8}$ *(output)* *5. End*

ALGORITHM 4.10: Calculate the sentence value using ElMo
Input: $\boldsymbol{W} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}, \ldots, \boldsymbol{w_{j,}})$ *where* $\boldsymbol{w_j}$ *represents the* $\boldsymbol{j^{th}}$ $(\boldsymbol{1 \leq j \leq m})$ *word in X.* *Output: sentence value* $\boldsymbol{ISS_9}$ *using ElMo.* 1. *Load pre-trained ELMo model* 2. *Tokenize sentence X into words: Let* $\boldsymbol{W} = [\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_j}]$ 3. *For each word* $\boldsymbol{w_i}$ *in* $\boldsymbol{W}$*: Pass* $\boldsymbol{w_i}$ *through the ELMo model* 4. *Extract the ELMo representation for* $\boldsymbol{w_i}$ *(ELMo typically gives 3 layers of vectors: [L1, L2, L3] per word)* *Combine the 3 layers into a single vector: vi = sum(L1, L2, L3) // For simplicity* 5. *Aggregate word vectors to form sentence embedding:*$\boldsymbol{ISS_8} = \boldsymbol{sum(v_i)}$ 6. *Return* $\boldsymbol{ISS_9}$ *as the sentence embedding* 7. *End*

ALGORITHM 4.11: SDWT-ELM algorithm for detection of sarcasm
Input: X, each context $x_i \in X$ *and* $1 \leq i \leq m$ *where* m *is the total number of sentences,* ψ_9 *Output: Prime score and non-prime score* *1. Define* $\rightarrow \psi_9$ *2. if* $\psi_9 \geq 1$: *3. execute* $\rightarrow (\psi_9 - 1)! \equiv -1 \; mod \; \psi_9$ *4. return* ψ_9 $\rightarrow$ *(prime score)*$\rightarrow$ *sarcasm* *5. else: return* $\rightarrow$ *non-sarcasm* *6. end*

The ψ_{9_i} is used to give the value in terms of prime and non-prime. Based on this prime value, we deduced that if ψ_9 is computed as the prime score, the sentence is tagged as "*sarcastic.*" Else, it is tagged as "*non-sarcastic*" in nature. The elaborated text of it is shown in *Eqn. (4.26)* and *Eqn. (4.25)*.

$$\psi_{9_i} = \textit{Prime score} \text{ (sarcastic)} \tag{4.25}$$

$$\psi_{9_i} = \textit{Non-Prime score} \text{ (non-sarcastic)} \tag{4.26}$$

Our proposed method SDWT-ELM aims to learn with mathematically optimized word embeddings to detect text as sarcastic or not. *Algorithm 4.11* derives a fixed size representation of a context set for this purpose by computing the elementwise maximum addition for every word vector and the final size of the sentence representation given by the ψ_9. This score uses Wilson's theorem to get the text value, with a prime score represented as sarcasm and a non-prime score as non-sarcasm.

4.5.1 Numerical Illustration of Proposed Method: SDWT-ELM

The exemplar sentences have been taken for the proposition of the proposed method SWD-ELM system. The sentences taken here are from the datasets utilized in this study.

S1: "The first professional baseball team was from an Ohio city."
S2: "speak up and give back if you want the economy to improve."

Table 4.17 Word embedding scores computation

Sentences	$\boldsymbol{ISS_8}$	$\boldsymbol{ISS_7}$	$\boldsymbol{ISS_6}$
S1	*4.8*	*4.3*	*4.3*
S2	*1.1*	*1.3*	*1.1*

Table 4.18 Sarcasm evaluation based on Wilson's Theorem

Sentences	***Wilson's theorem (ψ_9)***	***Tag***
S1	$(13-1)! \equiv -1 \bmod 13 = 0$	*Sarcasm*
S2	$(4-1)! \equiv -1 \bmod 4 \neq 0$	*Non-sarcasm*

In step 1, we determine the word embedding of text using Elmo, Word2Vec, and GloVe. *Table 4.17* shows the computed score of text from these embeddings. After fetching these scores into Wilson's theorem, we get the optimal score for sarcasm detection. *Table 4.18* shows the computation of Wilson's score for sarcasm detection. The statement S1 is deduced to a "*sarcastic*" label, where ψ_9 is the prime value, whereas S2 is labeled as "*non-sarcastic*" as it is an obtained non-prime value.

4.6 Experimental and Result Analysis

In this section, we present the analysis of the different collected datasets utilized in defined models. The performance of proposed methods over these datasets are compared with existing state-of-the-art approaches, respectively. Additionally, several metrics are analyzed, and their efficacies are inspected in this subsection. Language datasets are analyzed, with encouraging results.

4.6.1 Dataset Utilisation by Proposed Methods

Table 4.19 represents the statistics of the datasets used by the three proposed methods. This involves the utilization of four benchmark datasets for the task of sarcasm detection. The details of each dataset are given in *Chapter 2*, "*Literature review*," in *section 2.5*.

Table 4.19 Statistical variations of datasets

S. No.	*Datasets*	*Renamed as*	*Total number of sentences*
1.	*SARC [149]*	*Dataset-1*	*12,257*
2.	*Reddit comments [150]*	*Dataset-2*	*7,895*
3.	*Semeval 2018 task [151]*	*Dataset-3*	*11,025*
4.	*News Headlines [152]*	*Dataset-4*	*12,506*

4.6.2 Performance Evaluation of Proposed Methods on Dataset-1

Whilst the sarcasm detection task on dataset-1, UCRT-CGKO, SD-GKOT, and SLIOEN reached the highest levels of performance, with all four metrics- accuracy, precision, recall, and F1 score- close to 0.955, clearly surpassing most leading contemporary models. The proposed method, SWDT-ELM, also performed well, with slightly lower scores but still above 0.89 across all metrics. Conventional systems like HMM, GMM, SVM, M2Seq2Seq, and handcrafted feature methods, such as SVM+Tf-Idf+BOW, produced moderate results, generally between 0.7 and 0.8 on all metrics. Deep learning models such as DNN, CNN, LSTM [165], and GRU [166] provided competitive results, but the newer systems, especially UCRT-CGKO and SLIOEN, consistently performed better.

Overall, the findings showed that UCRT-CGKO, SLIOEN, and SWDT-ELM set a new standard for sarcasm recognition in text, with SD-GKOT also showing clear improvement over earlier approaches. A comparison analysis or proposed methods with state-of-the-art methods in terms of accuracy, precision, recall, and f1 score is shown in *Fig. 4.5*.

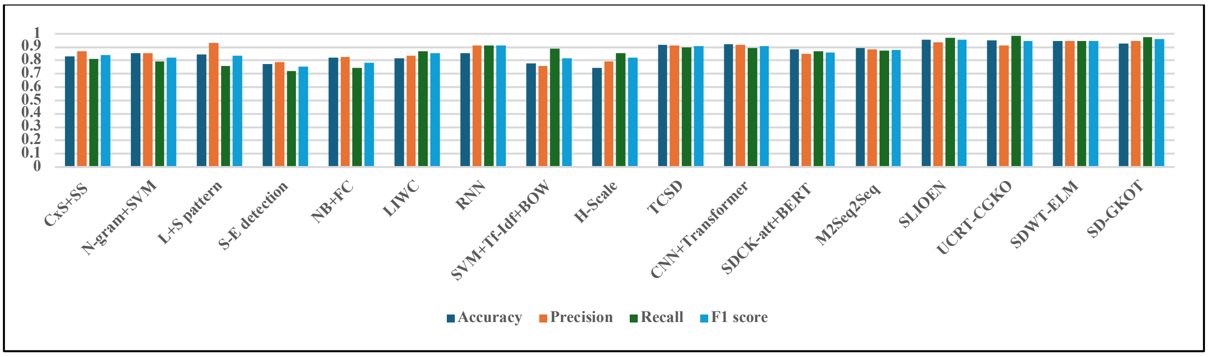


Fig. 4.5 Comparison plot of SLIOEN, UCRT-CGKO, SDWT-ELM and SD-GKOT with the predefined models on dataset-1 [149]

4.6.3 Performance Evaluation of Proposed Methods on Dataset-2

For the sarcasm detection dataset-2, proposed methods SD-GKOT, UCRT-CGKO, and SLIOEN showed the best performance, with accuracy, precision, recall, and F1 scores all approaching 0.969, clearly exceeding the results of most state-of-the-art solutions. The algorithm of SDWT-ELM also performed strongly, with all metrics slightly lower but still above 0.88.

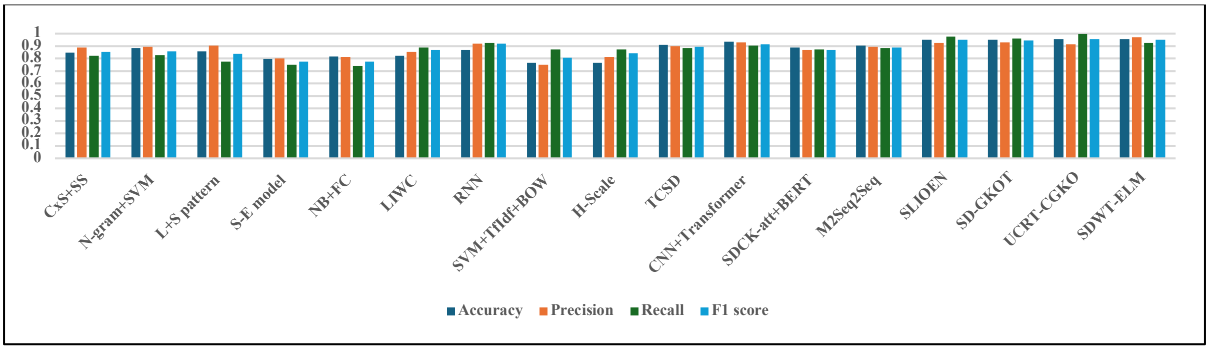


Fig. 4.6 Comparison plot of SLIOEN, SD-GKOT, UCRT-CGKO, and SWD-SEO with the predefined models on dataset-2 [150]

In contrast, classic models like BERT [166], RCNN+RoBERTa [110], and KnowleNet [2], as well as rule-based detection methods such as CNN [156], Bi-LSTM [158], SIARM [74], NF-CL [167], and FAT-MTL [168], achieved moderate scores, typically between 0.7 and 0.8 for all metrics. Transformer-based and ensemble models such as RoBERTa [166], SDCKatt+BERT, and M2Seq2Seq delivered competitive results, but the proposed methods, especially SLIOEN and SD-GKOT, consistently outperformed them.

Overall, the analysis revealed that SLIOEN, SD-GKOT, and UCRT-CGKO set a new benchmark for sarcasm detection, while SDWT-ELM also showed significant progress over traditional techniques. *Fig. 4.6* shows the performance evaluation metric of SLIOEN, SD-GKOT, LDL-SDTE, and SWD-SEO methods.

4.6.4 Performance Evaluation of Proposed Methods on Dataset-3

In the sarcasm classification task, proposed methods, SLIOEN, SD-GKOT, SDWT-ELM, and UCRT-CGKO achieved the highest results, with all four metrics-accuracy, precision, recall, and F1 score-reaching nearly 0.975, clearly outperforming most existing advanced methods. These proposed methods performed well, with recall scores above 0.94 for every metric. In comparison, traditional approaches such as the S-E model, SVM+Tf-Idf+BOW, and H-Scale SVM, Random Forest, and RNN, as well as feature-based techniques, produced moderate outcomes, generally ranging from 0.68 to 0.86 across all metrics. CNN-based and LSTM-based models delivered competitive results, but the proposed architectures, especially SLIOEN and SD-GKOT, consistently surpassed them. Overall, the results showed that SLIOEN, SD-GKOT, and SDWT-ELM set a new standard for sarcasm classification on this dataset, with UCRT-CGKO also demonstrating notable improvement over previous methods. *Fig. 4.7* shows the performance evaluation metric of SLIOEN, SD-GKOT, LDL-SDTE, and SWD-SEO methods.

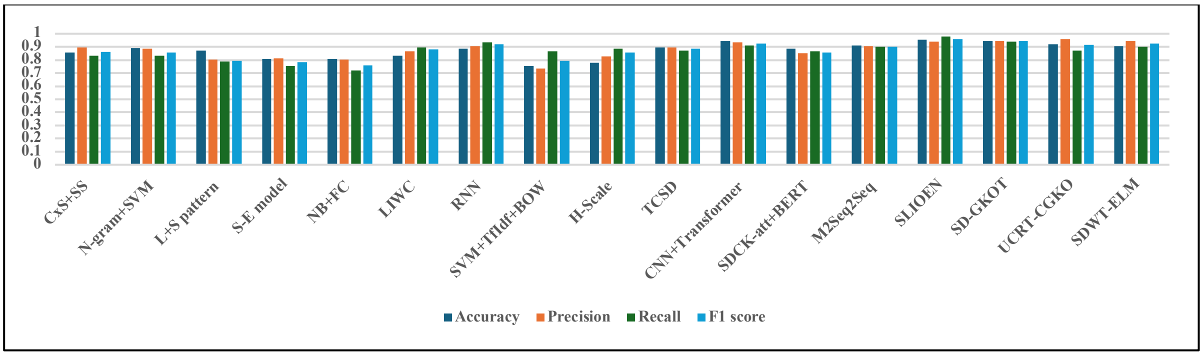


Fig. 4.7 Comparison plot of SLIOEN, SD-GKOT, UCRT-CGKO, and SDWT-ELM with the predefined models on dataset-3 [151]

However, the proposed methods capture better optimality in conditions with essential capturing of subtleties in sarcasm detection, which are overlooked by RNN, TCSD, SDCKatt+BERT, M2Seq2Seq, and CNN+Transformer. This concluded that our proposed method accomplished outstanding results for all datasets, surpassing all existing methods for every indicator. The comparison graphs of the proposed method with other state-of-the-art methods depict the robust performance of SLIOEN, SD-GKOT, SDWT-ELM, and UCRT-CGKO.

4.6.5 Performance Evaluation of Proposed Methods on Dataset-4

On Dataset-4, UCRT-CGKO, SDWT-ELM, and SD-GKOT achieved the highest performance, with all four metrics-accuracy, precision, recall, and F1 score-reaching close to 0.9, clearly surpassing most existing state-of-the-art methods. SLIOEN also performed strongly, with slightly lower scores above 0.85 for all metrics. In contrast, traditional models like CASCADE-CNN, Vanilla-CNN [156], and SMSD-BiLSTM [158], as well as pattern-related feature methods [167], [168] showed moderate results, generally ranging from 0.7 to 0.8 across metrics. Experimental results indicated that the proposed method SD-GKOT and SDWT-ELM outperformed the robust baseline MTL-DNN [46] and RNN-GRU-SVM [169] by averaged 7.66% and 8.02% in terms of accuracy respectively.

BERT-based models [110], BERT+Comet [170], and ELMo-BiLSTM delivered competitive results, but the proposed methods, especially UCRT-CGKO and SDWT-ELM, consistently outperformed them. Overall, the graph demonstrated that UCRT-CGKO, SDWT-ELM, SD-GKOT, and SLIOEN set a new benchmark for sarcasm detection on this dataset, showing marked improvement over earlier approaches. The efficacy indicators of these proposed methods for dataset 4 are illustrated in *Fig. 4.8*.

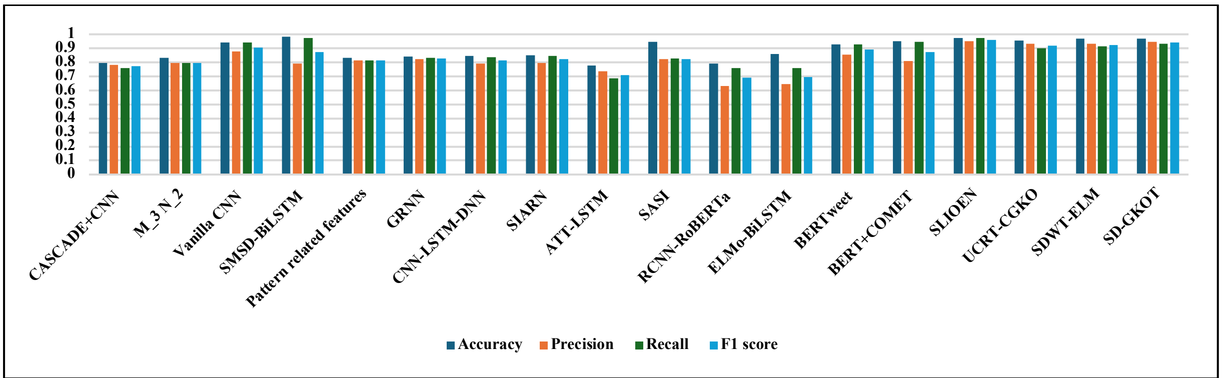


Fig. 4.8 Comparison plot of SLIOEN, UCRT-CGKO, SDWT-ELM and SD-GKOT with the predefined models on dataset-4 [152]

4.7 Discussion

In this section, we present a discussion of the five proposed methods with certain suitable findings. In this section, we present the statistical validation tests and their effectiveness for the proposed methods. Additionally, we discuss the computation cost of time of each of the proposed methods. Furthermore, we present the challenges possessed by the defined proposed methods for sarcasm detection that need to be tackled in future aspects.

4.7.1 Statistical Validations Tests Performed on Four Proposed Methods

4.7.1.1 Statistical Validation using Z-Test

We have already given the formula and its explanation in chapter 3. *Tables 4.20* and *4.21* show the final results of the proposed methods.

Table 4.20 Two Proportion Z-Test statistics across datasets

	SLIOEN		*SD-GKOT*	
Parameters	*Sample 1*	*Sample 2*	*Sample 1*	*Sample 2*
Sample size (N)	*1047*	*769*	*1433*	*664*
Sample proportion (c)	*0.8299*	*0.8231*	*0.8855*	*0.8885*
Favorable cases (X)	*869*	*633*	*1269*	*590*
P	*0.8270*		*0.8886*	
z	*0.3788*		*0.2016*	
Hypothesis status	H_0 *is not rejected*		H_0 *is not rejected*	

Table 4.21 Two Proportion Z-Test statistics across datasets

	UCRT-CGKO		*SDWT-ELM*	
Parameters	*Sample 1*	*Sample 2*	*Sample 1*	*Sample 2*
Sample size (N)	*925*	*497*	*2557*	*1277*
Sample proportion (c)	*0.9189*	*0.9008*	*0.8263*	*0.8285*
Favorable cases (X)	*850*	*447*	*2113*	*1058*
P	*0.9153*		*0.8270*	
z	*0.669*		*0.1697*	
Hypothesis status	H_0 *is not rejected*		H_0 *is not rejected*	

For two population proportions (c_1 and c_2), null hypotheses (H_0) and alternative hypotheses (H_a) are defined as follows:

H_0: $c_1 = c_2$, i.e., the accuracy score of sample 1 = accuracy score of sample 2

H_a: $c_1 \neq c_2$, i.e., the accuracy score of sample 1 ≠ , the accuracy score of sample 2

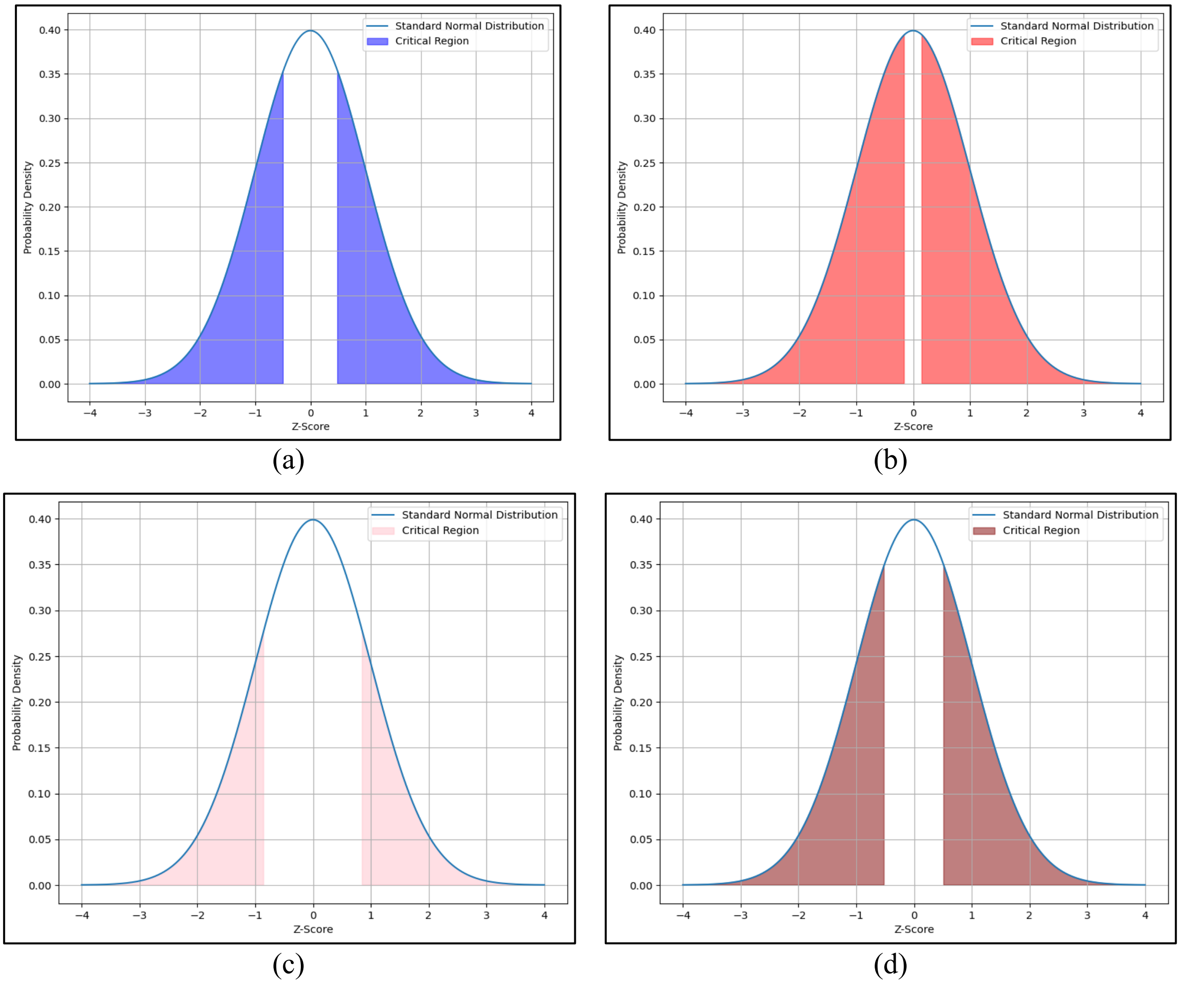


Fig. 4.9 Statistical z-test plots for the proposed methods (a) SLIOEN, (b) SD-GKOT, (c) UCRT-CGKO, and (d) SDWT-ELM

In the proposed method, the condition of the null hypothesis is accepted when the critical range z– value at $\alpha = 0.05$ in *Eqn. (3.15)* gives the value of the two sum proportions of two samples less than the computed z score, also called z-statistics in *Eqn. (3.16)*.

Figs. 4.9 (a), *(b)*, *(c)*, and *(d)* graphically depict the accepted critical region of the hypothesis test. The null hypothesis (H_0) was not rejected, indicating that there is insufficient evidence to suggest that the population proportions (c_1) and (c_2) do not differ at a significance level of

($\alpha = 0.05$). This finding implies that the accuracy of our model remains consistent across different sample sizes from the same datasets.

4.7.1.2 Statistical Validations using Kolmogorov-Smirnov (KS) Test

In the Kolmogorov-Smirnov (KS) test, we compared two independent samples taken from the two datasets. The aim is to determine whether the two samples have similar distributions without assuming any change in the particular distribution of the data based on the empirical cumulative distribution function (ECDF). *Table 4.22* shows how this test works.

For two population proportions (c_1 and c_2), null hypotheses (H_0) and alternative hypotheses (H_a) are defined as follows:

$$H_0: c_1 = c_2, \text{ i.e., accuracy of sample 1} = \text{accuracy of sample 2}$$

$$H_a: c_1 \neq c_2, \text{ i.e., the accuracy of sample 1} \neq \text{accuracy of sample 2}$$

Table 4.22 Two Proportion KS-Test statistics across datasets

	SD-GKOT	
Parameters	*Sample 1*	*Sample 2*
Sample size (N)	*1147*	*828*
Sample proportion (c, ECDF)	*0.9546*	*0.9553*
Favorable cases (X)	*1095*	*791*
D- value (D_{N_1, N_2})	*0.9553*	
D_α	*0.3101*	
Hypothesis status	*H_0 is not rejected*	

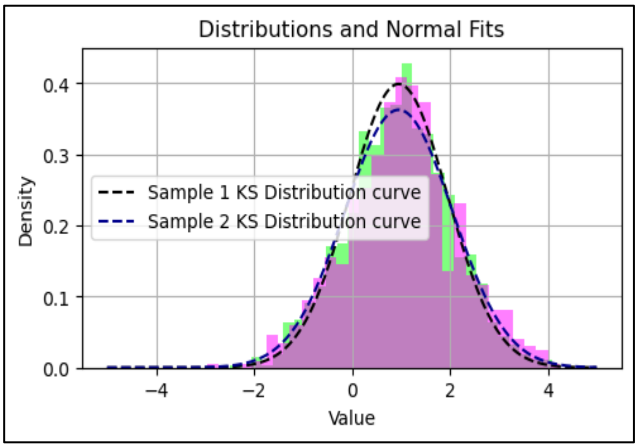


Fig. 4.10 Statistical validation of SD-GKOT using the Kolmogorov-Smirnov Test

Figs. 4.10 depicts the given D- value calculated as 0.9553, greater than the significance level $\alpha = 0.05$ where the two curves' peaks meet. This implied that the null hypothesis had been accepted, signifying that the two given samples belonged to the same distribution.

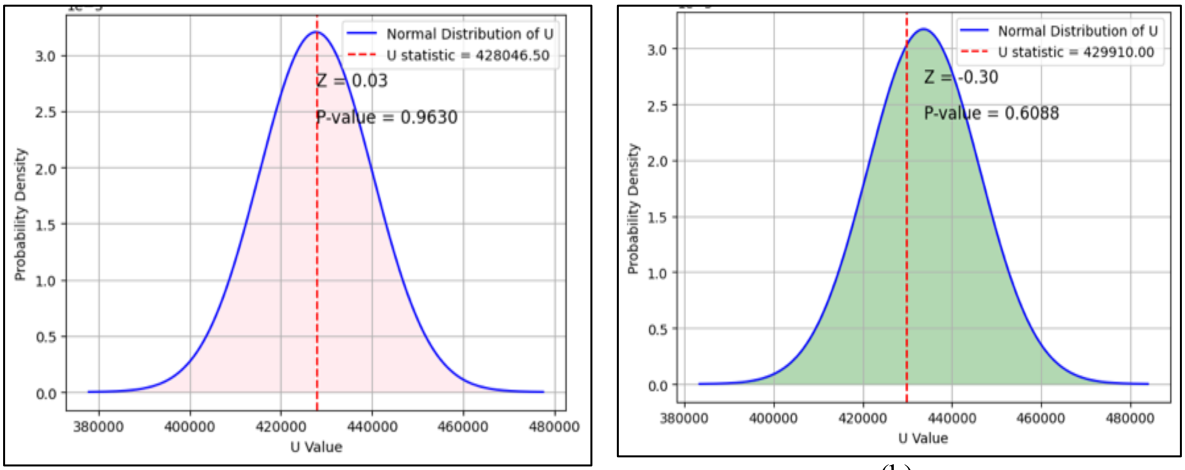
4.7.1.3 Statistical Validation using Mann-Whitney U Test

The Mann-Whitney U test is used to determine whether or not two distinct samples originating from different populations have the same distribution. Based on the provided sample data, we constructed a distribution by employing the mean and standard deviation from the number of appropriately categorized sentences in each sample shown in *Table 4.23*.

Figs. 4.11 (a) and *(b)* depicts the given z-value calculated as 0.07 (using *Eqn. (3.21)*) and 1.18 (using *Eqn. (3.22)*), greater than the significance level $\alpha = 0.05$ where the two curves' peaks meet. This implied that the null hypothesis had been accepted, signifying that the two given samples belonged to the same distribution.

Table 4.23 Computation of Mann-Whitney U test results for SDO-BCRT and SD-GKOT

	SLIOEN		***SD-GKOT***	
Parameters	*Sample 1*	*Sample 2*	*Sample 1*	*Sample 2*
Sample size (N)	*1661*	*515*	*1668*	*520*
Sample proportion (ECDF)	*0.8687*	*0.8679*	*0.8663*	*0.8750*
Favorable cases (X)	*1443*	*447*	*1445*	*455*
U- value	*0.8679*		*0.8679*	
$\boldsymbol{mean_{N_1,N_2}}$	*0.3769*		*0.3769*	
$\boldsymbol{Std_{N_1,N_2}}$	*0.4146*		*0.4146*	
$\boldsymbol{z_B}$	*0.07*		*1.18*	
Hypothesis status	H_0 *is not rejected*		H_0 *is not rejected*	



(a) (b)

Fig. 4.11 Statistical validation of (a) SLIOEN and (b) SD-GKOT using the Mann-Whitney U Test

4.7.2 Ablation study of SLIOEN, UCRT-CGKO and SD-GKOT

An ablation test has been conducted to determine the impact of essential parameters used in the proposed method. The essential parameter whose change in value affects the performance of the proposed method is a threshold value for sarcasm detection (τ_6, τ_7, τ_8). *Table 4.24* revealed the change in these values compared to the actual computed scores of 0.4, 0.5, and 20, respectively.

In the proposed sarcasm detection models, various threshold scores significantly influenced the evaluation results across all datasets. When the threshold parameter ω was changed from the optimal value of 0.5 to values greater than 0.5, there was a noticeable drop in performance.

Specifically, the number of false negatives has increased, which caused sarcastic sentences to be incorrectly labelled as non-sarcastic. This misclassification reduced the recall and accuracy scores. Similarly, using τ_6 values less than 0.5 led to a rise in false positives, where non-sarcastic sentences were wrongly predicted as sarcastic. This caused a decline in precision and overall model accuracy. These outcomes highlighted that deviating from $\tau_6 = 0.5$ weakened the performance of the SLIOEN model in terms of accuracy, precision, recall, and F1-score.

Moreover, the parameter τ_7 played a vital role in enhancing the semantic understanding of sarcasm in the SD-GKOT method. When τ_7 was set at 0.4, the model achieved improved performance with fewer false positives and false negatives. However, using τ_7 values greater than 0.4 led to more false negatives and lower recall. On the other hand, setting τ_7 below 0.4 increased false positives, reducing precision and accuracy. These changes confirmed that the fixed $\tau_7 = 0.4$ value was essential for stable performance.

Table 4.24 Ablation study on proposed methods SLIOEN, SD-GKOT and UCRT-CGKO

Dataset	Removed module	Accuracy	Precision	Recall	F1- score
Dataset-1 [149]	$\tau_6 > 0.5$	*0.8782*	*0.8656*	*0.9075*	*0.8860*
	$\tau_6 < 0.5$	*0.8454*	*0.9218*	*0.8980*	*0.9097*
	SLIOEN $\boldsymbol{\tau_6 = 0.5}$	***0.9543***	***0.9386***	***0.9693***	***0.9537***
	$\tau_7 < 0.4$	*0.8263*	*0.7879*	*0.8997*	*0.8380*
	$\tau_7 > 0.4$	*0.7447*	*0.8819*	*0.7082*	*0.7855*
	SD-GKOT $\boldsymbol{\tau_7 = 0.4}$	***0.9446***	***0.9460***	***0.9396***	***0.9428***
	$\tau_8 < 20$	*0.8116*	*0.7706*	*0.8672*	*0.8215*
	$\tau_8 > 20$	*0.7564*	*0.8901*	*0.7143*	*0.7954*
	UCRT-CGKO $\boldsymbol{\tau_8 = 20}$	***0.9355***	***0.9380***	***0.9429***	***0.9386***
Dataset-2 [150]	$\tau_6 > 0.5$	*0.8849*	*0.8322*	*0.9080*	*0.8684*
	$\tau_6 < 0.5$	*0.8683*	*0.9167*	*0.8568*	*0.8857*
	SLIOEN $\boldsymbol{\tau_6 = 0.5}$	***0.9539***	***0.9405***	***0.9796***	***0.9596***
	$\tau_7 < 0.4$	*0.8001*	*0.7631*	*0.8725*	*0.8141*
	$\tau_7 > 0.4$	*0.7388*	*0.8738*	*0.7030*	*0.7791*
	SD-GKOT $\boldsymbol{\tau_7 = 0.4}$	***0.9457***	***0.9743***	***0.9107***	***0.9414***
	$\tau_8 < 20$	*0.8895*	*0.8472*	*0.9470*	*0.8951*
	$\tau_8 > 0.4$	*0.7780*	*0.9087*	*0.7644*	*0.7383*
	UCRT-CGKO $\boldsymbol{\tau_8 = 0.4}$	***0.9678***	***0.9419***	***0.9800***	***0.9626***
Dataset-3 [151]	$\tau_6 > 0.5$	*0.8788*	*0.8517*	*0.8996*	*0.8749*
	$\tau_6 < 0.5$	*0.8625*	*0.9371*	*0.8232*	*0.8764*
	SLIOEN $\boldsymbol{\tau_6 = 0.5}$	***0.9516***	***0.9236***	***0.9751***	***0.9486***
	$\tau_7 < 0.4$	*0.8559*	*0.8108*	*0.9204*	*0.8621*
	$\tau_7 > 0.4$	*0.7503*	*0.8987*	*0.7254*	*0.7149*
	SD-GKOT $\boldsymbol{\tau_7 = 0.4}$	***0.9488***	***0.9282***	***0.9607***	***0.9442***
	$\tau_8 < 0.4$	*0.8459*	*0.8268*	*0.8724*	*0.8456*
	$\tau_8 > 0.4$	*0.7735*	*0.9044*	*0.7641*	*0.7464*
	UCRT-CGKO $\boldsymbol{\tau_8 = 0.4}$	***0.9426***	***0.9379***	***0.9520***	***0.9631***
Dataset-4 [152]	$\tau_6 < 0.5$	*0.8759*	*0.8422*	*0.9376*	*0.8829*
	$\tau_6 > 0.5$	*0.7746*	*0.8647*	*0.7954*	*0.7656*
	SLIEON $\boldsymbol{\tau_6 = 0.5}$	***0.9632***	***0.9420***	***0.9584***	***0.9527***
	$\tau_7 < 0.4$	*0.8559*	*0.8108*	*0.9204*	*0.8621*
	$\tau_7 > 0.4$	*0.7503*	*0.8987*	*0.7254*	*0.7149*
	SD-GKOT $\boldsymbol{\tau_7 = 0.4}$	***0.9508***	***0.9382***	***0.9532***	***0.9647***
	$\tau_8 < 20$	*0.8895*	*0.8538*	*0.9156*	*0.8701*
	$\tau_8 > 20$	*0.7677*	*0.8848*	*0.7531*	*0.7249*
	UCRT-CGKO $\boldsymbol{\tau_8 = 20}$	***0.9745***	***0.9672***	***0.9521***	***0.9533***

In addition, modifying the τ_8 threshold beyond 20 or below 20 in the UCRT-CGKO model caused a deflection in the predicted outcomes, resulting in lower sarcasm detection rankings. This showed that maintaining $\tau_8 = 20$ was critical for consistent results. Finally, the contextual presence of primes, formed using a combination of four prime numbers, also influenced detection quality, especially when paired with the threshold $\tau_8 = 20$.

4.7.3 Computation Cost of Proposed Methods

The time complexity (T) measures the number of computational steps needed to accomplish a given task on a specific input dataset.

Table 4.25 Computational time complexities of the four proposed methods

S. no.	***Proposed methods***	***Time complexity***
1	*SLIOEN*	$O(m + log\, m)$
2	*SD-GKOT*	$O(m + log(mn))$
3.	*UCRT-CGKO*	$O(nm)$
4.	*SDWT-ELM*	$O(n + \log(n))$

We evaluate the time complexity ($T(n, m)$) of the algorithms under different scenarios, where n denotes the number of alternative operations and m represents the number of sentences. The values of $T(n, m)$ Indicate the performance of the three defined algorithms in *Table 4.25*.

4.7.4 Challenges of the Four Proposed Methods

There are some challenges undertaken by the four proposed methods based on optimization techniques using different probabilistic distributions. *Table 4.26* shows the tags that are mislabelled by the proposed methodologies for sarcasm detection.

Table 4.26 Misclassification by four proposed methods

Sentences	***Actual tag***	***Proposed method tags***
"Oh fantastic!!!! Another meeting that could have been an email	*Sarcasm*	*Non-sarcasm*
Damn… man Youuuul ok greattt nerdy…	*Sarcasm*	*Non-sarcasm*
Wow.. it was soo organized, loved waiting in long lines.	*Sarcasm*	*Non-sarcasm*

The following are some major issues faced by the proposed methodologies:

a) The proposed methods failed to give up the interpretation of sentence type in terms of literal and non-literal speech, with most rhetorical questions.

b) It also showed the difficulty in understanding the long-range dependence of implicit and explicit sarcastic tone in sentences.

4.8 Summary

This chapter introduces four hybrid optimization-based methods for sarcasm detection, each leveraging advanced mathematical optimization techniques. The proposed method, SLIOEN, detects sarcasm using logistic distribution-based ion motion optimization. The other proposed method, SD-GKOT, provides a method for sarcasm detection with subtle cues using the Gauss-Kuzmin optimization technique. Additionally, SDWT-ELM applies Wilson's technique with Gumbel distribution for probabilistic analysis, effectively identifying sarcasm through full language marker interpretation. Furthermore, UCRT-CGKO combines the Chinese Remainder Theorem with Game Theory to detect sarcasm.

These proposed methods collectively address the shortcomings of existing sentiment analysis approaches, especially in handling erroneous or ambiguous content in social media. The proposed techniques demonstrate improved accuracy and contextual understanding in sarcasm detection with novel interpretations of mathematical optimization techniques. The chapter highlights the importance of mathematical optimization and contextual interpretation in advancing sarcasm analysis. Future work may involve refining these models to detect different sarcasm types and further analysing linguistic traits.

Chapter 5
Categorization and Detection of Types of Sarcasm in English Text

Once we detect that the text or comment is sarcastic, the follow-up step is to detect the type of sarcasm. Types of sarcasm refer to the distinct stylistic and emotional variations present in sarcastic sentences conveying negative intent in text. Types of sarcasm come in different forms, such as hyperbole, rhetorical devices, dropped negation, offensive, humorous, humblebrag, and oxymoron in nature. Each of these types differs in its tone, purpose, and linguistic structure, posing a unique and clearer interpretation.

Understanding and classifying types of sarcasm helps in developing more nuanced and effective models. This leads to better working of NLP models with more accurate sarcasm analysis across various applications. It is important to categorize types of sarcasm to better understand the real meaning behind words and help machines interpret text more accurately.

Traditional models for sarcasm detection focus on a binary classification problem, but neglect classifying and detecting its types. This chapter introduces categorization and detection for types of sarcasm in English text using various optimization techniques.

The sections in this chapter are structured as follows. In *Section 5.1*, we provide background knowledge of types of sarcasm and their usefulness. *Sections 5.2* to *5.4* give details for categories of sarcasm with their subtypes. *Sections 5.5* to *5.7* give the elaborated working of the proposed methods using various optimization techniques for the categorization of sarcastic English text. *Section 5.8* provides detailed experimentation and results analysis of all the proposed methods. Finally, in *Section 5.9*, we summarize the proposed methodologies.

5.1 Introduction

Types of sarcasm involve categorizing sarcastic expressions into their specific sub forms. The types of sarcasm include learning linguistic, communicative, and intentional characteristics [7] of sarcastic sentences.

For example, consider a sarcastic statement, "*I'm a genius when it comes to chatting up new acquaintances*". This statement is classified as '*humblebrag sarcasm*' due to its self-mocking nature. *Fig. 5.1* gives an illustration of some of the types of sarcastic comments in English text.

Fig. 5.1 Types of sarcastic comments in English text

We have also categorized various subtypes of sarcasm into different groups. The three subcategories present this thesis are given below:

- Meaning Inversion Sarcasm
- Surface-Depth Sarcasm
- Masked Intentions Sarcasm

Understanding types of sarcasm significantly enhances various NLP tasks, including opinion mining, information retrieval, and market research [45]. Additionally, it enhances the intent of tonal traits, giving clarity in communication, like customer service, social media monitoring, and mental health support [171]. Furthermore, it helps AI systems to interpret nuanced human communication more effectively, with context-aware interactions.

Each type of sarcasm conveys a different intent and inherent undertone, making it crucial for accurate sarcasm interpretation. Some studies [172] and [173] have emphasized the need to classify types of sarcasm for capturing sarcasm as a negative intent of expression. Despite the advancements in sarcasm detection, the area of classification of its types has received less attention. Some of the conventional classification techniques [29], [109], [50] showed effective sarcasm detection but ignored the task of identifying its types.

Various machine learning [42], [97], [45] and ensemble models [53], [50] have performed the sarcasm detection task, but neglected to give the interpretation of its subtypes. Similarly, deep learning [70], [174], [171] and transformer-based studies [31], [87] focused only on sarcasm detection yet failed to further distinguish between varying sarcastic subtypes. Many studies [69], [110], [157] relied on unrealistic dataset balancing, often using balanced sarcasm datasets. However, in the case of types of sarcasm, there are relatively rare annotated real-world datasets.

Limited contextual understanding is another limitation, where utterances are not clearly analysed without background knowledge [174], [171]. Moreover, a lack of fine-grained differentiation prevents models from recognizing specific sarcasm types, which often leads to vague or incorrect categorisation.

To address these limitations, we propose novel frameworks for detecting types of sarcasm in English text using various optimization techniques.

5.2 Meaning Inversion Sarcasm

This categorization learns those sarcastic sentences where the literal meaning of a statement is intentionally flipped to express the opposite of what is said. The speaker says one thing but means another, typically the reverse. This is a core mechanism of sarcasm, helping to express criticism, humour, or mockery in indirect ways.

For example, a friend arrives very late, and the other friend says, "*Wow, you're so punctual!*" The speaker in the sentence did not actually mean to say punctual to that person, but used it in a taunting way. Within meaning inversion, several specific subtypes of this categorization are identified as follows.

- Rhetorical devices
- Hyperbole
- Dropped negation
- Like-prefixed

Detecting the meaning inversion sarcasm is crucial for sarcasm detection because the literal words alone often misinterpret the speaker's true intent. Without recognizing meaning inversion, readers or listeners may misinterpret sarcastic remarks as sincere. This can lead to misunderstandings in communication and weak model performance in NLP tasks. *Fig. 5.2* gives the subcategories of meaning inversion sarcasm.

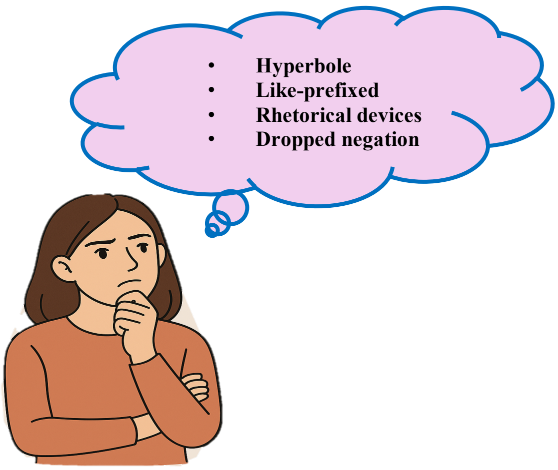


Fig. 5.2 Subtypes of meaning inversion sarcasm

Meaning inversion sarcasm happens when a person says something positive, but the real intention is negative or opposite to the literal words. This type of sarcasm depends on language twists where the speaker deliberately uses exaggeration, or reversed expressions to mock or criticize.

It often includes features such as hyperbole (overstatement), like-prefixed phrases (e.g., "like that's ever going to happen"), or dropped negation (omitting "not" to change the tone). For example, *"Oh, you're such a genius for forgetting your wallet again."* Here, the speaker does not actually praise the person; instead, the statement is intended to ridicule them for being careless.

5.2.1 Rhetorical Devices

This type of sarcasm often leverages understatement and rhetorical questions to achieve its effect. For example, asking *"Do you even know how to drive?"* instead of knowing that the other person doesn't know how to drive, but asking in a sarcastic tone, with an insult. Consider another statement saying, "*What planet are you living on? Romulus?*" This sentence shows the presence of mocking by the speaker to the other person, asking such stupid questions sarcastically.

5.2.2 Hyperbole

This type of sarcasm involves exaggeration to emphasize the absurdity or criticize the behaviour of the other person. For example, saying "*You're as smart as a genius!*" to someone who has done something foolish. Consider another example, *"I waited a thousand years for you!"* said by the person after a short delay, to emphasize impatience.

5.2.3 Dropped Negation

This type of sarcasm occurs when a word or phrase, initially used mockingly with positive wit, drops the negation "*not*" in the context. For example, the word "*awesome*" is used sarcastically to describe something "*not-awesome*," to the point where this negative connotation is widely understood.

The phrase "*I love how...*" almost universally means "*I hate how...*" in certain contexts, used sarcastically to express displeasure. "*You're doing just fine!*" is said when someone is clearly failing, implying they are not doing well. The sentence omits a negation, flipping the literal meaning.

5.2.4 Like-Prefixed

This specific type of sarcasm is identified in linguistic analysis. It is characterized by the use of "*like*" as a prefix. The sentence starts with "*like*" to mock a statement. For example, "*What are you, like, stuck in the 1950s?*" The speaker indirectly mocks or teases the other person in a taunting way, using "*like*" as positive sentiment. Consider another sentence, *"Like I care what you think."* The speaker clearly does not care, despite using the positive phrasing "*Like I care*".

5.3 Surface-Depth Sarcasm

In this categorization, the sarcasm is directly stated either as explicitness or subtle indirect cues as implicitness. The example for explicit sarcasm, *"Oh, you're just so helpful!"* In this sentence, the person refuses to help, shown as clear, direct, hidden negative intent.

Consider another example, "*Must be nice to leave early every day.*" In this sentence, the speaker sounds like a simple comment, but implies envy or criticism, giving rise to implicit sarcasm.

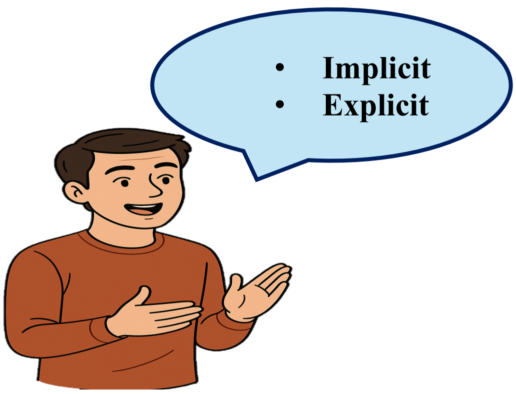


Fig. 5.3 Subtypes of surface-depth sarcasm

Understanding these types is essential for correctly detecting sarcastic tone in communication. Recognizing these types of sarcasm helps in building better sarcasm detection models. Ignoring them can lead to misinterpreted sarcasm, affecting dialogue understanding and intent recognition of text. *Fig. 5.3* shows the types of sarcasm as implicit and explicit nature.

5.3.1 Implicit sarcasm

In this type of sarcasm, the speaker still says something that seems ordinary, but the real meaning is hidden beneath [127]. This sarcasm doesn't come with strong vocal or visual clues. Instead, the listener must rely on context, background knowledge, or shared understanding to figure out the sarcastic intent.

For example, *"That's really helpful," said when someone makes things worse,* giving explicit sarcasm. There's no obvious sign of mockery, but the sarcasm is there, just tucked into the situation. Implicit sarcasm can be dry, and it's often missed unless the listener understands the speaker's intent.

5.3.2 Explicit sarcasm

This type of sarcasm happens when someone says something that sounds positive, but their tone of delivery clearly suggests they mean the opposite [130]. It usually includes exaggeration, mocking emphasis, or a noticeable change in expression that signals the listener not to take the words at face value. For example, *"Wow, that was brilliant!"* is said after someone makes a silly mistake.

Even without much context, it's clear to the speaker which is being sarcastic and doesn't actually think the action was smart. It relies heavily on cues like tone of voice, facial expression, or obvious exaggeration, especially in spoken language.

5.4 Masked Intentions Sarcasm

In this categorization, sarcasm occurs when the speaker's true intent is hidden, often using tone or context to signal the opposite meaning. The sarcasm lies not in the words themselves, but in the speaker's intended message. This type reveals how sarcasm depends on pragmatic understanding, not just what is said, but why it is said.

For example, *"Oh sure, take your time, we're only waiting on you."* Here, the words in the sentence sound polite, but the speaker's intention is to express annoyance, not patience. Within masked intentions, specific subtypes of sarcasm are presented as follows.

- Callousness
- Humour
- Offensiveness
- Humblebrag
- Oxymoron

Detecting the speaker's true intention is crucial for interpreting sarcasm accurately, especially in subtle or complex conversations. Without understanding the speaker's intent, sarcasm may be misunderstood as sincere. *Fig. 5.4* shows the subcategories for masked intention sarcasm.

Fig. 5.4 Subtypes of masked intentions sarcasm

Whereas, masked intentions sarcasm is more subtle and indirect. The speaker hides negative feelings behind neutral or polite words. The surface meaning seems normal or kind, but the underlying tone carries criticism or disapproval. This type focuses on concealed emotional intent, where sarcasm is delivered softly to avoid direct confrontation. For example, *"Thanks for your helpful advice."* Here, the speaker appears to show gratitude, but the real meaning is opposite implying the advice was unhelpful or unwanted.

In this type, sarcasm is context-dependent and often recognized through tone, situation, or prior conversation, rather than through explicit reversal of meaning.

5.4.1 Callousness

This type of sarcasm sounds emotionally detached or unsympathetic in nature. It's often used when someone mocks others feelings or situations in a cold or dismissive way [114]. The sarcastic tone adds to the harshness, making the comment sound more biting or uncaring. For example, *"Oh no, your phone battery died? What a tragedy!"* In this sentence, the speaker shows the maximization of someone's frustration in a mocking form.

5.4.2 Humour

This type of sarcasm is used playfully, with the main goal of making someone laugh or lightening the mood. Though the words may contradict the meaning, the intention is to amuse, not insult. For example, *"You're clearly a morning person!"* This sentence gives the presence of playful sarcasm.

5.4.3 Offensiveness

This type of sarcasm is sharper and often carries a personal insult. It's used to indirectly criticize, ridicule, or express anger in a more veiled way [115]. The message is usually negative, even if wrapped in a seemingly polite phrase. For example, *"Thanks for all your help, oh wait, you did nothing!"* This sentence conveyed a sarcastic tone, disguised as gratitude.

5.4.4 Humblebrag

Humblebrag sarcasm involves pretending to be modest while actually showing off. It often masks self-praise under a fake complaint or exaggerated humility. For example, *"Ugh, I have to wake up early for my award ceremony again"*. In this sentence, the speaker gives a complaint, but in actuality, he is bragging about the scenario.

5.4.5 Oxymoron

This type of sarcasm uses contradictory words or phrases to emphasize irony. It creates a sarcastic effect by placing opposing ideas side by side for humorous or critical impact. For example, *"What a beautifully disastrous presentation!"* In this sentence, the speaker combines praise with criticism in a single phrase as "*beautifully disastrous*".

5.5 Proposed Methodology for Meaning Inversion Sarcasm

After defining sarcasm and its various types in the previous section, we now introduce the methodologies for these categorizations and detection. In this section, we propose a method that focuses on meaning inversion sarcasm using the parabolic fractal optimization technique. We categorise it into four subtypes: hyperbole (HPS), rhetorical devices (RDS), like-prefixed (LPS), and dropped negation (DNS) sarcasm.

This method helps to capture the subtle variations in sarcastic expression. By doing so, the method becomes more sensitive to context, tone, and structure. We detect these types of sarcasm in the SOBH-REB method (explained in *Chapter 3*) in English text. Initially, we separate all sarcastic sentences derived from the proposed (SOBH-REB) method, each accompanied by its optimal sarcasm score (ψ_1). We divide the types of sarcasm into two sets. The first set $\mathrm{Z1} = \{\mathrm{DNS}, \mathrm{RDS}\}$ and the second set $\mathrm{Z2} = \{\mathrm{LPS}, \mathrm{HPS}\}$.

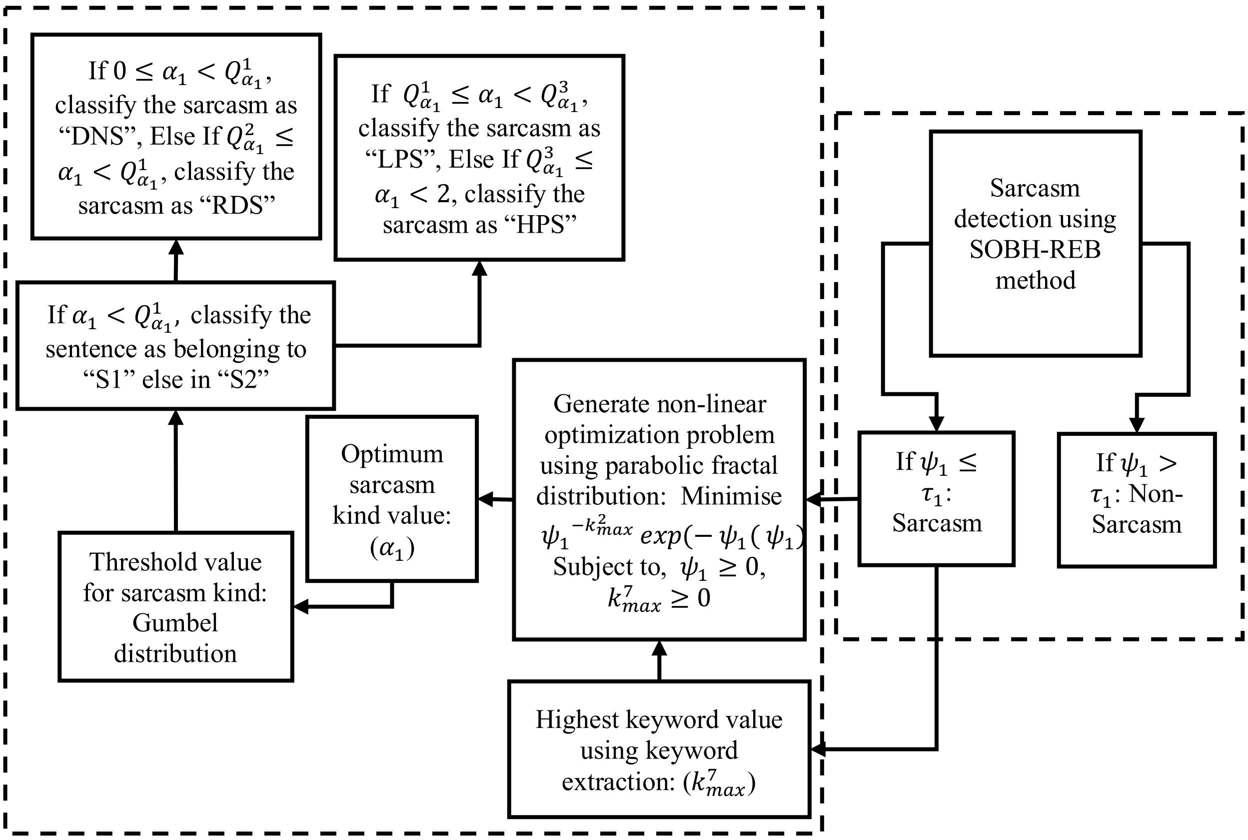


Fig. 5.5 Flowchart of SOBH-REB method for categorization of types of sarcasm as hyperbole, like-prefixed, dropped negation and rhetorical devices

We first employed the RAKE model to extract the highest keyword value (k^7_{max}). Then, calculate the optimum sarcasm kind value (α_1) by using the minimised parabolic fractal distribution function. This objective function is subject to two constraints, i.e., ψ_1 and k^7_{max}.

Eqn. (5.1) defines the minimised non-linear optimisation problem with parabolic fractal distribution.

$$\left.\begin{array}{r}\alpha_1 = \text{ Minimise } \psi_1{}^{-k_{max}^7} \exp(-\psi_1 \,(\log \psi_1)^2) \\ Subject\ to, \psi_1 \geq 0\ and\ k_{max}^7 \geq 0\end{array}\right\} \tag{5.1}$$

The main task of this function is to minimise the defined optimisation problem to get the α_1. *Fig. 5.5* shows the flowchart of the proposed method, SOBH-REB, for the classification of types of sarcasm.

Then α_1 is compared with the defined threshold values for types of sarcasm-1 ($Q_{\alpha_1}^1$), sarcasm-2 ($Q_{\alpha_1}^2$) and sarcasm-3 ($Q_{\alpha_1}^3$). The Gumbel distribution [175] has been utilised to calculate the threshold value for sarcasm kinds shown in *Tables 5.1, 5.2*, and *5.3*. *Eqn. (5.2)* demonstrated the formula of the Gumbel distribution used for the calculation of $Q_{\alpha_1}^1$, $Q_{\alpha_1}^2$ and $Q_{\alpha_1}^3$.

$$Q_{\alpha_1}\left(mean_{k_{max}^7}; mean_{\alpha_1}, \sigma\right) = e^{-e^{\frac{-(mean_{k_{max}^7} - mean_{\alpha_1})}{\sigma}}} \tag{5.2}$$

Table 5.1 Numeric score computation of the threshold value $Q_{\alpha_1}^1$

Parameter	Values
Threshold value for types of sarcasm-1 ($Q_{\alpha_1}^1$)	*Dataset -1 [149]:* $Q_{\alpha_1}^1 = e^{-e^{\frac{-(9.9910-0.9204)}{0.5093}}} = 0.9999 \approx 1$ *Dataset -2 [150]:* $Q_{\alpha_1}^1 = e^{-e^{\frac{-(11.345-1.245)}{0.7346}}} = 0.9765 \approx 1$ *Dataset -3 [151]:* $Q_{\alpha_1}^1 = e^{-e^{\frac{-(6.346-0.6382)}{0.2801}}} = 0.9893 \approx 1$

Table 5.2 Numeric score computation of threshold value $Q_{\alpha_1}^2$

Parameter	Values
Threshold value for types of sarcasm-2 ($Q_{\alpha_1}^2$)	*Dataset -1 [149]:* $Q_{\alpha_1}^2 = e^{-e^{\frac{-(5.044-4.44)}{0.975}}} = 0.5437 \approx 0.5$ *Dataset -2 [150]:* $Q_{\alpha_1}^2 = e^{-e^{\frac{-(8.648-3.568)}{0.7756}}} = 0.5671 \approx 0.5$ *Dataset -3 [151]:* $Q_{\alpha_1}^2 = e^{-e^{\frac{-(5.678-3.967)}{0.8870}}} = 0.5501 \approx 0.5$

Table 5.3 Numeric score computation of threshold value $Q_{\alpha_1}^3$

Parameter	Values
Threshold value for types of sarcasm-3 ($Q_{\alpha_1}^3$)	*Dataset -1 [149]:* $Q_{\alpha_1}^3 = e^{-e^{\frac{-(15.78-0.6770)}{0.6678}}} = 1.5572 \approx 1.5$ *Dataset -2 [150]:* $Q_{\alpha_1}^3 = e^{-e^{\frac{-(17.94-0.9750)}{0.1245}}} = 1.6013 \approx 1.5$ *Dataset -3 [151]:* $Q_{\alpha_1}^3 = e^{-e^{\frac{-(14.23-0.954)}{0.3345}}} = 1.4934 \approx 1.5$

ALGORITHM 5.1: Classification of sarcastic sentences in terms of meaning inversion types of sarcasm
Input: $\boldsymbol{\psi_1}$, $\boldsymbol{k_{max}^7}$, $\boldsymbol{Q_{\alpha_1}^1}$, $\boldsymbol{Q_{\alpha_1}^2}$ *and* $\boldsymbol{Q_{\alpha_1}^3}$ *Output:* $\boldsymbol{\alpha_1}$ *(optimum sarcasm kind value), sarcasm kinds, i.e., HPS, RDS, LPS, and DNS*   *1. Define two pre-defined sets Z1 and Z2 as Z1=* $\{\boldsymbol{DNS}, \boldsymbol{RDS}\}$ *and Z2 =* $\{\boldsymbol{LPS}, \boldsymbol{HPS}\}$ *2. Generate the non-linear optimization problem consisting of parabolic fractal distribution:* $\boldsymbol{\alpha_1} =$ *Minimise* $\boldsymbol{\psi_1}^{-\boldsymbol{k_{max}^7}} \boldsymbol{exp}(-\boldsymbol{\psi_1} (\boldsymbol{log}\, \boldsymbol{\psi_1})^2)$ *Subject to,* $\boldsymbol{\psi_1} \geq \boldsymbol{0}$, *and* $\boldsymbol{k_{max}^7} \geq \boldsymbol{0}$ *3. Print the output → "optimum sarcasm kind value (*$\boldsymbol{\alpha_1}$*)"*   *If* $\boldsymbol{\alpha_1} < \boldsymbol{Q_{\alpha_1}^1}$, *the sentence contains sarcasm from set Z1, else from set Z2*   *If* $\boldsymbol{0} \leq \boldsymbol{\alpha_1} < \boldsymbol{Q_{\alpha_1}^2}$, *classify the sarcasm as "DNS."*   *Else If* $\boldsymbol{Q_{\alpha_1}^2} \leq \boldsymbol{\alpha_1} < \boldsymbol{Q_{\alpha_1}^1}$, *classify the sarcasm as "RDS".*   *If* $\boldsymbol{Q_{\alpha_1}^1} \leq \boldsymbol{\alpha_1} < \boldsymbol{Q_{\alpha_1}^3}$, *classify the sarcasm as "LPS."*   *Else, If* $\boldsymbol{Q_{\alpha_1}^3} \leq \boldsymbol{\alpha_1} \leq \boldsymbol{2}$, *classify the sarcasm as "HPS."*

The $mean_{\alpha_1}$ is defined as the mean value of α_1 shown in *Eqn. (5.3)* and σ as the standard deviation of α_1, where $mean_{\alpha_1}$taken as the sample mean of all ψ_1 shown in *Eqn. (5.4)*.

$$mean_{\alpha_1} = \frac{\sum_{i=1}^{i=m} \alpha_{1_i}}{m} \tag{5.3}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{i=h} \left(\alpha_{1_i} - mean_{\alpha_1}\right)^2}{m}} \text{ Where, } sample_{mean} = \frac{\sum_{i=1}^{i=m} \psi_{1_i}}{m} \tag{5.4}$$

$mean_{k_{max}^7}$ denotes the mean value of all k_{max}^7 shown in *Eqn. (5.5)* where i is from 1 to m (total number of sentences).

$$mean_{k^7_{max}} = \frac{\sum_{i=1}^{i=m} \alpha_{1_i}}{m} \tag{5.5}$$

First, we compare the value of α_1 with the $Q^1_{\alpha_1}$. This helps to select between the two sets of sarcasm. If $\alpha_1 \geq Q^1_{\alpha_1}$, then the sentence belongs to $Z_2 = \{LPS, HPS\}$ and if $\alpha_1 < Q^1_{\alpha_1}$, then the sentence belongs to $Z_1 = \{DNS, RDS\}$. If $Q^2_{\alpha_1} \leq \alpha_1 < Q^1_{\alpha_1}$, then we classify the sarcasm as "RDS," and for $0 \leq \alpha_1 < Q^2_{\alpha_1}$, It is classified as "DNS".

Similarly, for the set Z_2, if $Q^1_{\alpha_1} \leq \alpha_1 < Q^3_{\alpha_1}$, then the sentence is classified as "LPS," and $Q^3_{\alpha_1} \leq \alpha_1 \leq 2$, we tag it as "HPS." *Algorithm 5.1* describes the classification of sarcastic comments into different categories.

5.6 Proposed Methodology for Surface-Depth Sarcasm

This section presents the methodology for surface-depth sarcasm using the optimisation technique. This type shows two categories, i.e., implicit or explicit sarcasm, that reflect how it is delivered either through obvious contradictions or context-driven hints.

We introduce a classification framework based on the LDL-SDTE method, guided by a mathematical optimisation technique. This method helps to capture strong verbal cues in explicit sarcasm and trace hidden linguistic patterns in implicit cases. By learning linguistic, tonal, and contextual features through optimisation, the proposed method adapts to give various sarcastic expressions.

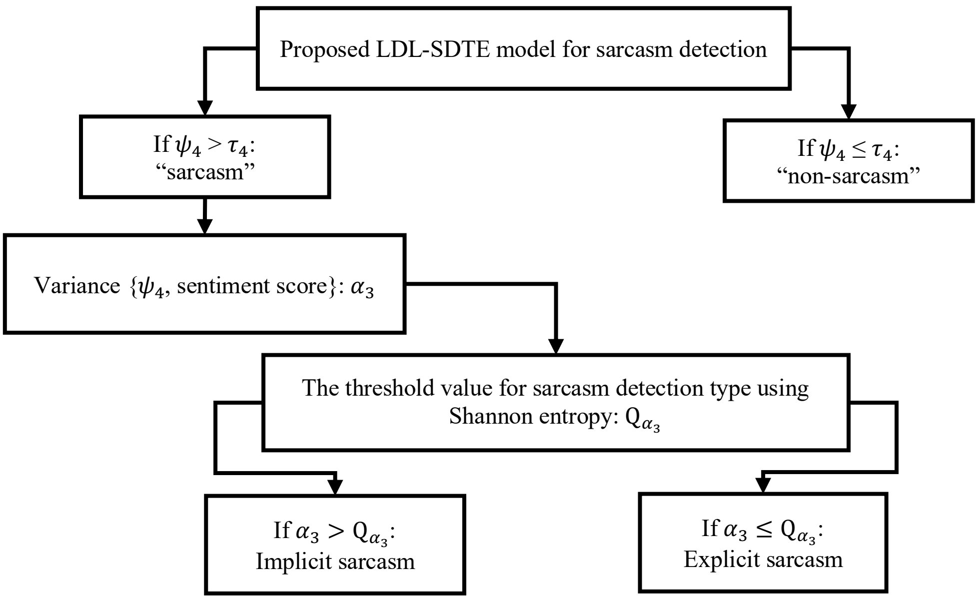


Fig. 5.6 Flowchart of the LDL-SDTE method for categorization of types of sarcasm as implicit or explicit in nature

First, we isolate all sarcastic sentences produced by LDL-SDTE in sarcasm detection and pair them with their respective optimal sarcasm scores (ψ_4). Applying the statistical measure, i.e., calculating the variance of the ψ_4 and an average of positive and negative sentiments from SentiWordNet for each sentence gives the optimal sarcasm kind value α_3. The following steps are followed in the proposed method, as shown in *Fig. 5.6.* The comparison between α_3, and the threshold value for types of sarcasm Q_{α_3} help in deduction for the type of sarcasm.

$$Q_{\alpha_3} = -\left(\sum_i \left(p(\psi_{4_i}) \log\left(p(\psi_{4_i})\right)\right)\right) = 0.7 \tag{5.6}$$

The Q_{α_3} is calculated using Shannon entropy as depicted in *Eqn. (5.6)*. The $p(\psi_{4_i})$ is the conditional probability of the variance of the final score. Based on the defined comparison, we deduce that if $\alpha_3 > Q_{\alpha_3} \rightarrow$ *implicit* sarcasm else, it is *explicit* sarcasm.

5.7 Proposed Methodologies for Masked Intention Sarcasm

In this part of the thesis, we shift focus to sarcasm that is driven by the speaker's underlying intention, rather than just the surface-level contradiction of words. This category, referred to as meaning intentions, influenced sarcasm. It captures the sarcastic tone, which is shaped by social emotions, intent, or interpersonal context directly. We develop optimization-based methods that interpret sarcasm by analysing its intent in this categorization.

We classify such sarcasm into subtypes: callous and non-callous, humorous and non-humorous, offensive and non-offensive, and humblebrag and oxymoron. These groupings help decode the intended prosodic layers beneath the surface of the text. The proposed methods applied optimization techniques to learn features like tone, sentiment polarity, and contextual cues. With the help of learning these types, one can further identify whether a sarcastic comment is intended to mock, amuse, downplay, or subtly boast.

The following subsections detail the behaviour of each subtype along with its detection methodology.

5.7.1 Proposed Method for Callous and Non-Callous

In this subsection, we use the Gauss-Kuzmin distribution-based optimization technique in the proposed SDO-BCRT method (explained in *Chapter 3*) to detect sarcastic sentences as either callous or non-callous. *Fig. 5.7* shows the SDO-BCRT method for categorization of types of sarcasm in terms of callous and non-callous.

First, we extract the sarcastic sentences identified by the SDO-BCRT and assign each corresponding optimal sarcasm score (ψ_2). Afterwards, we formulate the non-linear optimization problem using the Gauss-Kuzmin distribution for this categorization depiction. The Gauss-Kuzmin distribution is taken as the maximized objective function in the formulated optimization problem. This function is subject to the obtained constraints, i.e., $\mathrm{k}_{\max}^{1}$, ψ_2, r, and ϕ_2. The formulated optimization problem helps in detecting the presence of the inherent intent nature of sarcasm, as shown in *Eqn. (5.7)*.

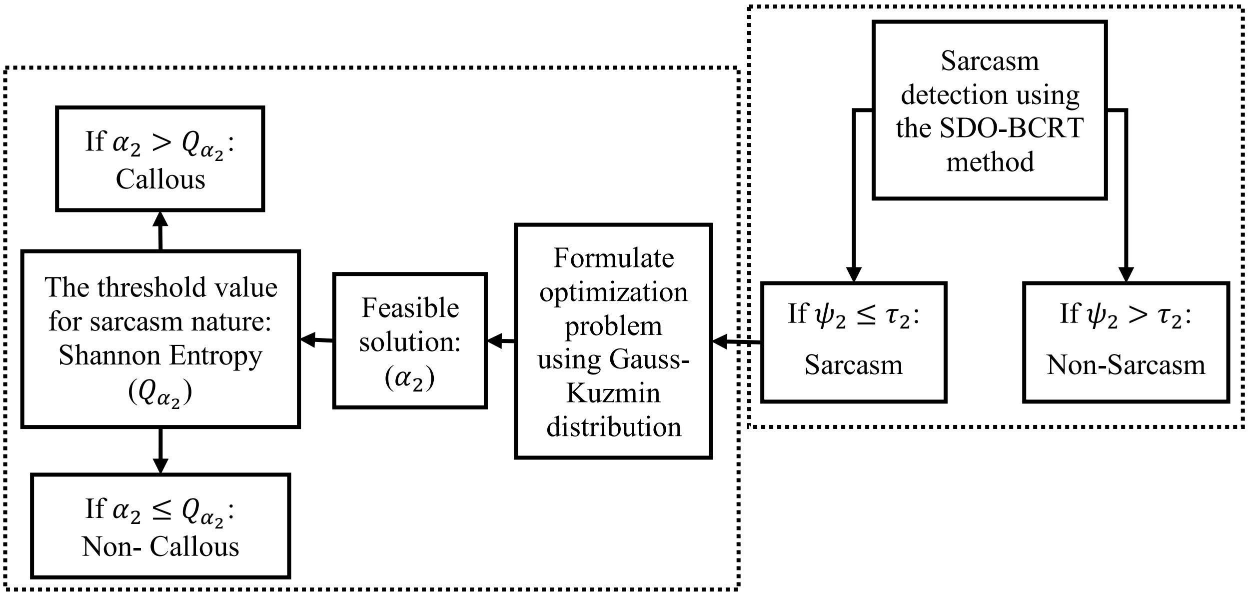


Fig. 5.7 Flowchart of SDO-BCRT method for categorization of types of sarcasm as callous and non-callous

$$\left.\begin{aligned} \alpha_2 &= Maximise\ \log_{10}\left(1+\frac{1}{\psi_2}\right) \\ Subject\ to, \psi_2 &= r\left(\frac{\left|\phi_2\ \mathrm{k}_{\max-1}^{1}\right|}{\mathrm{K}}\right) \ge 0, \text{if } \phi_2\ \neq 0 \ge 0 \\ &= \log(\phi_2\ ) \ge 0,\ \text{if } \phi_2\ = 0 \ge 0 \\ \phi_2 &= \frac{1}{\pi r\left[1+\left(\frac{(PSS-NSS)^2}{2}\right)\right]} \ge 0 \\ r &= \frac{(PSS+NSS)}{2} \ge 0 \end{aligned}\right\} \tag{5.7}$$

The final computed value obtained from the defined optimization problem is termed as optimum sarcasm kind value (α_2). The α_2 is then compared with the defined threshold value for types of sarcasm (Q_{α_2}). The Q_{α_2} is calculated using the Shannon entropy [176] as shown in *Eqn. (5.8)*.

$$Q_{\alpha_2} = -\sum_{i=1}^{n} P(\alpha_{2_i}) \log_2 \left(P(\alpha_{2_i})\right) \tag{5.8}$$

The $P(\alpha_2)$ showed the probability-taking value on the optimal sarcasm score ε. *Eqn. (5.8)* gives the formula for calculating $Q_{\alpha_2} = 0.5$, as shown in *Table 5.4*.

Table 5.4 Threshold computation using Shannon entropy

Parameters	**Values**
Threshold value of sarcasm nature $(\boldsymbol{Q_{\alpha_2}})$	*(Dataset-1 [149]):* $Q_{\alpha_2} = -\sum_{i=1}^{12506} P(\alpha_{2_i})\, log_2 \left(P(\alpha_{2_i})\right) = 0.501 \approx 0.5$ *(Dataset-2 [150]):* $Q_{\alpha_2} = -\sum_{i=1}^{25378} P(\alpha_{2_i})\, log_2 \left(P(\alpha_{2_i})\right) = 0.556 \approx 0.5$ *(Dataset-3 [151]):* $Q_{\alpha_2} = -\sum_{i=1}^{11857} P(\alpha_{2_i})\, log_2 \left(P(\alpha_{2_i})\right) = 0.540 \approx 0.5$ *Dataset-4 [152]):* $Q_{\alpha_2} = -\sum_{i=1}^{11025} P(\alpha_{2_i})\, log_2 \left(P(\alpha_{2_i})\right) = 0.550 \approx 0.5$

ALGORITHM 5.2: Sarcasm inhibits intent nature classification viz non-linear optimization problem using Gauss Kuzmin distribution

Input: $\boldsymbol{S_{sarcasm}}$, $\boldsymbol{\psi_2}$, $\boldsymbol{\eta}$, $\boldsymbol{\phi_2}$, $\mathbf{k}^{\mathbf{1}}_{\mathbf{max}}$ *and* $\boldsymbol{r}$

Output: callous or non-callous classification based on $\boldsymbol{\alpha_2}$

1. *For each sentence, S repeats steps 2 to 5.*
2. *Formulate the non-linear optimization problem using the Gauss- Kuzmin distribution:* $\boldsymbol{\alpha_2}$ = *Maximise* $\mathbf{log_{10}}\left(\mathbf{1} + \frac{\mathbf{1}}{\boldsymbol{\psi_2}}\right)$

subject to, $\boldsymbol{\psi_2} = \begin{cases} \boldsymbol{r}\left(\frac{\left|\boldsymbol{\phi_2}\, \mathbf{k}^{\mathbf{3}}_{\mathbf{max}} - \mathbf{1}\right|}{\mathrm{K}}\right) \geq \mathbf{0}, \text{ if } \boldsymbol{\phi_2} \neq \mathbf{0} \\ \mathbf{log}(\boldsymbol{\phi_2}) \geq \mathbf{0}, \text{ if } \boldsymbol{\phi_2} = \mathbf{0} \end{cases} \geq \mathbf{0}$ *and* $\boldsymbol{\phi_2} = \frac{\mathbf{1}}{\boldsymbol{\pi r}\left[\mathbf{1} + \left(\frac{(\boldsymbol{PSS} - \boldsymbol{NSS})^{\mathbf{2}}}{\mathbf{2}}\right)\right]} \geq \mathbf{0}$ *and* $\boldsymbol{r} = \frac{(\boldsymbol{PSS} + \boldsymbol{NSS})}{\mathbf{2}} \geq \mathbf{0}$

3. *Output → "feasible solution (*$\boldsymbol{\alpha_2}$*)"*
4. *If* $\boldsymbol{\alpha_2} > \boldsymbol{Q_{\alpha_2}}$ *→ "callous"*
5. *Else "non-callous"*

The classification is done based on comparison between α_2 and Q_{α_2}. If $\alpha_2 > Q_{\alpha_2}$, we classify the context as "*callous*," else it would be classified as "*non-callous*." This classification provides information about the sarcastic nature of a sentence as either misleading or insensitive in the public domain. *Algorithm 5.2* defines the depiction of the inherent intent of sarcastic context as either callous or not.

5.7.2 Proposed Method for Humblebrag and Oxymoron

This subsection addresses sarcasm expressed through humblebrag and oxymoron constructions, both of which mask intent behind contrasting meanings. To detect these forms of sarcasm, we apply the Logistic distribution as a minimized function in SD-GKOT to model the subtle polarity flips. The function captures the gradual shift in sentiment strength, often present in masked or dual-toned phrases. *Fig. 5.8* gives the classification for types of sarcasm that appear in the flowchart of the proposed SD-GKOT method.

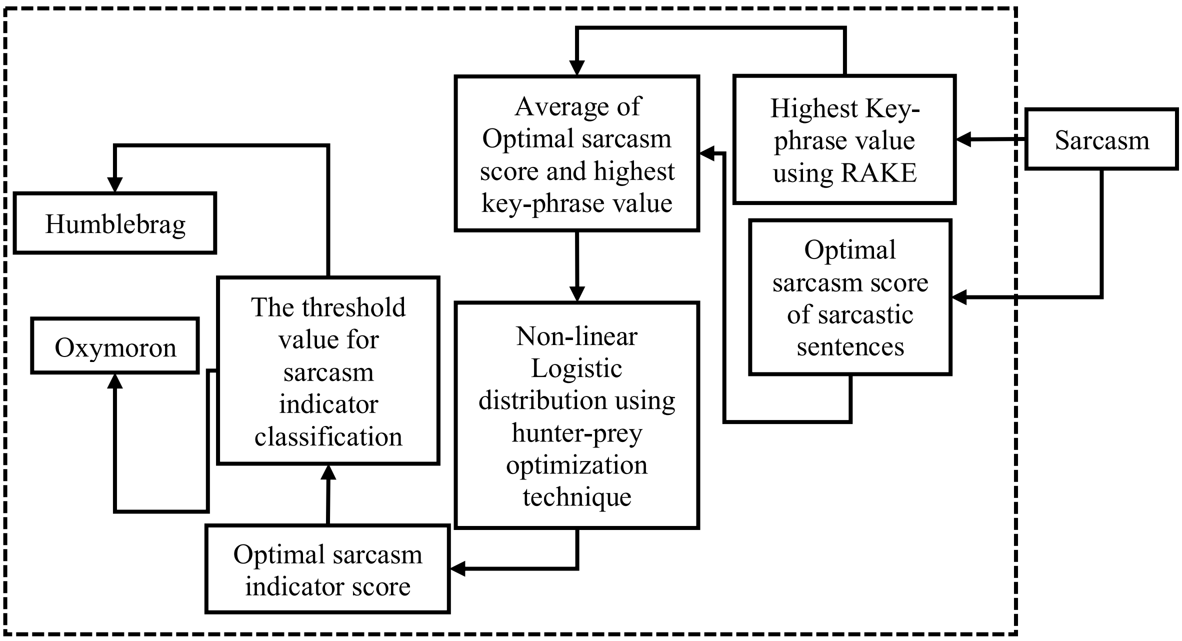


Fig. 5.8 Flowchart of SD-GKOT for categorization of types of sarcasm as Humblebrag and Oxymoron

The hunter-prey optimization technique (HPO) is used to solve this minimization problem effectively. It mimics predator-prey dynamics to iteratively search for optimal sarcasm indicators. It adjusts the feature weights for contrast, phrase contradiction, and indirect sentiment indicators. By minimizing the Logistic distribution through HPO, the method identifies phrases where the literal and implied meanings diverge subtly.

This method enhances the system's ability to handle complex sarcasm patterns that standard classifiers may overlook due to hidden intent or reversed meaning structures. First, we segregate all the sarcastic sentences obtained from the SD-GKOT methodology with their obtained optimal sarcasm score (ψ_7). Afterwards, key-phrase extraction is performed on all sarcastic sentences to acquire the highest key-phrase value (k_{max}^4) using the RAKE model. Then, apply the logistic distribution-based hunter-prey optimization technique [177].

The formulated optimization technique is subject to an average of ψ_7 and k_{max}^4, denoted as β, respectively. The choice of taking the logistic distribution as a minimized objective function allows for efficient optimization searching for the minimum solution with the best convergence. *Eqn. (5.9)* shows the optimization technique using the logistic distribution as the objective function. This function helps to determine the psychological features and arrangement of sarcastic gestures with affective traits.

$$\left.\begin{aligned} \alpha_4 = Minimize \left|\gamma - \frac{e^{-\beta}}{\beta\left(1+e^{-\beta}\right)^2}\right|^2 \\ Subject\ to, \beta = \frac{\psi_7 + k_{max}^4}{2} \geq 0 \end{aligned}\right\} \tag{5.9}$$

The aim of the minimized logistic distribution function using the hunter-prey optimization technique is to get the optimum sarcasm kind value (α_4). The value of γ in *Eqn. (5.9)* is obtained using the hunter-prey optimization technique. It is a bio-inspired optimization that simulates the hunting behaviour observed while chasing prey. The algorithm provides the interactions between hunters and prey, where hunters adapt their strategies to capture prey, and prey attempt to escape from the hunters.

This interaction provides a powerful framework for exploring and exploiting the search space to find the optimal sarcasm indicator score from the defined objective function. First, the initialization of the population of hunters and prey is randomly done within the search space. Then, simultaneously define the maximum and minimum speed range of both hunters and prey. The SD-GKOT method has taken the search space within the bounds, i.e., $\gamma \in [\gamma_{min} = -10, \gamma_{max} = 10]$ with a maximum of one hundred iterations.

$$\gamma_i = rand(1, V) \times (\gamma_{max} - \gamma_{min}) + \gamma_{min} \tag{5.10}$$

The initial position of the hunter is calculated using *Eqn. (5.10)*. The initial position of the hunter i is represented by γ_i, while $rand$ is a random value within the range of the dimension

V, which indicates the number of variables in the problem. The following step calculates the position γ_i for the hunter's search process, as described in *Eqn. (5.11)*.

$$\gamma_i(z+1) = \gamma_i(z) + 0.5\left[\left(2\delta n B_{pos} - \gamma_i(z)\right) + \left(2(1-\delta)n\theta - \gamma_i(z)\right)\right] \tag{5.11}$$

Eqn. (5.11) provides the updated rules for the hunter's positions, where $\gamma(z)$ represents the current position of the hunter, $\gamma(z+1)$ which is the updated or next position, B_{pos} denotes the prey's position, θ represents the average of all hunters positions, and n is an adaptive parameter determined using *Eqn. (5.12)*.

$$\left.\begin{array}{c} B = R_1 < \delta; \partial = (B == 0); \\ n = R_2 \times \partial + R_3 \times (\sim\partial) \end{array}\right\} \tag{5.12}$$

The vectors R_1, R_2, and R_3 are randomly generated within the interval $[0,1]$, while B is a binary vector containing values of 0 or 1, corresponding to the number of variables in the problem. The index ∂ refers to the positions in R_1 where the condition $B == 0$ holds. It is important to note that these values are randomized and reshuffled with each iteration. Additionally, δ is a balancing parameter that decreases progressively from 1 to 0.02 in each iteration, and its value is determined using *Eqn. (5.13)*.

$$\delta = 1 - \varsigma\left(\frac{0.98}{max\varsigma}\right) \tag{5.13}$$

Here, ς denotes the current iteration, while maxς represents the maximum number of iterations considered in the study. To determine the prey's position, B_{pos} , it is first necessary to compute the average of all hunters' positions (θ) using *Eqn. (5.14)*. The distance of each hunter from this average position, $\varepsilon_{(i)}$, is then calculated using the Euclidean distance shown in *Eqn. (5.15)*.

$$\theta = \frac{1}{n}\sum_{i=1}^{n}\gamma_i \tag{5.14}$$

$$\varepsilon_{(i)} = \left(\sum_{i=1}^{V}(\gamma_i - \theta_i)^2\right)^{1/2} \tag{5.15}$$

The search agent farthest from the average of all positions is identified as the prey (B_{pos}). At this stage, the prey moves away from the hunters in an attempt to evade capture. The prey's updated position is described by *Eqn. (5.16)*.

$$\left.\begin{array}{c} B_{pos} = \gamma_i | i \text{ is sorted by} \max value\ \varepsilon_i \\ B_{pos} = G_{pos} + \delta Z cos(2\pi r_4) \times \left(G_{pos} - \gamma_i(z)\right) \end{array}\right\} \tag{5.16}$$

In this context, r_4 is a random number within the range $[-1,1]$, and G_{pos} represents the global position of the agent. The hunter's position is updated to the most favorable position for pursuing the prey using the γ_i values. This updated position is then evaluated using the fitness function $f(\gamma_i)$, which corresponds to the i^{th} objective function. The best solution is stored at each iteration based on the fitness function, aiming to minimize the score.

In the final step, the positions of both hunters and prey are evaluated. The updated positions, with the best-known fitness value, are input into the fitness function. The hunter or prey with the lowest score, representing the best fitness, is considered the optimal solution γ. From the hunter-prey optimization, the value of γ is computed in the defined objective function, and finally, the value of the optimum sarcasm kind value (α_4) is calculated.

Afterward, the value of α_4 is compared with the calculated threshold value for types of sarcasm (Q_{α_4}) computed as 4.86. This value is obtained from Shannon entropy shown in *Eqn. (5.17)*. *Table 5.5* demonstrates the threshold value scales used in the proposed method, SD-GKOT. Where Q_{α_4} is the Shannon entropy, $p\left(\alpha_{4_i}\right)$ is the probability distribution of α_4, and $\mu_{\alpha_{4_i}}$ represents the average of optimal sarcasm scores.

$$Q_{\alpha_4} = -\frac{\sum_{i=1}^{i=m} p(\alpha_{4_i}) \log p(\alpha_{4_i})}{\sum_{i=1}^{i=m} \mu_{\alpha_{4_i}}} \tag{5.17}$$

Table 5.5 Parameters used in the proposed method

Parameters	**Values**
The threshold value for sarcasm indicator classification ($\boldsymbol{Q_{\alpha_4}}$)	*Dataset 1 [149]:* $Q_{\alpha_4} = -\left(-\frac{592.259}{121.55}\right) = 4.8721 \approx 4.86$ *Dataset 2 [150]:* $Q_{\alpha_4} = -\left(-\frac{707.19}{145.50}\right) = 4.8604 \approx 4.86$ *Dataset 3 [151]:* $Q_{\alpha_4} = -\left(-\frac{636.478}{131.37}\right) = 4.8446 \approx 4.86$

The humblebrag and oxymoron labeling is done based on a comparison between α_4 and Q_{α_4}. If $\alpha_4 \leq Q_{\alpha_4}$, the sentence is tagged as an "*oxymoron*," and if $\alpha_4 > Q_{\alpha_4}$, then the sentence is tagged as "*humblebrag*." *Algorithm 5.3* conveys the interpretation of sarcasm indicators classification in terms of humblebrag and oxymoron.

ALGORITHM 5.3: Classification of affective indicators of sarcasm using non-linear logistic hunter-prey optimization technique
Input: $\boldsymbol{S_{sarc}}$, $\boldsymbol{k^4_{max}}$, $\boldsymbol{\psi_7}$, *maximum number of iterations (max*$\boldsymbol{\varsigma}$*),* $\boldsymbol{Q_{\alpha_4}}$ *Output: Classification of sarcasm indicators as humblebrag and oxymoron based on optimal sarcasm tonal indicator score (*$\boldsymbol{\alpha_4}$*)* *1. For each sentence,* $\boldsymbol{S_{sarc}}$ *repeats steps from 2 to 6.* *2. Calculate* $\boldsymbol{k^4_{max}}$ *using the RAKE model.* *3. Compute* $\boldsymbol{\alpha_4}$ *by formulating a non-linear logistic distribution using the hunter-prey optimization technique:* $\boldsymbol{\alpha_4} =$ *Minimize* $\left\lvert \boldsymbol{\gamma} - \frac{e^{-\beta}}{\beta(1+e^{-\beta})^2} \right\rvert^2 = \boldsymbol{f(\gamma)} =$ *objective function* *Subject to* $\boldsymbol{\beta} = \frac{\psi_7 + k^4_{max}}{2} \geq \boldsymbol{0}$ *4. Initialize the hunter's and prey's position randomly within bounds* $[\boldsymbol{\gamma_{min}}, \boldsymbol{\gamma_{max}}]$ *5. For iteration 1 to max*$\boldsymbol{\varsigma}$*:* *For each hunter move towards prey, calculate its position within bounds:* $\boldsymbol{\gamma_i(z+1)} = \boldsymbol{\gamma_i(z)} + \boldsymbol{0.5}\left[\left(\boldsymbol{2\delta n B_{pos}} - \boldsymbol{\gamma_i(z)}\right) + \left(\boldsymbol{2(1-\delta)n\theta} - \boldsymbol{\gamma_i(z)}\right)\right]$ *For each prey move away from the hunter, calculate its position within bounds:* $\boldsymbol{B_{pos}} = \boldsymbol{G_{pos}} + \boldsymbol{\delta Z cos(2\pi R_4)} \times \left(\boldsymbol{G_{pos}} - \boldsymbol{\gamma_i(z)}\right)$ *6. After max*$\boldsymbol{\varsigma}$ *iterations, calculate the optimal objective function value by evaluating the fitness function:*$\boldsymbol{f(\gamma_i)}$ *Evaluate* $\boldsymbol{G_{pos}}$ *7. The best prey's position with minimized fitness value gives the optimal solution "*$\boldsymbol{\gamma}$*".* *8. Decision-based on comparison between* $\boldsymbol{\alpha_4}$ *and* $\boldsymbol{Q_{\alpha_4}}$*:* *If* $\boldsymbol{\alpha_4} > \boldsymbol{Q_{\alpha_4}}$*, classify the sentence as "humblebrag."* *Otherwise, classify the sentence as an "oxymoron."* *9. End*

5.7.3 Proposed Methods for Offensive and Non-Offensive

In this subsection, we target sarcasm that varies by degree of offensiveness in its intent and tone. Offensive sarcasm is often sharp, insulting, and masked under indirect wording. To model this difference, we employ the Poisson distribution as a minimized likelihood function in the UCRT-CGKO method. It helps in detecting sudden spikes in sentiment intensity with sparse textual patterns. Offensiveness often appears as low-frequency yet high-impact phrases, which this distribution captures effectively.

The optimization process adjusts weights for lexical intensity, target relevance, and aggression cues. The objective is to minimize the Poisson-based function, thereby identifying sarcastic expressions with offensive connotations. In addition, we propose a second method using the Gumbel distribution in SDWT-ELM for depicting types of sarcasm. The Gumbel distribution optimizes sarcastic sentences, focusing on high polarity shifts, hyperbolic tone, or sharp targeting. This optimization helps in accurate classification even when the offensive sarcasm is veiled or context-dependent. The flowchart of the UCRT-CGKO method classifies the types of sarcasm shown in *Fig. 5.9*.

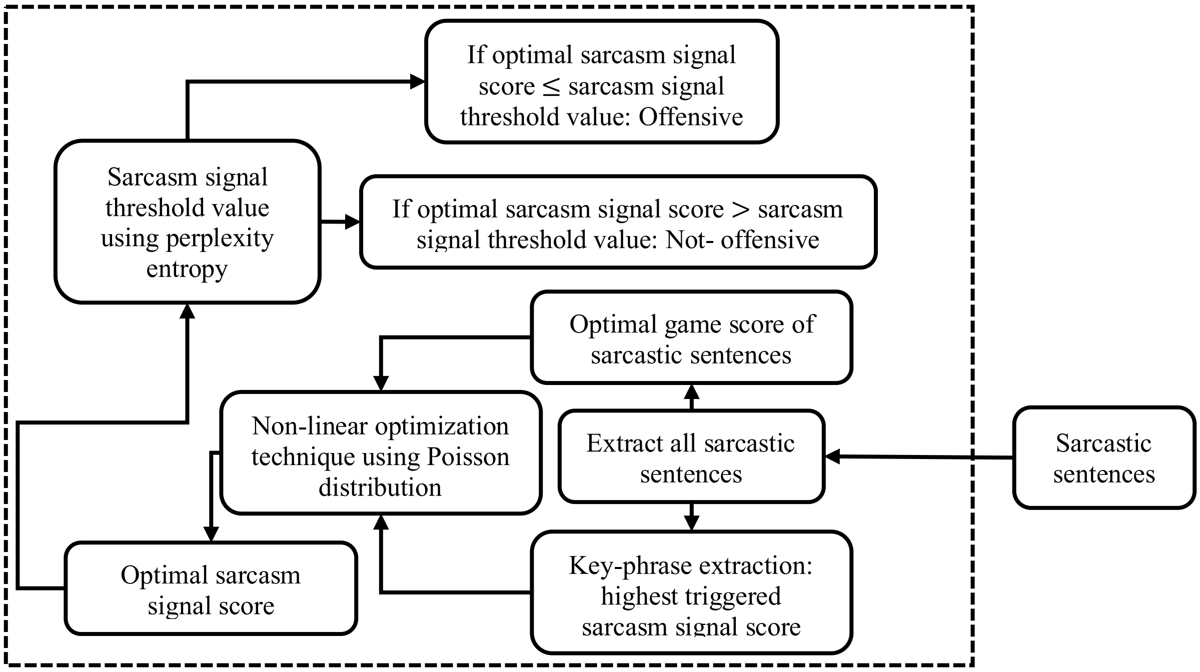


Fig. 5.9 Flowchart of UCRT-CGKO for categorization of types of sarcasm as offensive and non-offensive

First, we categorize types of sarcasm in the UCRT-CGKO method. Initially, the sarcastic sentences obtained from the UCRT-CGKO method are extracted with their optimal sarcasm score (ψ_8). The k_{max}^5 and sarcastic sentences with their ψ_8 are further interjected into the optimization technique using the Poisson distribution as a minimised objective function.

Utilization of this distribution as an objective function potentially incorporates the complex traits retrieved from the dataset. The *Eqn. (5.18)* gives the formulation of a non-linear optimization technique using the Poisson distribution.

$$\alpha_5 = \left| \text{Minimise } \frac{e^{k_{max}^5 \times k_{max}^5}{}^{-\left(\frac{\psi_8}{10}\right)}}{\left(\frac{\psi_8}{10}\right)!} \right|, \text{ Subject to: } k_{max}^5 \geq 0 \text{ and } \psi_8 \geq 0 \quad (5.18)$$

The optimal value obtained from this optimization technique is termed the optimum sarcasm kind value (α_5). The α_5 is then compared with the computed threshold value for types of sarcasm (Q_{α_5}). It is calculated as 0.577 using the perplexity entropy structure shown in *Eqn. (5.19)*.

$$Q_{\alpha_5} = 2^{\sum_{i=1}^{m} p(\alpha_{5_i}) \log_2(\alpha_{5_i})} \quad (5.19)$$

The sarcasm indicator classification is done based on the comparison between α_5 and Q_{α_5}. If $\alpha_5 \leq Q_{\alpha_5}$, then the sentence is tagged as "*offensive*," else it is tagged as "*non-offensive*." *Algorithm 5.4* defines the procedure of depicting the implicit intent of sarcasm signals in terms of offensive and non-offensive nature.

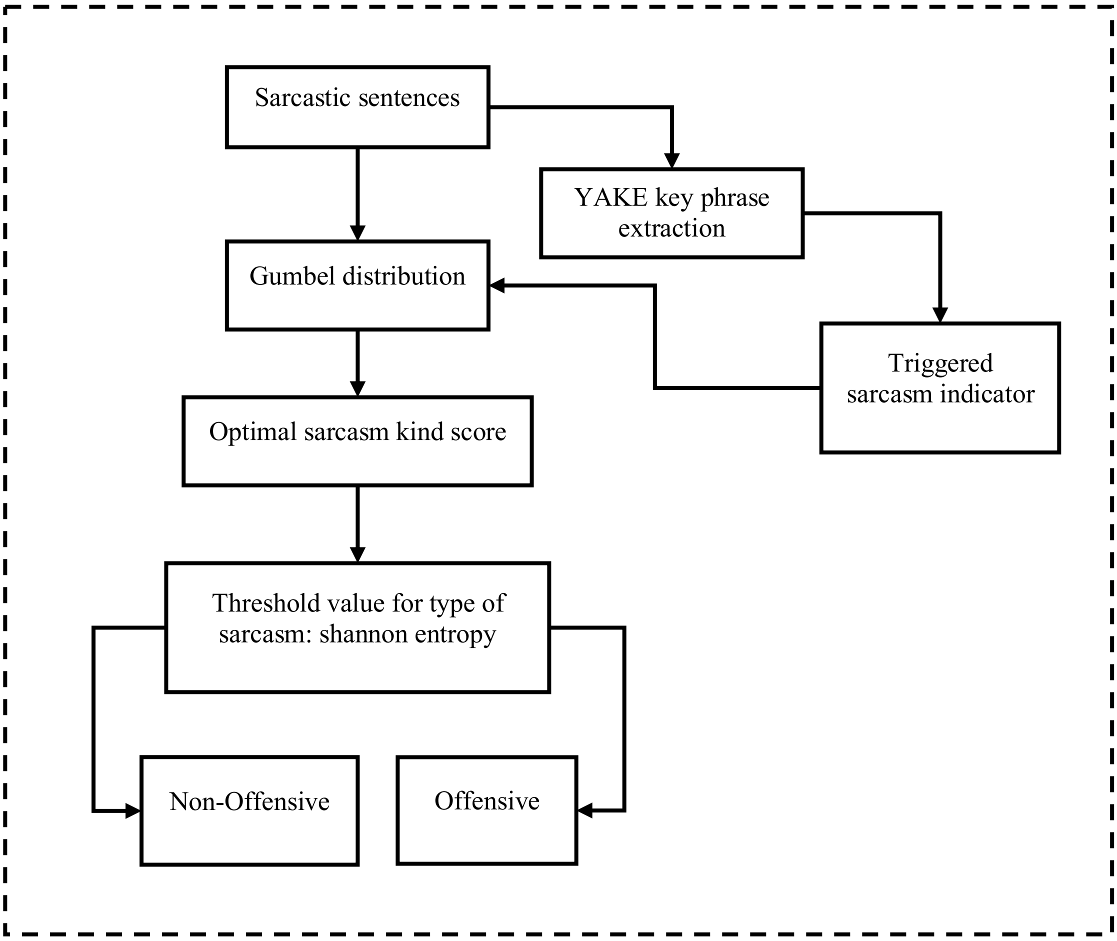


Fig. 5.10 Flowchart of SDWT-ELM for categorization of types of sarcasm as offensive and non-offensive

ALGORITHM 5.4: Classify inherent sarcasm tonal trait signal
Input: S_{sarc}, k_{max}^{5}, ψ_8, *and sarcasm signal threshold value* (Q_{α_5}).   *Output: Optimal sarcasm signal score* (α_5) *based on which classify sarcasm signals as offensive or not-offensive*   1. *Compute* α_5 *by formulating discrete optimization problem using poison distribution:* $\alpha_5 = \left\| Minimise \frac{e^{k_{max}^{5}} \times k_{max}^{5\ -\left(\frac{\psi_8}{10}\right)}}{\left(\frac{\psi_8}{10}\right)!} \right\|$, *Subject to:* $k_{max}^{5} \geq 0$ *and* $\psi_8 \geq 0$ 2. *Classification based on comparison between* α_5 *and* Q_{α_5}*:*   *if* $\alpha_5 \leq Q_{\alpha_5}$*: return "offensive"*   *Else return "not- offensive"*   3. *End*

Fig. 5.10 gives a detailed illustration of offensive language depiction in the SDWT-ELM method. First, we filter the sarcastic sentences generated through SDWT-ELM, associated with its optimal sarcasm score (ψ_9). Additionally, the YAKE model is utilized to get the highest sarcasm-triggered score (k_{max}^{7}). Furthermore, we perform the Gumbel distribution mechanism to depict abusive content with the obtained optimum sarcasm kind value (α_6). The Gumbel distribution employed in the proposed investigation is shown in *Eqn. (5.20)* which gives the optimum sarcasm kind value (α_6).

$$\alpha_6 = e^{-e^{\frac{-(\psi_9 - \mu_{\psi_9})}{k_{max}^{7}}}} \tag{5.20}$$

The μ_{ψ_9} is the average value taken for the combined ψ_9 scores over the dataset, i.e., $\mu_{\psi_9} = \frac{\sum_{i=1}^{i=m} \psi_{9_i}}{2}$, where m is the total number of sentences. This ψ_9 is then compared with the threshold value for types of sarcasm (Q_{α_6}) calculated using Shannon entropy as shown in *Eqn. (5.21)*.

$$Q_{\alpha_6} = -\sum p(\psi_9)\log(p(\psi_9)) \tag{5.21}$$

The Q_{α_6} is the probability of the optimal score taking on the value ψ_{9_i}, where $1 \leq i \leq m$ and m is the total number of sentences in the dataset. The value of Q_{α_6} is computed as 2.7. If $\alpha_6 < Q_{\alpha_6}$, then the sentence is tagged as "*offensive*," and if $\alpha_6 \geq Q_{\alpha_6}$, then the sentence is tagged as "*non-offensive*." *Algorithm 5.5* defines the abusive content interpreted using the probability distribution structure.

ALGORITHM 5.5: Offensive content depiction from sarcastic sentences
Input: ψ_9, k_{max}^7 *and* μ_{ψ_9} *Output: Offensive and Non-offensive* 1. *Calculate the veracity disposition of sentence:* $\alpha_6 = e^{-e^{\frac{-(\psi_9-\mu_{\psi_9})}{k_{max}^7}}}$ 2. *If* $\alpha_6 \geq Q_{\alpha_6} \rightarrow$ *"Offensive"* 3. *Else "Non-offensive"*

5.7.4 Proposed Method for Humour and Non-Humour

In this subsection, we address types of sarcasm in terms of humour and non-humour text. Humorous sarcasm is light-hearted, witty, and intended to amuse or gently mock. *Fig. 5.11* shows the framework of the proposed method to detect types of sarcasm.

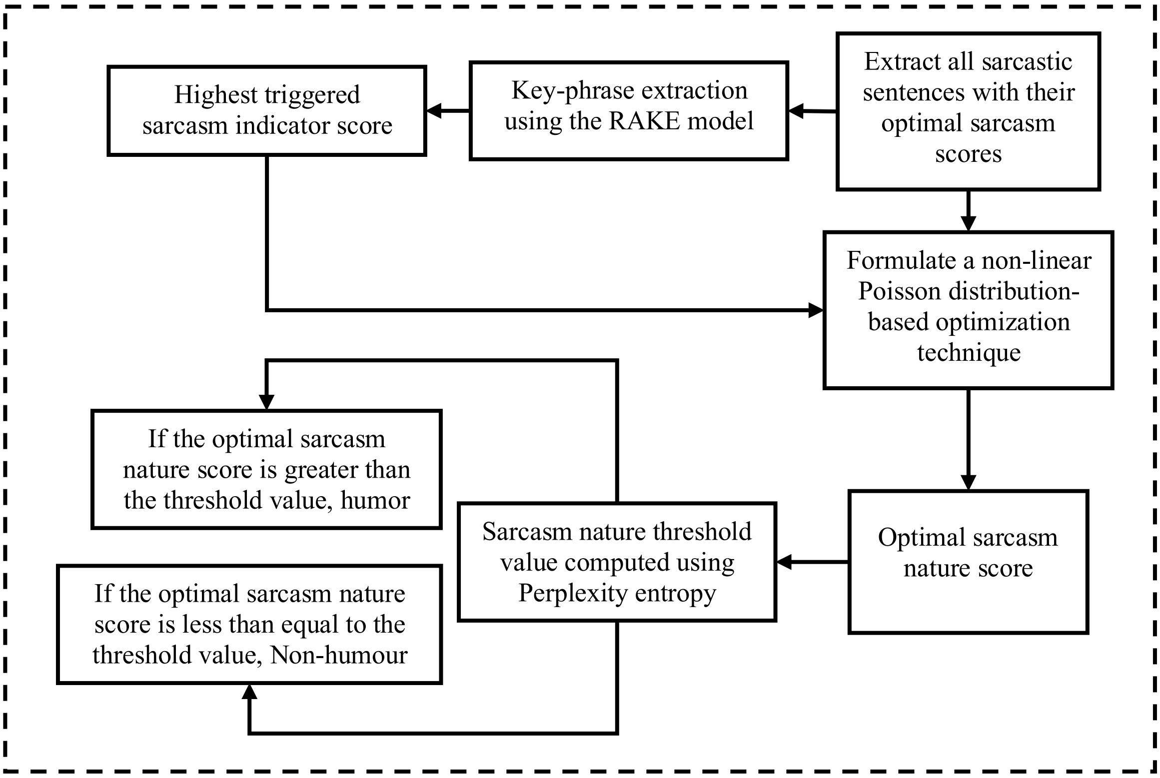


Fig. 5.11 Architectural diagram of SLIOEN for categorization of types of sarcasm as humour and non-humour

We apply the SLIOEN model, which focuses on fine-grained humour hierarchy generation, to distinguish between them. The model extracts and weighs features like exaggeration, surprise, tone, and incongruity, which are strong indicators of humour. The Poisson function models sporadic humorous cues structure, where laughter triggers and occur at low but impactful frequencies. By minimizing the Poisson function, the model effectively isolates humour-driven indicators from general sarcasm cues.

The optimization process ensures that the SLIOEN model does not confuse criticism with comedy, allowing for more accurate classification across real-world sarcastic expressions. After getting ψ_6 from SLIOEN, we extract sarcastic sentences and apply the RAKE model to get the highest keyword value (k^3_{max}). The k^3_{max} and ψ_6 are further interjected into the formulated non-linear optimization technique using the Poisson distribution. Utilization of this distribution helps in determining the syntactic markers, sentiment features, use of punctuation sets, and language transmission.

The aim of the formulated optimization technique is to get the optimal solution of the minimized objective function. The optimal value obtained from this optimization technique is termed the optimal sarcasm nature score (α_7). *Eqn. (5.22)* shows the optimization technique using the minimized Poisson distribution as the objective function.

$$\left.\begin{array}{c}\alpha_7 = \left|\text{Minimise } \frac{e^{k^3_{max} \times \rho - \psi_6}}{\psi_6!}\right| \times 100 \\ Subject\ to, \psi_6 \geq 0 \text{ and } k^3_{max} \geq 0\end{array}\right\} \tag{5.22}$$

In this step, we determine the inherent intent of sarcasm signals in terms of humour and non-humour. The proposed method used a threshold value for the type of sarcasm as (Q_{α_7}) defined in *Eqn. (5.23)*, which is calculated using Shannon entropy as shown in *Eqn. (5.24)*.

$$Q_{\alpha_7} = 2^{\sigma} \tag{5.23}$$

$$\sigma = \frac{\sum_{i=1}^{W} p(\psi_{6_i}) \log_2(\psi_{6_i})}{10} \tag{5.24}$$

The $p(\psi_6)$ is the probability distribution of obtaining the optimal sarcasm nature score of the sentence. With the help of this structure, the Q_{α_7} is computed as 0.688. *Table 5.6* demonstrates the threshold value scales used in the proposed method.

The classification of the type of sarcasm is done based on the comparison between α_7 and Q_{α_7}. If $\alpha_7 > Q_{\alpha_7}$, then the sentence is tagged as "*humour*", else it is tagged as "*non-humour.*" *Algorithm 5.5* defines the procedure of depicting the inherent, implicit intent of intonations affecting sarcasm in terms of humor and non-humor.

Table 5.6 Parameters employed in the proposed method

Parameters	***Values***
Sarcasm Type Threshold value ($\boldsymbol{Q_{\alpha_7}}$)	*Dataset -1 [149]:*$\sigma_{Dataset-1} = \frac{\sum_{i=1}^{w} p(\psi_{7_i})\, log_2(\psi_{7_i})}{10} = -0.5837$*;* $Q_{\alpha_7} = 2^{\sigma} = 2^{-0.5837} = 0.6672 \approx 0.688$; *Dataset -2 [150]:* $\sigma_{Dataset-2} = \frac{\sum_{i=1}^{w} p(\psi_{7_i})\, log_2(\psi_{7_i})}{10} = -0.5183$*;* $Q_{\alpha_7} = 2^{\sigma} = 2^{-0.5183} = 0.6981 \approx 0.688$ *Dataset -3 [151]:* $\sigma_{Dataset-3} = \frac{\sum_{i=1}^{w} p(\psi_{7_i})\, log_2(\psi_{7_i})}{10} = -0.5261$*;* $Q_{\alpha_7} = 2^{\sigma} = 2^{-0.5261} = 0.6944 \approx 0.688$

ALGORITHM 5.5: Classify inherent indicators affecting sarcasm using a non-linear optimization problem
Input: S_{sarc}, ψ_6, k_{max}^3, *and sarcasm nature threshold score* (Q_{α_7}).   *Output: Optimal sarcasm nature score* (α_7) *based on which to classify sarcasm indicators as humour or non-humour*   *1. Compute* α_7 *by formulating a non-linear optimization problem using a poison distribution:* $\alpha_7 = \left\| Minimise\ \frac{e^{k_{max}^3 \times \rho^{-\psi_6}}}{\psi_6!} \right\| \times 100$*, Subject to:* $\psi_6 \geq 0$ *and* $k_{max}^3 \geq 0$ *2. Classification based on the comparison between* η *and sarcasm nature threshold score* (∂)*: if* $\alpha_7 > Q_{\alpha_7}$*; return "humour"*   *Else return; "non-humour."*   *3. End*

5.8 Experimental and Result Analysis

The primary focus of this thesis is sarcasm detection. We validated each of the proposed methods on established datasets and benchmarked them against current state-of-the-art methods. These experiments confirm that the proposed methods maintain competitive accuracy and robustness in detecting sarcastic expressions in English text.

In addition to the main task of sarcasm detection, we introduce novel concepts for the types of sarcasm categorization. This categorization of sarcastic statements into distinct types includes humour, rhetorical devices, like-prefixed, dropped negation, hyperbole, implicit, explicit, callous, and offensive, using optimization techniques. Since the proposed methods for the types of sarcasm are entirely new, there are no annotated datasets for quantitative comparisons. Therefore, evaluation is primarily qualitative, and it highlights the potential benefits of distinguishing between various types of sarcasms.

To assess the reliability of our proposed methods, SDO-BCRT and UCRT-CGKO, we carried out an inter-annotator agreement analysis focused on classifying different types of sarcasm. For this task, we engaged human annotators who were asked to review and label each sarcastic sentence types. This helped us gauge how consistently sarcasm types were interpreted and provided meaningful feedback on the accuracy and sensitivity of our models.

The annotator first performs an analysis of the callous nature of sarcastic sentences within the proposed SDO-BCRT method. The three annotators were assigned to tag the sarcastic sentences as callous or non-callous. The annotators are from various domains, such as computer enthusiasts, social media analysis experts, and English linguistics experts. Each expert independently labelled the sarcastic sentences of the datasets as callous or non-callous, as shown in *Table 5.7*.

Table 5.7 Statistics of callous and non-callous comments among three annotators

Dataset	***Sarcastic sentences***	***Annotator 1***		***Annotator 2***		***Annotator 3***		***Average***	
		Callous	***Non-Callous***	***Callous***	***Non callous***	***Callous***	***Non callous***	***Callous***	***Non callous***
Dataset-1 [149]	*5975*	*2463*	*3512*	*1476*	*4499*	*1511*	*4464*	*2496.6 ≈ 2497*	*3478.3 ≈ 3478*
Dataset-2 [150]	*14592*	*6586*	*8006*	*3782*	*10810*	*4953*	*9639*	*6603*	*7989*
Dataset-3 [151]	*5743*	*2539*	*3204*	*1537*	*4206*	*1481*	*4262*	*2085.6 ≈ 2086*	*3657.3 ≈ 3657*
Dataset-4 [152]	*5156*	*2322*	*2834*	*1781*	*3375*	*1466*	*3690*	*2323*	*2833*

For each pair of annotators (1 vs. 3, 1 vs. 2, and 2 vs. 3), we compute the Cohen's kappa score between annotators 1 and 2, 1 and 3, and 2 and 3 pairwise as shown in *Table 5.8*. Cohen's Kappa is defined as in *Eqn. (5.25)*.

$$\varepsilon = \frac{h_0 - h_e}{1 - h_e} \tag{5.25}$$

Where h_0 is the observed proportion of agreement between the annotators. h_e is the expected proportion of agreement by chance, based on the marginal probabilities of each category. Once we have the pairwise values, we compute the average Cohen's kappa score. Averaging all these ε_{avg} gives an overall indication of very high inter-annotator agreement, as shown in *Eqn. (5.26)*.

$$\varepsilon_{avg} = \frac{(\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4)}{3} \tag{5.26}$$

Table 5.8 Cohen's Kappa score between three annotators for datasets 2, 3, and 4

	ε for Dataset-1 [149]	***ε for Dataset-2 [150]***	***ε for Dataset-3 [151]***	***ε for Dataset-4 [152]***
Annotators 1 and 2	*0.90*	*0.92*	*0.91*	*0.83*
Annotators 2 and 3	*0.80*	*0.84*	*0.82*	*0.90*
Annotators 1 and 3	*0.70*	*0.77*	*0.74*	*0.91*
Cohen's Kappa averaged score (ε_{avg})	*0.80*	*0.84*	*0.82*	*0.88*

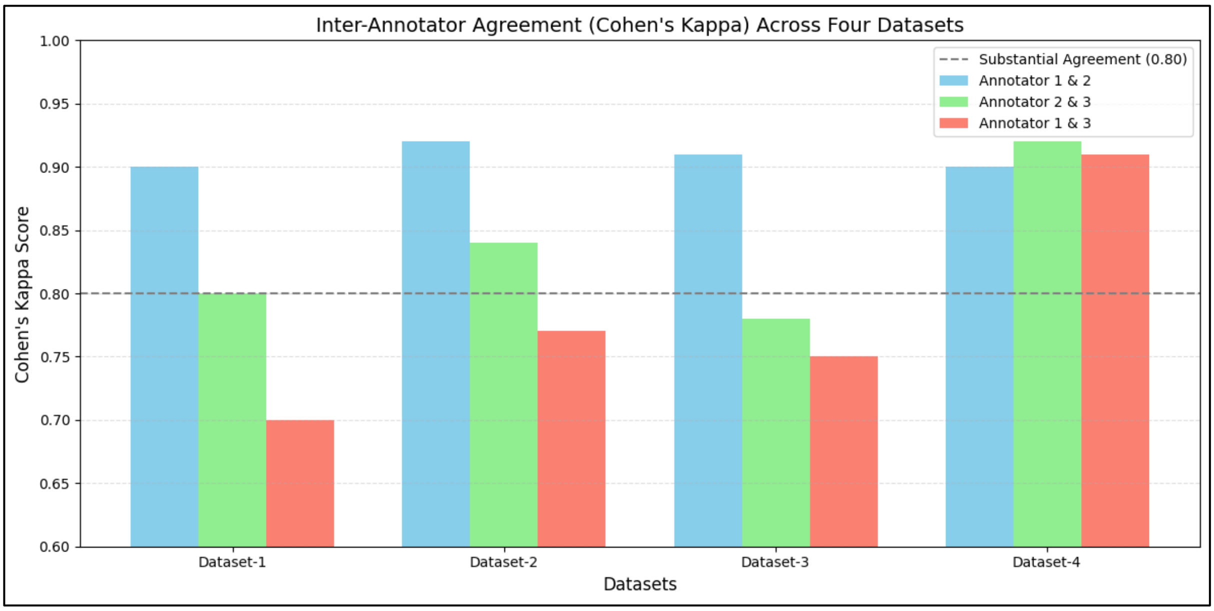


Fig. 5.12 Cohen's kappa scores of three annotators for dataset-1, 2, 3, and 4 with a substantial agreement line

The computed ε_{avg_1}, ε_{avg_2}, ε_{avg_3}, and ε_{avg_4} are 0.80, 0.84, 0.82, and 0.88 for datasets 2, 3, and 4, respectively. We calculate the final average Cohen's kappa score for each of the four datasets, resulting in a combined kappa value of 0.83. *Fig. 5.12* shows the overall kappa score among annotators 1, 2, and 3, which gives substantial agreement and confirms consistency in labelling callous and non-callous sarcasm. This high inter-annotator reliability validates our unsupervised callousness classification approach.

After obtaining the annotated datasets, we evaluate the performance metrics of the proposed (SDO-BCRT) method. Then, we assessed callousness by averaging the annotations from three annotators, as shown in *Table 5.7*. *Table 5.9* illustrates the distribution of callous and non-callous statements found within sarcastic sentences by the proposed SDO-BCRT method.

Table 5.10 shows the comparison analysis between three annotators and the proposed method in terms of performance metrics, i.e., accuracy, precision, recall, and F1 score.

Table 5.9 Distribution of sentences over the four datasets

Datasets	**Callous**	**Non-callous**	**Total number of sarcastic sentences**
Dataset-1 [149]	*2,214*	*3,761*	*5,975*
Dataset-2 [150]	*6,433*	*8,159*	*14,592*
Dataset-3 [151]	*1,570*	*4,173*	*5,743*
Dataset-4 [152]	*2,489*	*2,667*	*5,156*

Table 5.10 SDO-BCRT analysis with averaged scores from three annotators on dataset-1, 2, 3, and 4

	Dataset-1 [149]				***Dataset-2 [150]***			
	Accuracy	*Precision*	*Recall*	*F1 score*	*Accuracy*	*Precision*	*Recall*	*F1 score*
SDO-BCRT	*0.9255*	*0.9072*	*0.9183*	*0.9127*	*0.9633*	*0.9365*	*0.9759*	*0.9558*
	Dataset-3 [151]				***Dataset-4 [152]***			
	Accuracy	*Precision*	*Recall*	*F1 score*	*Accuracy*	*Precision*	*Recall*	*F1 score*
SDO-BCRT	*0.9058*	*0.8588*	*0.9164*	*0.8867*	*0.9057*	*0.8982*	*0.8838*	*0.8909*

Fig. 5.13 shows the ROC curve for callousness classification with enhanced AUC values of 0.92, 0.97, 0.91, and 0.90 for datasets 2, 3, and 4, respectively. This ensures that the callousness classification is empirically validated and effectively benchmarked in SDO-BCRT.

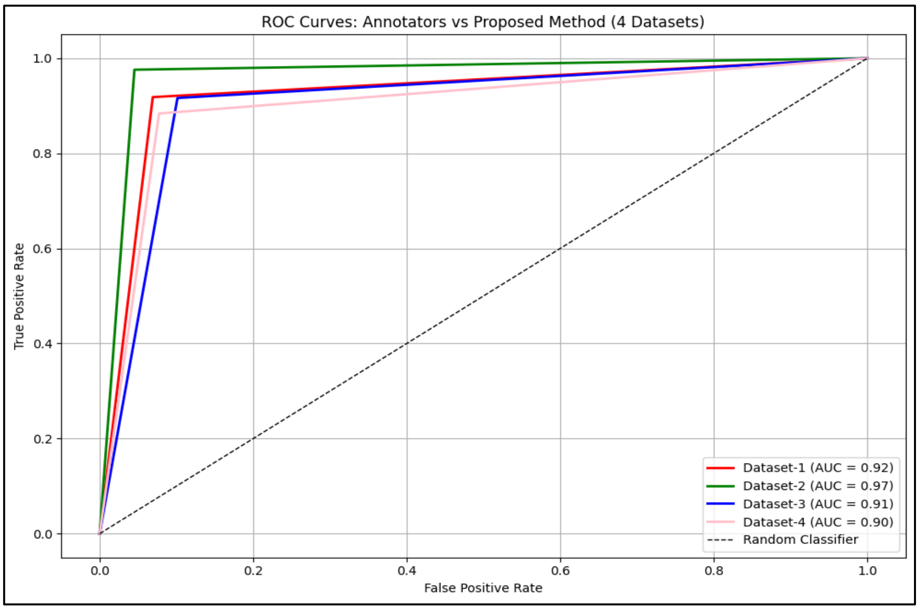
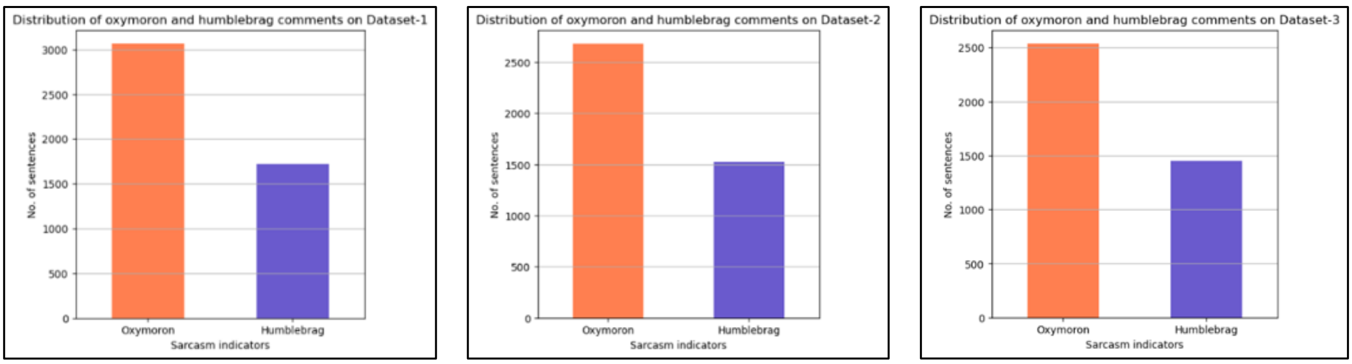


Fig. 5.13 ROC curve for callousness classification on datasets 1, 2, 3, and 4

Fig. 5.14 (a), (b), and (c) give the distribution plot of the obtained humblebrag and oxymoron sentences taken from the sarcastic sentences using the proposed methodology (SD-GKOT).

Now, we again perform inter-annotator agreement analysis on the SLIOEN method to depict the presence of humour and non-humour contexts in sarcastic sentences. As in the previous section, we recruited the same human annotators to perform the task. These annotators are assigned the task of tagging the sarcastic sentences as humorous or non-humorous. Each expert independently labelled the sarcastic sentences as humour or non-humour, as shown in *Table 5.11*. For each pair of annotators (1 vs. 3, 1 vs. 2, and 2 vs. 3), we compute the Cohen's kappa score between annotators 1 and 2, 1 and 3, and 2 and 3 pairwise as shown in *Table 5.12*. Cohen kappa score is defined as in *Eqn. (5.26)*.



(a) (b) (c)

Fig. 5.14 Distribution plot for oxymoron and humblebrag comments on (a). dataset-1, (b). dataset-2 and (c). dataset-3

Table 5.11 Statistics of humour and non-humour comments among three annotators

Dataset	Sarcastic sentences counts	Annotator 1		Annotator 2		Annotator 3		Average	
		Humour	Non-Humour	Humour	Non-Humour	Humour	Non-Humour	Humour	Non-Humour
Dataset-1 [149]	*5755*	*2122*	*3633*	*2064*	*3691*	*2247*	*3508*	*2144.3 ≈ 2144*	*3610.6 ≈ 3611*
Dataset-2 [150]	*7367*	*2538*	*4829*	*1996*	*5371*	*2471*	*4896*	*2353*	*5032*
Dataset-3 [151]	*4884*	*2780*	*2104*	*3251*	*1633*	*2995*	*1889*	*3008.6 ≈ 3009*	*1875.3 ≈ 1875*

Table 5.12 Cohen's Kappa score between three annotators for dataset-1, 2 and 3

	ε for Dataset-1 [149]	***ε for Dataset-2 [150]***	***ε for Dataset-3 [151]***
Annotator 1 and 2	*0.89*	*0.91*	*0.83*
Annotator 2 and 3	*0.80*	*0.84*	*0.90*
Annotator 1 and 3	*0.79*	*0.80*	*0.82*
Cohen's Kappa averaged score (k_{avg})	*0.82*	*0.85*	*0.85*

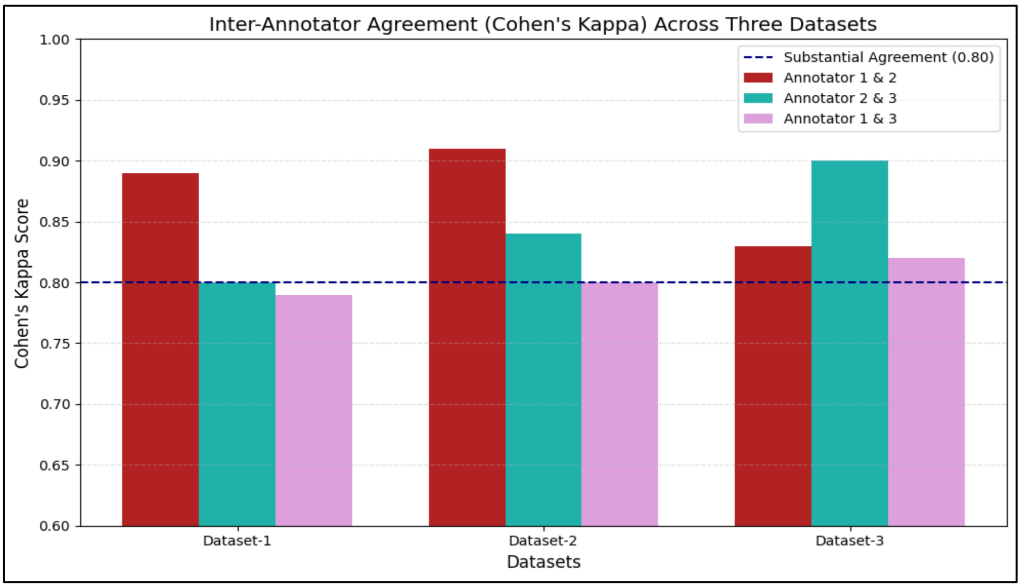


Fig. 5.15 Cohen's Kappa scores of three annotators for dataset-1, 2, and 3, with a substantial agreement line for SLIOEN

Once we have the pairwise values, we compute the average Cohen's kappa score. The computed ε_{avg_1}, ε_{avg_2}, and ε_{avg_3} is 0.82, 0.85, and 0.85 for dataset-1, 2, and 3 respectively. We calculated the final average Cohen's kappa score for each of the four datasets, resulting in a combined kappa value of 0.84. Figure 5.15 shows the overall kappa score among annotators 1, 2, and 3, indicating a substantial agreement, confirming consistency in labelling humour and non-humour sarcasm. This high inter-annotator reliability validates the unsupervised humour classification methods.

After obtaining the annotated datasets, we evaluate the performance metrics of the proposed method. Then, we assessed humour by averaging the annotations from three annotators, as shown in *Table 5.12*. A comparison between these averaged values and the proposed method's results for humour and non-humour comments is presented in *Table 5.13*. *Table 5.14* illustrates the distribution of humour and non-humour statements found within sarcastic sentences by the proposed method.

Table 5.14 shows the comparative analysis between three annotators and the proposed method in terms of performance metrics, i.e., accuracy, precision, recall, and F1 score.

Table 5.13 Distribution of sentences over the four datasets

Datasets	Humour	Non-humour	Total number of sarcastic sentences
Dataset-1 [149]	*2,606*	*3,149*	*5,755*
Dataset-2 [150]	*3,350*	*4,017*	*7,367*
Dataset-3 [151]	*2,242*	*2,642*	*4,884*

Table 5.14 SLIOEN analysis with averaged scores from three annotators on dataset-1, and 2

	Dataset-1 [149]				**Dataset-2 [150]**			
	Accuracy	*Precision*	*Recall*	*F1 score*	*Accuracy*	*Precision*	*Recall*	*F1 score*
SLIOEN	*0.9109*	*0.8973*	*0.9027*	*0.8999*	*0.9333*	*0.9164*	*0.9480*	*0.9319*

	Dataset-3 [151]			
	Accuracy	*Precision*	*Recall*	*F1 score*
SLIOEN	*0.9005*	*0.9352*	*0.8953*	*0.9148*

Fig. 5.16 shows the ROC curve for humour classification with enhanced AUC values of 0.92, 0.91, and 0.90 for dataset-1, 2, and 3, respectively. This ensures the humour classification is both empirically validated and effectively benchmarked in SLIOEN.

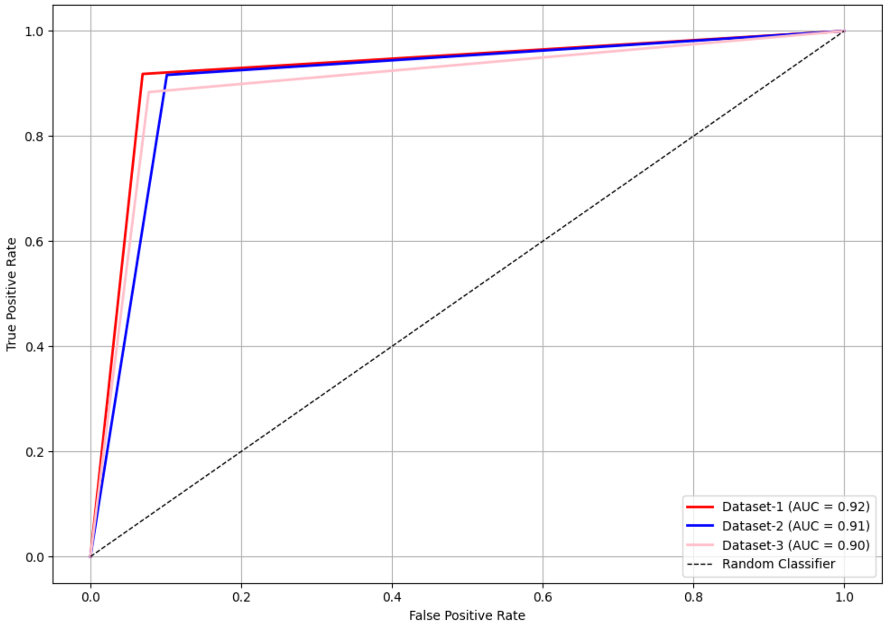


Fig. 5.16 ROC curve for humour classification on dataset- 1, 2, and 3

In this section, the study about the inter-annotator agreement scheme of the offensive nature of sarcastic sentences is performed in UCRT-CGKO. We have taken three annotators, who are assigned to tag the sarcastic sentences as offensive or non-offensive. Each expert independently labelled the sarcastic sentences of the datasets as offensive or non-offensive, as shown in *Table 5.15*.

After obtaining the annotated datasets, we evaluate the performance metrics of the proposed method. Then, we assessed offensiveness by averaging the annotations from three annotators, as shown in *Table 5.16*.

Table 5.15 Statistics of offensive and non-offensive comments among three annotators in UCRT-CGKO

Dataset	*No. of sarcastic sentences*	*Annotator 1*		*Annotator 2*		*Annotator 3*		*Average*	
		Offensive	*Non-Offensive*	*Offensive*	*Non-Offensive*	*Offensive*	*Non-Offensive*	*Offensive*	*Non-Offensive*
Dataset-1 [149]	*2038*	*1375*	*663*	*1478*	*560*	*1434*	*736*	*1429*	*653*
Dataset-2 [150]	*2270*	*1415*	*855*	*1400*	*870*	*1511*	*759*	*1442*	*828*

Table 5.16 Cohen's Kappa score between three annotators for dataset-1 and 2 for UCRT-CGKO

	k for Dataset-1 [149]	***k for Dataset-2 [150]***
Annotator 1 and 2	*0.944*	*0.948*
Annotator 2 and 3	*0.923*	*0.927*
Annotator 1 and 3	*0.943*	*0.961*
Cohen's Kappa averaged score (k_{avg})	*0.936*	*0.945*

The computed k_{avg_1}, and k_{avg_2} is 0.936 and 0.945 for dataset-1 and 2 respectively. We calculate the final average Cohen's kappa score for each of the two datasets, resulting in a combined kappa value of 0.9405. The *Fig. 5.17* shows the overall kappa score among annotators 1, 2, and 3 indicates substantial agreement, confirming consistency in labelling offensive vs. non-offensive sarcasm. This high inter-annotator reliability validates the unsupervised offensiveness classification method.

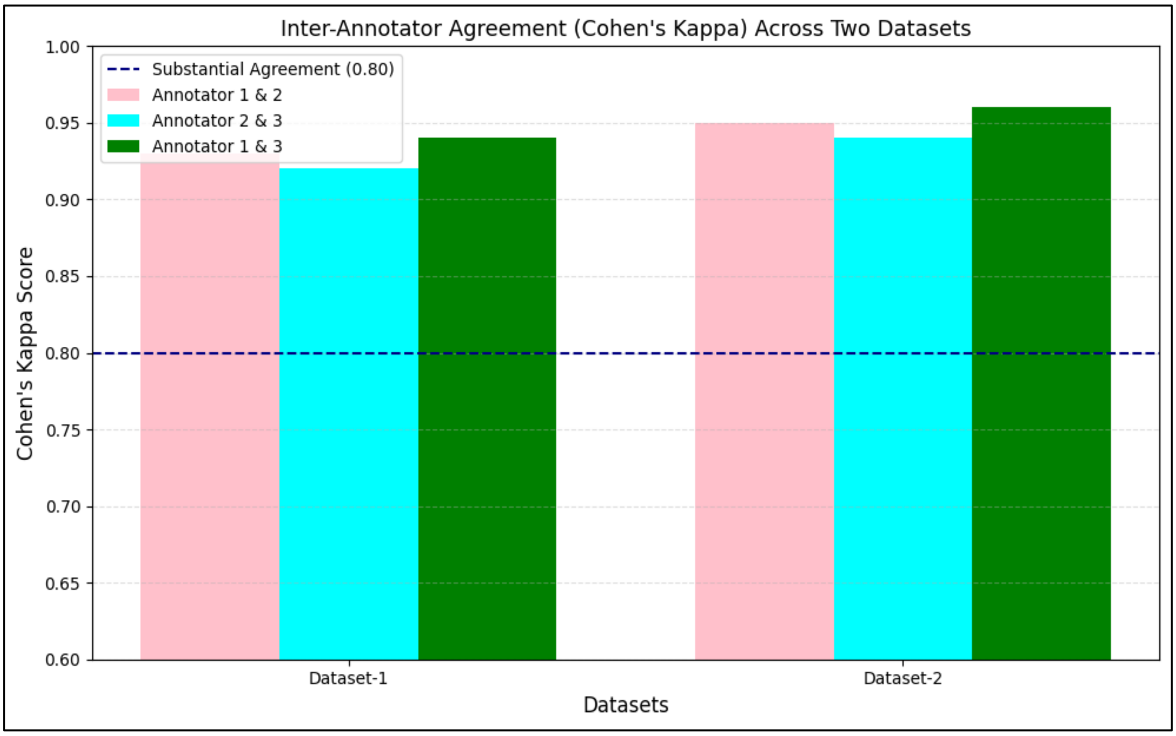


Fig. 5.17 Cohen's Kappa scores of three annotators for dataset-1 and 2 with a substantial agreement line in UCRT-CGKO

Table 5.17 illustrates the distribution of offensive and non-offensive statements found within sarcastic sentences by the proposed method.

Table 5.17 Distribution of sentences over the two datasets in UCRT-CGKO

Datasets	**Offensive**	**Non-Offensive**	**Total number of sarcastic sentences**
Dataset-1 [149]	*1,417*	*6,21*	*2,038*
Dataset-2 [150]	*1,456*	*7,14*	*2,270*

Fig. 5.18 shows the ROC curve for offensiveness classification with enhanced AUC values of 0.97 and 0.98 for dataset-1 and dataset-2, respectively. It ensures the offensiveness classification is both empirically validated and effectively benchmarked in UCRT-CGKO. *Table 5.18* shows the comparison analysis between three annotators and the proposed method in terms of performance metrics, i.e., accuracy, precision, recall, and F1 score.

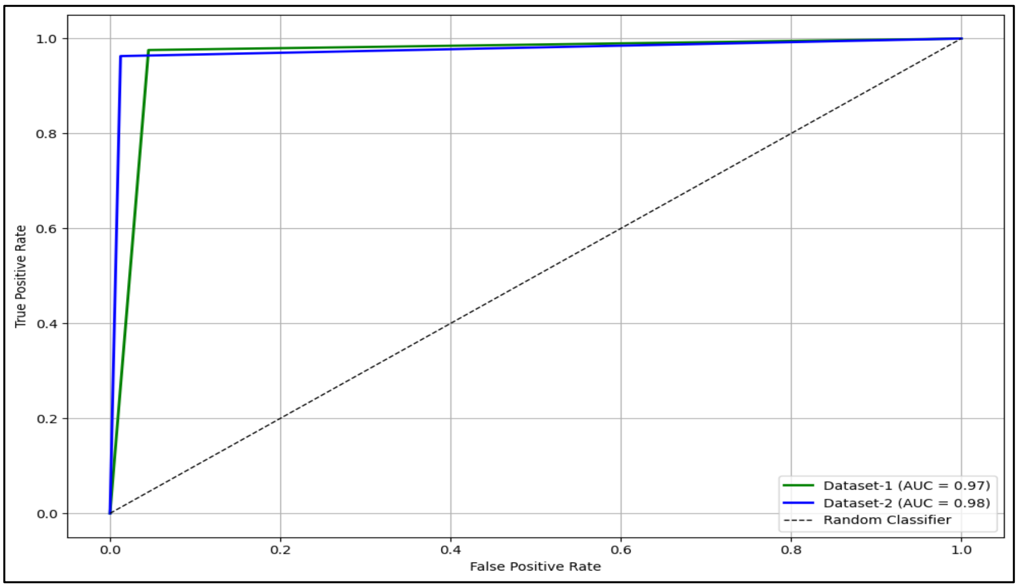


Fig. 5.18 ROC curve for offensiveness classification on dataset- 1 and 2

Table 5.18 UCRT-CGKO analysis from three annotators on dataset-1 and 2

	Dataset-1 [149]				***Dataset-2 [150]***			
	Accuracy	***Precision***	***Recall***	***F1 score***	***Accuracy***	***Precision***	***Recall***	***F1 score***
UCRT-CGKO	*0.9546*	*0.8993*	*0.9592*	*0.9282*	*0.9687*	*0.9375*	*0.9044*	*0.9206*

These experimental findings suggest that the new classification for types of sarcasm features promises a more nuanced linguistic analysis application. However, due to the absence of annotated datasets for the types of sarcasm, we suggest future research focusing on developing standardized benchmark datasets for sarcasm types. Such efforts will enable more rigorous quantitative comparisons and further advances in this emerging area.

In conclusion, the experimental results support the effectiveness of all proposed methods for demonstrating the innovative potential of incorporating classification for types of sarcasm. This study has given new ideas for future research, leading to more sophisticated models with the capability of capturing the multifaceted nature of sarcastic communication.

5.9 Summary

This chapter presented a comprehensive categorization and detection of types of sarcasm, focusing on how each form shows underlying intention using various optimization techniques. This chapter gives study for types of sarcasm analysis through meaning inversion, implicit and explicit cues, and masked intentions driven context.

To address the detection challenges for types of sarcasm, we introduce various optimization-based methodologies. Distributions such as parabolic fractal, Gauss-Kuzmin, Poisson, logistic, and Gumbel are minimized or maximized to capture specific behavioural patterns in sarcastic English text. Additionally, metaheuristic algorithms like hunter-prey optimization are applied to tune feature weights, sentiment contrasts, and polarity shifts.

By aligning the nature of sarcasm with suitable mathematical frameworks, this chapter demonstrated that sarcasm detection can move beyond binary classification. The proposed methods support fine-grained sarcasm interpretation, improving accuracy and contextual sensitivity. The outcomes of this chapter form a foundation for developing intelligent sarcasm detection methods for future perspectives.

Chapter 6

Sarcasm Detection of Hindi Text

The amount of Hindi content on the internet continues to rise through weblogs and reviews. It is crucial to understand sarcasm in Hindi as it can change the meaning of a sentence entirely. Understanding sarcasm in Hindi text helps in analysing opinions, which is useful in social media analysis and customer feedback.

Most existing research on sarcasm detection focuses on English and other widely used languages, with limited attention to Hindi. Additionally, the use of optimization techniques in this context remains largely unexplored. This chapter presents an unsupervised method for detecting sarcasm in Hindi text using optimization-based techniques.

The organization of sections in this chapter is as follows. In *section 6.1,* we introduce the proposed method, namely - “simulated probabilistic optimisation Zipf Mandelbrot distribution via entropy paradigm for sarcasm detection (SPOZ-ESD)”. In *section 6.2,* we present the description of the Hindi datasets and a comparison analysis of the proposed method with the state-of-the-art systems. *Section 6.3* provides details about the statistical validation tests used in the proposed study.

The challenges and time complexity are also included in this section. Finally, *Section 6.4* concludes the chapter and presents the key findings.

6.1 Introduction

Hindi serves as the official language of India, playing a significant role in day-to-day communication [26]. It is one of the most spoken languages in the world, with a vast number of native speakers. Hindi holds a strong presence on the internet, especially across social media and digital web platforms. Its use in online education, entertainment, and public discourse continues to expand rapidly [98]. This rising digital footprint highlights Hindi’s growing importance in global language networks [178].

Understanding sarcasm in Hindi is important due to its frequent use in everyday conversations. It appears in humour, criticism, and social commentary, often carrying layered meanings. Learning to detect it helps improve communication and emotional interpretation in society. It also supports better analysis of online content, especially on social media platforms. Sarcasm detection in Hindi is thus essential for both linguistic research and real-world applications.

Sarcasm detection in Hindi text involves identifying instances where the intended meaning contrasts sharply with the literal expression. This phenomenon is particularly complex in Hindi due to its rich linguistic structure and culturally embedded forms of expression. Hindi follows the free-word order arrangement of the words. *Fig. 6.1* shows three sentences that depict similar meanings with different free word order rearrangements.

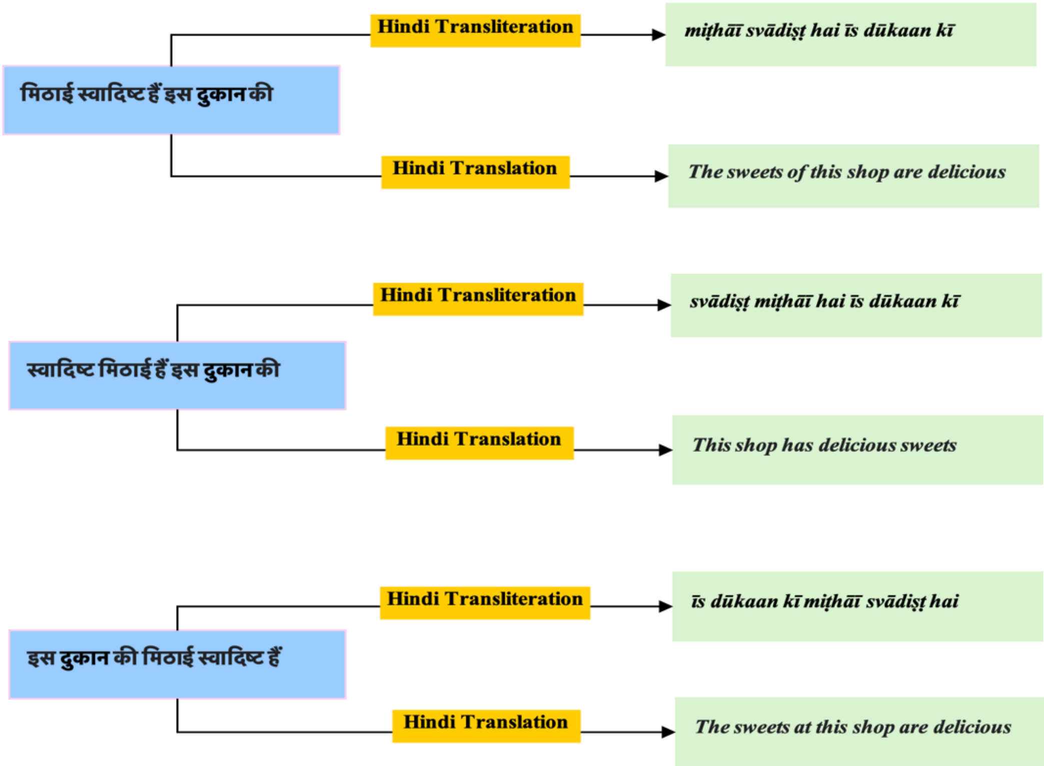
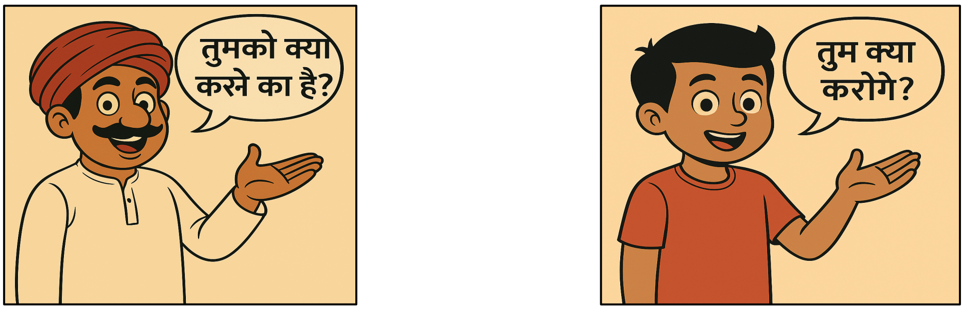


Fig. 6.1 Example of Hindi comments demonstrating free word order

Lot of work in sarcasm detection is done in English [14], [15] text. However, when it comes to Hindi, limited resources are available to fully capture the aspects of the figure of speech [16]. The field of sarcasm detection in Hindi has been limited by the absence of categorized datasets, fewer sentiment ratings, and a lack of sentiment analysis tools, such as part-of-speech (POS) taggers [25], [26].

Some existing methods have used machine learning [39], [46], deep learning [179], [180], and transformer [181], [182] based techniques for sarcasm detection in the Hindi text. However, these methods do not correctly and consistently classify the Hindi sentences. Speakers often use different tones and complex patterns, which makes it hard to detect sarcasm in Hindi text

[183]. Hindi exhibits considerable tonal, phonetic, and structural variation across different regions of India, making its interpretation highly dependent on geographical and linguistic context. These variations are shaped by regional languages that influence pronunciation, accent, and word usage in spoken Hindi [23]. Many of these languages, such as Marathi, Bengali, and others, use the Devanagari script, leading to shared vocabulary but differing speech patterns. For instance, speakers in Uttar Pradesh may pronounce 'वचन' (vachan) as 'बचन' (bachan), replacing 'व' (va) with 'ब' (ba), while Bengali speakers may pronounce 'आप' (aap) as 'ओप' (op), substituting 'अ' (a) with 'ओ' (o). Similarly, a Marathi speaker might say "तुमको क्या करने का है?", while a Delhi speaker may use "तुम क्या करोगे?", both expressing similar intent with distinct regional influence shown in *Fig. 6.2*.



(a). (b).

Fig. 6.2 Tonal variations in Hindi text in distinct regions, such as (a). Marathi and (b). Delhi

Sarcasm, which often relies on tone, wordplay, and cultural context, becomes even harder to detect when layered with regional influences and informal speech [17], [184]. This complexity highlights the need for robust sarcasm detection models tailored specifically for Hindi, accounting for dialectal shifts and tonal ambiguity.

To address the given issues, we performed sarcasm detection of Hindi text using a mathematical optimization technique.

6.1.1 Proposed Method

The proposed method aims to detect sarcasm based on a mathematical optimization technique in Hindi comments. The SPOZ-ESD works in three core steps to perform sarcasm detection.

Step 1: Compute Individual Sentence Values (ISV) and Average Embedding Score

Step 2: Determine the Optimal Sentiment Score and Highest Triggered Sarcasm Indicator Score

Step 3: Detect Sarcasm in Hindi Text

The architectural flow of the proposed method SPOZ-ESD is shown in *Fig. 6.3*. The detailed explanations of steps for the proposed SPOZ-ESD method for sarcasm detection are given in subsequent subsections.

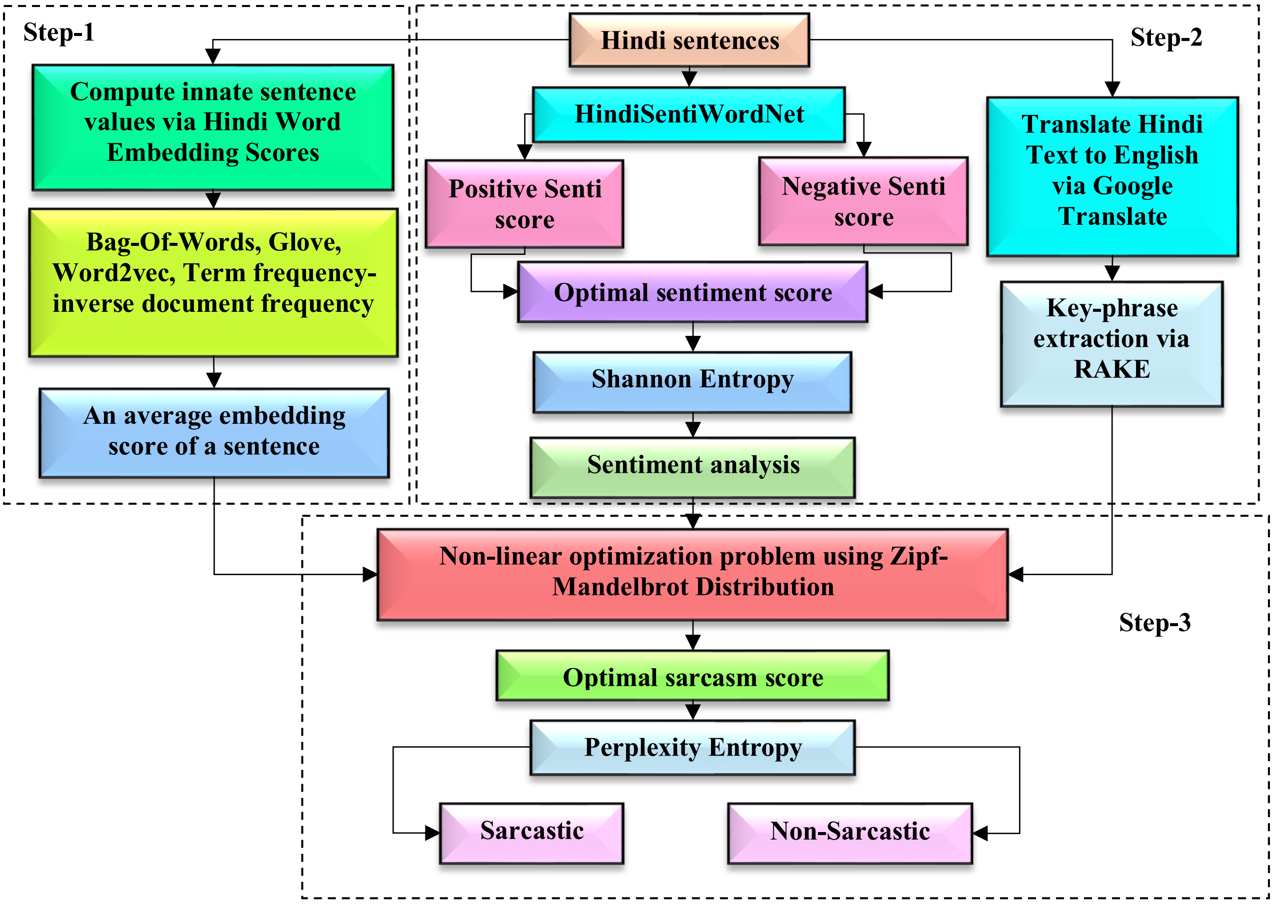


Fig. 6.3 Architectural flow of the proposed method (SPOZ-ESD) for sarcasm detection in Hindi text

6.1.1.1 Compute Individual Sentence Values (ISV) and Average Embedding Score

In the first step, the numeric values for the sentence incorporating four-word embedding algorithms, viz., BOW, Tf-Idf, word2vec, and GloVe are calculated. We generate four values - the sum of BOW (ISS_5), sum of Tf-Idf (ISS_6), sum of word2vec (ISS_7), and sum of glove (ISS_8) respectively. The ISV is the set of four individual sentence scores, such that ISV = $\{ISS_5, ISS_6, ISS_7, ISS_8\}$

Eqns. (6.1) to (6.4) depict the calculation of ISS_5, ISS_6, ISS_7, and ISS_8, respectively. Here, e represents the embedding vector for each method.

$$ISS_5 = \sum_{i=1}^{i=j} e^{BOW}{}_{w_i}, \text{ total summation value from BOW} \quad (6.1)$$

$$ISS_6 = \sum_{i=1}^{i=j} e^{TIF}{}_{w_i}, \text{ total summation value from Tf-Idf} \quad (6.2)$$

$$ISS_7 = \sum_{i=1}^{i=j} e^{W2V}{}_{w_i}, \text{ total summation value from word2vec} \quad (6.3)$$

$$ISS_8 = \sum_{i=1}^{i=j} e^{GLV}{}_{w_i}, \text{ total summation value from glove} \quad (6.4)$$

After calculating the ISV of a sentence, we further calculate the average embedding score (e^{avg}) by taking an average of four ISV values as shown in *Eqn. (6.5).*

$$e^{avg} = \frac{(ISS_5+ISS_6+ISS_7+ISS_8)}{4} \quad (6.5)$$

6.1.1.2 Determine the Optimal Sentiment Score and Highest Triggered Sarcasm Indicator Score

In the second step, we employ the HindiSentiWordNet (HSWN) to determine the overall sentiment of the Hindi sentence. The sentiment of the sentence obtained from HSWN is determined in terms of two polarities, i.e., the positive senti-score (PSS) and the negative senti-score (NSS). We compute the optimal sentiment score (ϕ_5) by considering the average absolute difference between PSS and NSS, as shown in *Eqn. (6.6)*.

$$\phi_5 = \frac{|PSS-NSS|}{2} \quad (6.6)$$

ALGORITHM 6.1: Calculate Optimal Sentiment Score using HSWN
Input: $W -$ *set of words in a sentence, HSWN,* λ_5 *Output:* ϕ_5*, positive sentiment or negative sentiment.* 1. *Let* $W = \{ w_1, w_2, w_3, \ldots, w_h \}$ *where* w_i *represents the* i^{th} $1 \le i \le m$ *words in the input sentence.* 2. *For all* w_i in the sentence *If* $w_i \in$ *HSWN,* $PSS = \frac{positive\ sentiment\ value\ of\ w_i}{k}$ *and* $NSS = \frac{neutral\ sentiment\ value\ of\ w_i}{k}$ 3. *Compute* $\phi_5 = \frac{\lvert PSS-NSS\rvert}{2}$ 4. *End*

The method to perform the sentiment classification and calculate the value of *PSS* and *NSS* is shown in *Algorithm 6.1*.

The sentiment of the sentence is analysed based on a sentiment threshold value (λ_5). If $\phi_5 \leq \lambda_5$, the sentence is tagged as "*positive sentiment*," else, the sentence is tagged as "*negative sentiment*." The value of λ_5 is calculated using the absolute Shannon entropy shown in *Eqn. (6.7)*.

$$\lambda_5 = -\sum_{i=1}^{m} p(\phi_{5_i}) \log_2(\phi_{5_i}) \tag{6.7}$$

The $p(\phi_5)$ is the probability distribution of the ϕ_5 of the sentence and m is the total number of sentences. The value of λ_5 is computed as 0.3 using Shannon entropy, as shown in *Table 6.1*.

Table 6.1 Computation of sentiment threshold value

Parameter	***Value***
Sentiment threshold value (λ_5)	*Dataset-1:* $\lambda_5 = -\sum_{i=1}^{6656} p(\phi_{5_i})\, log_2(\phi_{5_i}) \approx 0.3$ *Dataset-2:* $\lambda_5 = -\sum_{i=1}^{7169} p(\phi_{5_i})\, log_2(\phi_{5_i}) \approx 0.3$

Once we deduce the sentiment of the sentence, the next step is to identify the sarcasm indicator. We use the RAKE method to calculate the supreme keyword value (k_{max}^6). We first translate the Hindi text into English and then apply the RAKE method to get the tuple of words with respective numerical scores. Thus, we take keywords with the maximum score as a triggered sarcasm indicator. The highest keyword value, k_{max}^6, is used in optimization.

6.1.1.3 Detect Sarcasm in Hindi Text

In the final step, sarcasm detection is performed using a probabilistic non-linear optimization method. The optimization uses the Zipf-Mandelbrot distribution as the minimized objective function. The Zipf-Mandelbrot distribution used as an objective function helps to reduce errors by approximating the target function in sarcasm. *Eqn. (6.8)* shows the formulated optimization problem consisting of the Zipf-Mandelbrot distribution.

$$\left.\begin{array}{c} \psi_9 = \text{ Minimise } \dfrac{\frac{1}{(e^{avg}+k_{max}^6)^{|\phi_5|}}}{H_{m,k_{max}^6,|\phi_5|}} \times 100 \\ \text{Subject to, } H_{m,k_{max}^6,|\phi_5|} = \sum_{i=1}^{m} \frac{1}{(i+k_{max}^6)^{|\phi_5|}} \geq 0 \text{ and } \frac{1}{(e^{avg}+k_{max}^6)^{|\phi_5|}} \geq 0 \end{array}\right\} \tag{6.8}$$

The formulated non-linear optimization problem aims to minimize the objective function to get the sentence's optimal sarcasm score (ψ_9). The ψ_9 is then compared with the computed sarcasm detection threshold value (τ_9). It is calculated as 0.5 using the perplexity entropy [179] structure shown in *Eqn. (6.9)*. Its formulation is shown in *Table 6.2*.

$$\tau_9 = 2^{H(\psi_9)} \tag{6.9}$$

$$H(\psi_9) = \frac{\sum_{i=1}^{m} p(\psi_{9_i}) \log_2(\psi_{9_i})}{100} \tag{6.10}$$

The modified Shannon entropy ($H(\psi_9)$) is shown in *Eqn. (6.10)*, computed based on ψ_9. The $p(\psi_9)$ is the probability distribution of ψ_9.

Table 6.2 Computation of the sarcasm detection threshold value

Parameter	***Value***
Sarcasm detection threshold value (τ_9)	*Dataset-1[185]:* $\tau_9 = 2^{H(\psi_9)} = 2^{-1.10469} = 0.4657 \approx 0.5$ $H_{Dataset-1}: H(\psi_9) = \frac{\sum_{i=1}^{6656} p(\psi_{9_i})\, log_2(\psi_{9_i})}{100} = -1.1046$ *Dataset-2[186] [187]:* $\tau_9 = 2^{H(\psi_9)} = 2^{-1.0688} = 0.4773 \approx 0.5$ $H_{Dataset-2}: H(\psi_9) = \frac{\sum_{i=1}^{7169} p(\psi_{9_i})\, log_2(\psi_{9_i})}{100} = -1.0668$

ALGORITHM 6.2: Sarcasm Detection using the Proposed Method SPOZ-ESD
Input: $\boldsymbol{e^{avg}}$, $\boldsymbol{k_{max}^6}$, $\boldsymbol{\phi_5}$, $\boldsymbol{\tau_9}$ *Output: Sarcasm and non-sarcasm detected based on optimal sarcasm score* ($\boldsymbol{\psi_9}$)   1. *Compute* $\boldsymbol{\psi_9}$ *by formulating the non-linear optimization problem:* $\boldsymbol{\psi_9} = \text{Minimise} \frac{\frac{\mathbf{1}}{\left(e^{avg}+k_{max}^6\right)^{\lvert\phi_5\rvert}}}{\boldsymbol{H_{m,k_{max}^6,\lvert\phi_5\rvert}}} \times \mathbf{100};$ subject to, $\boldsymbol{H_{m,k_{max}^6,\lvert\phi_5\rvert}} = \sum_{i=1}^{m} \frac{\mathbf{1}}{\left(i+k_{max}^6\right)^{\lvert\phi_5\rvert}} \geq \mathbf{0}$ and $\frac{\mathbf{1}}{\left(e^{avg}+k_{max}^6\right)^{\lvert\phi_5\rvert}} \geq \mathbf{0}$ 2. *If* $\boldsymbol{\psi_9} \leq \boldsymbol{\tau_9} \rightarrow$ *"sarcasm"*   *Else "non-sarcasm"*   3. *End*

The decision of sarcasm detection is made based on the comparison between ψ_9 and τ_9. If $\psi_9 \leq \tau_9$, the sentence is tagged as "*sarcastic*," else, the sentence is tagged as "*non-sarcastic*." *Algorithm 6.2* defines the procedure of sarcasm detection based on the proposed method SPOZ-ESD.

6.1.2 Numerical Illustration of Proposed Method: SPOZ-ESD

To demonstrate the implementation of the proposed method, the two examples are taken from the defined datasets.

S1: "अन्य लोगों की समस्या सुनने के लिए अच्छा है .. क्योंकि आपको एहसास होगा कि आपकी समस्याएं उनकी तुलना में छोटी हैं"

S2: "वाह! कितना साफ कमरा है, बस कूड़ा डालने की ही कमी है।"

In accordance with step 1, first, we need to calculate the ISVs of ISS_5, ISS_6 ISS_7, and ISS_8 of S1 and S2 using *Algorithms 6.1*, *6.2*, *6.3*, and *6.4*. *Table 6.3* shows the calculation of the ISVs scores.

Table 6.3 ISV scores for S1 and S2

Sentences	ISS_5	ISS_6	ISS_7	ISS_8	e^{avg}
S1	*19*	*5.146*	*6.488*	*12.97*	*12.97*
S2	*11*	*2.840*	*3.521*	*6.336*	*5.924*

In step 2, we perform sentiment evaluation to get ϕ_5 using *Algorithm 6.5* and applying key-phrase extraction to get the triggered sarcasm score k_{max}^6. The computation of ϕ_5 and k_{max}^6 are shown in *Tables 6.4* and *6.5,* respectively.

Table 6.4 Numeric scores of ϕ_5 for S1 and S2

Sentences	***PSS***	***NSS***	ϕ_5	***Sentiment tag***
S1	*0.25*	*0.625*	*0.1875*	*Positive*
S2	*0.625*	*1.25*	*0.3125*	*Negative*

In the final step, we perform sarcasm detection using *Algorithm 6.6*. *Table 6.6* shows the computation of ψ_9 used for sarcasm detection. Based on the decision using *Algorithm 6.6*, we deduce that sentence S1 is a "*non-sarcastic*" sentence while S2 is detected as a "*sarcastic*" sentence.

Table 6.5 Computation of k_{max}^{6}

Sentences	***Translate to English***	***Triggered context***	$\boldsymbol{k_{max}^{6}}$
S1	*"Is good to hear other people's problem .. Because you will realise that your problems are small compare to them"*	*"problems are small compare to them"*	*4*
S2	*"Wow! What a clean room, it's just missing some trash to complete the look."*	*"missing some trash to complete the look"*	*4*

Table 6.6 Computation of ψ_9 score using SPOZ-ESD

Sentences	$\boldsymbol{e^{avg}}$	$\boldsymbol{\phi_5}$	$\boldsymbol{k_{max}^{6}}$	$\boldsymbol{\psi_1}$	***Tag***
S1	*12.97*	*0.1875*	*4*	0.9768	*Non-sarcasm*
S2	*5.924*	*0.3125*	*4*	0.4431	*Sarcasm*

6.2 Experimental and Result Analysis

In this section, we present the experimentation and results of the proposed SPOZ-ESD method. We begin by outlining the datasets used for the experiments. Following this, we evaluate the performance of SPOZ-ESD on the two datasets. A detailed macro and micro-level evaluation of the SPOZ-ESD method is then provided. Finally, we compare SPOZ-ESD with state-of-the-art techniques using standard evaluation metrics on both datasets.

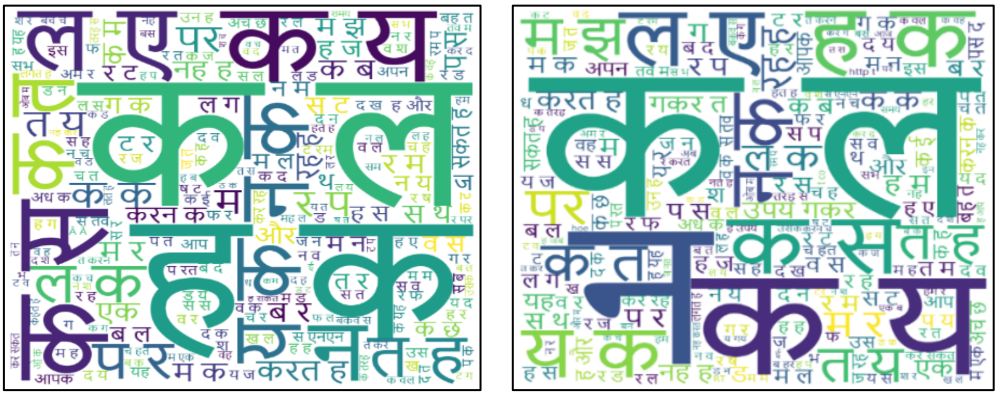
6.2.1 Dataset Description

The proposed method is tested on two benchmark datasets for sarcasm detection in Hindi text. Dataset 1 consisted of News Headlines [185], and the dataset-2 consisted of comments from Twitter [186], [187] in Hindi.

Table 6.7 Statistics of datasets used in the proposed method (SPOZ-ESD)

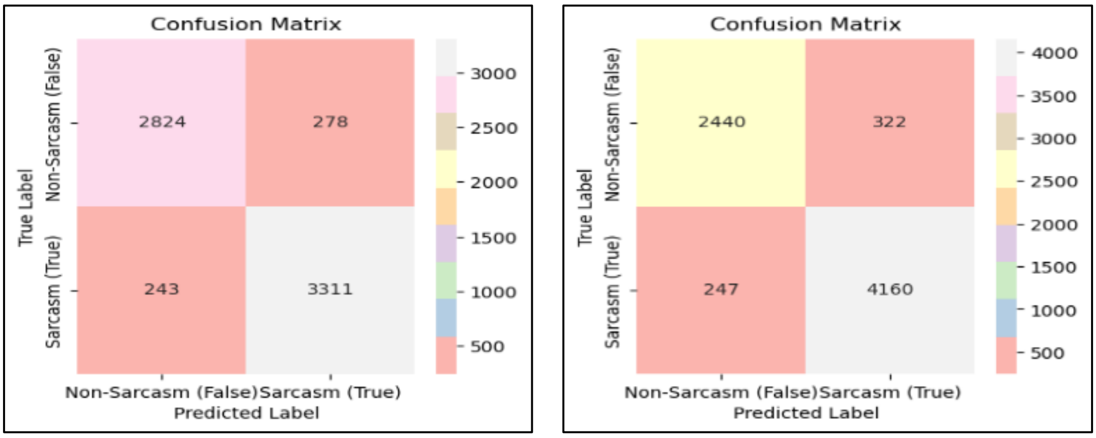
Datasets	**Sarcastic sentences**	**Non-sarcastic sentences**	**Total number of sentences**
Dataset-1 [185]	*3,554*	*3,102*	*6,656*
Dataset-2 [186] [187]	*4,407*	*2,762*	*7,169*

Table 6.7 gives the statistics of the datasets used in the proposed method. *Fig. 6.4 (a)* and *(b)* depict the word cloud associated with the two datasets. *Fig. 6.5 (a)* and *(b)* represent the confusion matrices of the two datasets.



(a). Word cloud for Dataset-1 (b). Word-cloud for Dataset-2

Fig. 6.4 Word cloud for (a) Dataset-1 and (b) Dataset-2



(a). Confusion matrix for Dataset-1 (b). Confusion matrix for Dataset-2

Fig. 6.5 Confusion matrices generated for Dataset-1 and Dataset-2

6.2.2 Performance Evaluation of the Proposed Method

The evaluation metrics utilized for performance evaluation were accuracy, precision, recall, F1 score, specificity, MCC, misclassification rate, false-positive rate, false-negative rate, and false discovery rate. The formulas for each of them are discussed in *Chapter 1, Section 1.7* of the "*Introduction*".

The accuracy scores achieved by the SPOZ-ESD are 0.9217 (dataset-1) and 0.9206 (dataset-2). The precision values are obtained as 0.9270 (dataset-1) and 0.9281 (dataset-2). The recall metric achieved scores of 0.9225 (dataset-1) and 0.9439 (dataset-2). The F1-score is computed as 0.93162 (dataset-1) and 0.9359 (dataset-2), respectively.

Meanwhile, the MCC was defined as a metric assessing the performance of learning methods in dual and multi-class categorization. It yielded values of 0.8426 (dataset-1) and 0.8397 (dataset-2). The false positive rate, indicated the ratio of false positive predictions, possessed values of 0.0905 (dataset-1) and 0.0958 (dataset-2). Whereas, the false negative reflected the incorrectly identifying negative assertions computed as 0.05711 (dataset-1) and 0.04962 (dataset-2).

The false discovery rate, measured as false positive predictions relative to true positives, also stands at 0.08709 (dataset-1) and 0.08746 (dataset-2). The misclassification rate calculated as 0.08020 (dataset-1) and 0.08567 (dataset-2) showing incorrectly classification. *Fig. 6.6* showed the performance of the SPOZ-ESD across two datasets in terms of defined metrics.

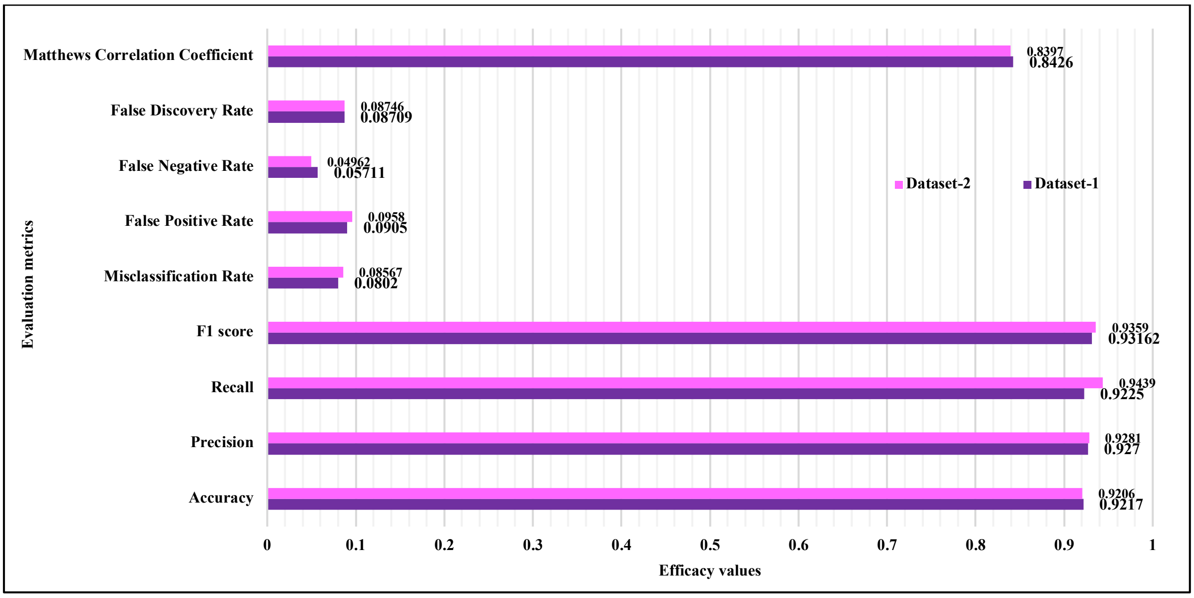


Fig. 6.6 Performance evaluation of SPOZ-ESD over two datasets

6.2.3 Macro and Micro Evaluation of the Proposed Method

Macro and micro statistics determine how a specific methodology functions with varied datasets. The mathematical formula of each of them has already been discussed in *Chapter 1*, *Section 1.7* of the "*Introduction*". The cumulative micro and macro precisions of the two datasets have been calculated as 0.9253 and 0.9275, respectively. The micro and macro recall scores across the two datasets have been computed as 0.9377 and 0.9332, respectively. The proposed paradigm's overall macro and micro evaluation on the two datasets is illustrated in *Fig. 6.7*.

The macro and micro F1 scores for the two datasets are calculated as 0.9337 and 0.9314, respectively. Scores close to 1.00 indicated consistent findings. These values demonstrated that the SPOZ-ESD produced consistent results with confirmed efficacy when applied to various datasets.

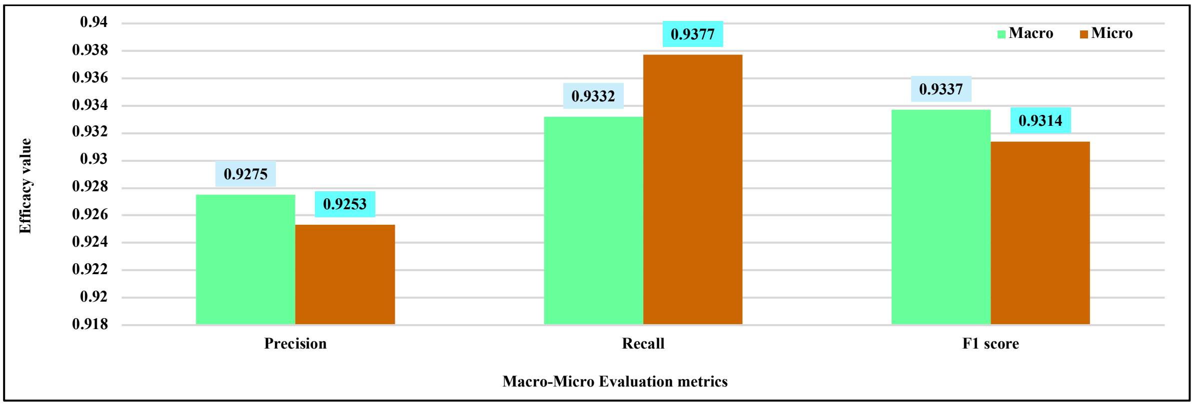


Fig. 6.7 Macro-micro evaluation of SPOZ-ESD on dataset-1 and dataset-2

6.2.4 Comparison Analysis of the Proposed Method with Existing State-of-the-Art Methods on Dataset-1

While comparing the performance of SPOZ-ESD with existing state-of-the-art methods, we observed that SPOZ-ESD achieved enhanced results in terms of accuracy, precision, recall, and F1 score for sarcasm detection. PBCT [10] and CNN-LSTM [78] Methods used embedded parsing with supervised learning but ignored the contextual indicator interpretation of sarcasm. The SPOZ-ESD accomplished a better accuracy rate, surpassing the PBCT [10] and CNN-LSTM [78] methods by 23.35% and 2.24%, respectively.

According to the experimental findings, the SPOZ-ESD performed better than the most robust method, TANA [100], by 5.12% in terms of recall. Additionally, SPOZ-ESD accomplished a better recall score, surpassing the CBPSD [101] by 15.15%. The SPOZ-ESD accomplished a better F1score by 23.41% and 19.63%, surpassing SVM [25] and POS-HMM [26] respectively. *Fig. 6.8* depicts the performance measure of SPOZ-ESD compared with other existing state-of-the-art methods on dataset-1.

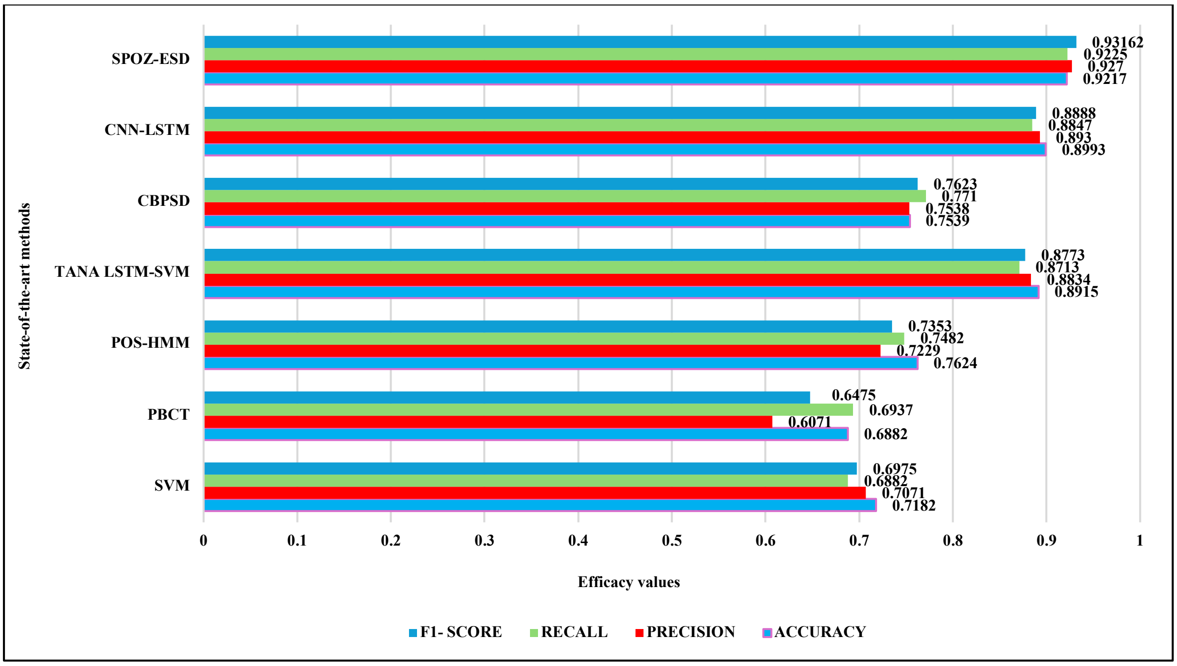


Fig. 6.8 Comparison analysis of SPOZ-ESD with existing state-of-the-art methods on dataset-1

6.2.5 Comparison Analysis of the Proposed Method with Existing State-of-the-Art Methods on Dataset-2

The SPOZ-ESD exhibited a 25.05% and 3.65% gain in accuracy score compared to PBCT [10] and CNN-LSTM [78], respectively. According to the experimental findings, the SPOZ-ESD performed better than the SVM [25] method by 20.92% in terms of the precision score. It performed poorly due to the inconsistent specific discrepancies with neglected sentence polarity. Additionally, the SPOZ-ESD accomplished a better recall rate, surpassing POS-HMM [26] and TANA [100] by 18.59% and 6.14%, respectively.

The decrement in the performance of these methods occurred due to ignorance of the utterance-level collaboration of language and knowledge within it. The SPOZ-ESD performed well in terms of F1 rating, showing an 18.32% increment in its performance compared to CBPSD [101]. *Fig. 6.9* depicts the performance measure of SPOZ-ESD compared with other existing state-of-the-art methods on dataset-2. The incremental performance of SPOZ-ESD in both dataset- 1 and 2 is due to the presence of the implicit intent expressed using a mathematical optimization technique.

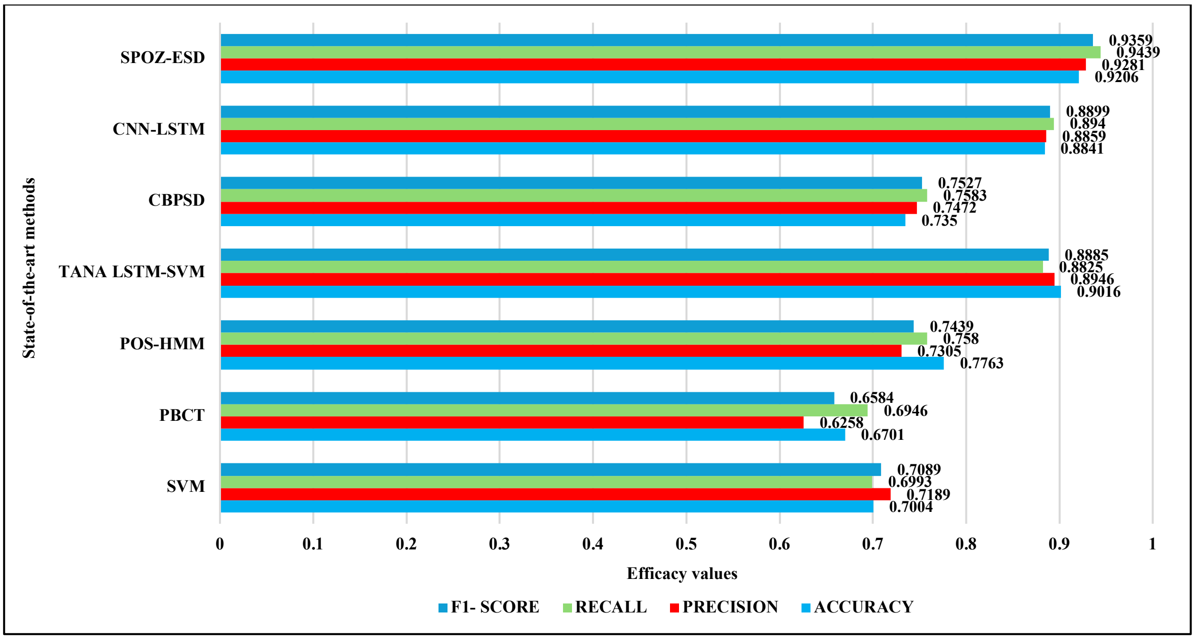
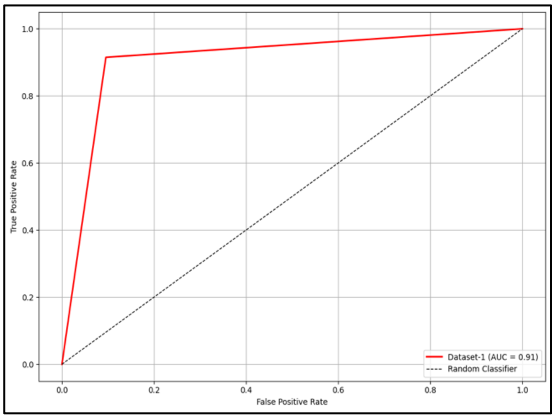
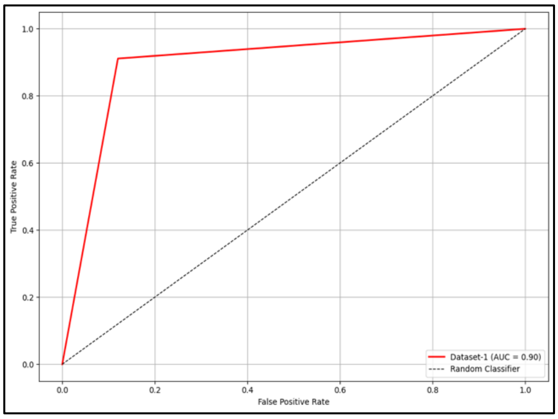


Fig. 6.9 Comparison analysis of SPOZ-ESD with existing state-of-the-art methods on dataset-2

The enhanced performance of the proposed method indicates that when contrasted with alternative approaches, i.e., the existing state-of-the-art methods, the SPOZ-ESD yielded better results. *Fig. 6.10 (a)* and *(b)* give the two datasets receiver operating characteristic (ROC) curves. The given curves aid in illustrating the relationship between false-positive and true-positive rates. It shows how well the SPOZ-ESD worked based on binary classification across various limits, with good results.



(a) ROC curve of dataset-1



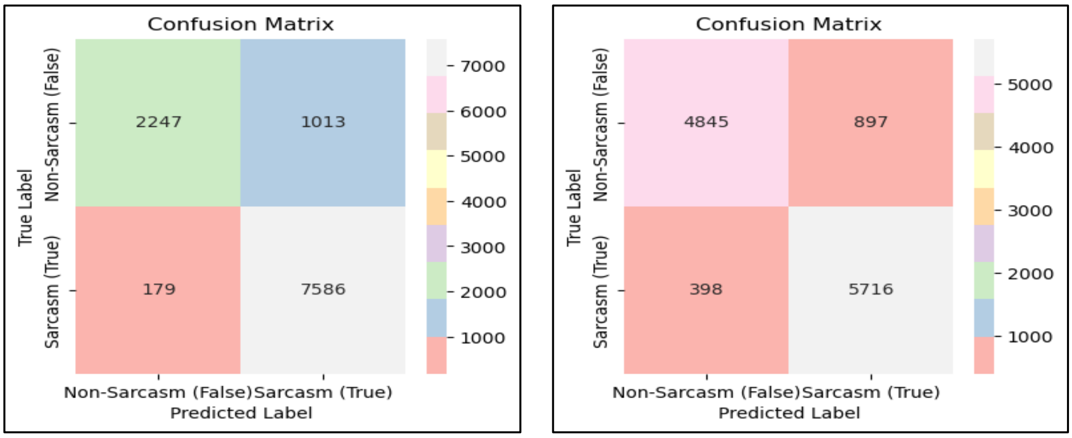
(b) ROC curve of dataset-2

Fig. 6.10 ROC curve for (a) dataset-1 and (b) dataset-2

Additionally, the proposed (SPOZ-ESD) method has implemented on two English datasets, i.e., semeval-2017 [188] and SARC [189]. These datasets were utilized to validate the performance of the proposed method for sarcasm detection in English. *Table 6.8* shows the results obtained for SPOZ-ESD on the two English datasets. *Fig. 6.11 (a)* and *(b)* show the confusion matrix generated for the two English datasets.

Table 6.8 Results of SPOZ-ESD implemented on English datasets

Datasets	Accuracy	Precision	Recall	F1 score
Semeval-2017 [188]	*0.8907*	*0.8643*	*0.9349*	*0.8982*
SARC [189]	*0.8918*	*0.8821*	*0.9769*	*0.9271*



(a) Confusion matrix for Semeval-2017 (b) Confusion matrix for SARC

Fig. 6.11 Confusion matrices generated for Semeval-2017 and SARC datasets

6.3 Discussion

In this section, we present a comprehensive validation of the SPOZ-ESD method through four hypothesis tests. We also conducted an ablation study to evaluate the individual contributions of method components. Additionally, a case study is included to demonstrate the practical applicability of SPOZ-ESD. The computational cost analysis of the proposed method is also discussed in this section. Finally, we outline the key challenges encountered during the implementation and evaluation phases.

6.3.1 Statistical Validation of the Proposed Method Using the z-Test

A Z-test is performed to compare the proportions of two populations, one from the SARC dataset and the other from Twitter comments. Two distinct samples are extracted from these datasets, as shown in *Table 6.9*. The test analyzed the proportions under the null hypothesis, summarizing the results in *Table 6.9*.

Table 6.9 Two Proportion Z-Test statistics across two samples

	SPOZ-ESD	
Parameters	*Sample 1*	*Sample 2*
Sample size (N)	*999*	*573*
Sample proportion (c)	*0.8018*	*0.8062*
Favorable cases (X)	*801*	*462*
P	*0.8034*	
z	*0.2113*	
Hypothesis status	H_0 *is not rejected*	

For two population proportions (c_1 and c_2), null hypotheses (H_0) and alternative hypotheses (H_a) are defined as follows:

H_0: $c_1 = c_2$, i.e., the accuracy score of sample 1 = accuracy score of sample 2

H_a: $c_1 \neq c_2$, i.e., the accuracy score of sample 1 $\neq$ accuracy score of sample 2

In the proposed method, the condition of the null hypothesis is accepted when the critical range z-value ($\alpha = 0.05$) is greater than the z-statistic value. The p-value is calculated using *Eqn. (3.15)* (defined in *Chapter 3*) computed as 0.8034. Now, the z-statistic value is computed using *Eqn. (3.16)* obtained as 0.2113. This showed that the calculated z-statistic is less than the critical region z-value at $\alpha = 0.05$ which claimed that there is less evidence to reject the null hypothesis (H_0). In *Fig. 6.12,* the critical region of the hypothesis is depicted graphically. This finding suggested that the consistency of the proposed method's accuracy holds across diverse sample sizes.

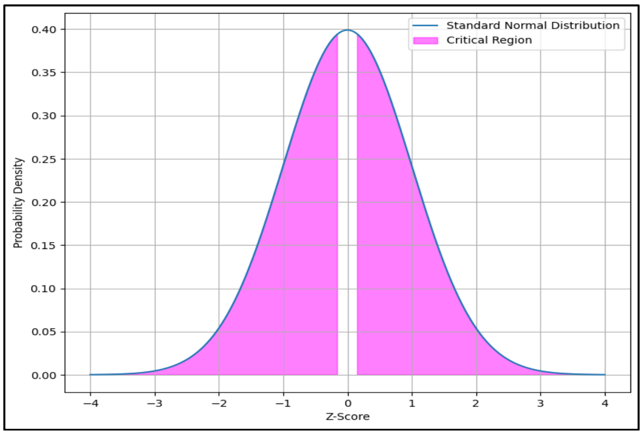


Fig. 6.12 Statistical validation of the SPOZ-ESD using the z-test

6.3.2 Statistical Validation of the Proposed Method Using the Kolmogorov-Smirnov Test

In the Kolmogorov-Smirnov (KS) test, we compared two independent samples taken from the two datasets. The aim is to determine whether the two samples have similar distributions without assuming any change in the distribution based on ECDF. *Table 6.10* shows the working of this test.

Table 6.10 Two Proportion KS Test statistics across two samples

	SPOZ-ESD	
Parameters	*Sample 1*	*Sample 2*
Sample size (N)	*1133*	*496*
Sample proportion (c, ECDF)	*0.8923*	*0.8951*
Favorable cases (X)	*1011*	*444*
D- value (D_{N_1, N_2})	*0.8889*	
D_α	*0.8901*	
Hypothesis status	H_0 *is not rejected*	

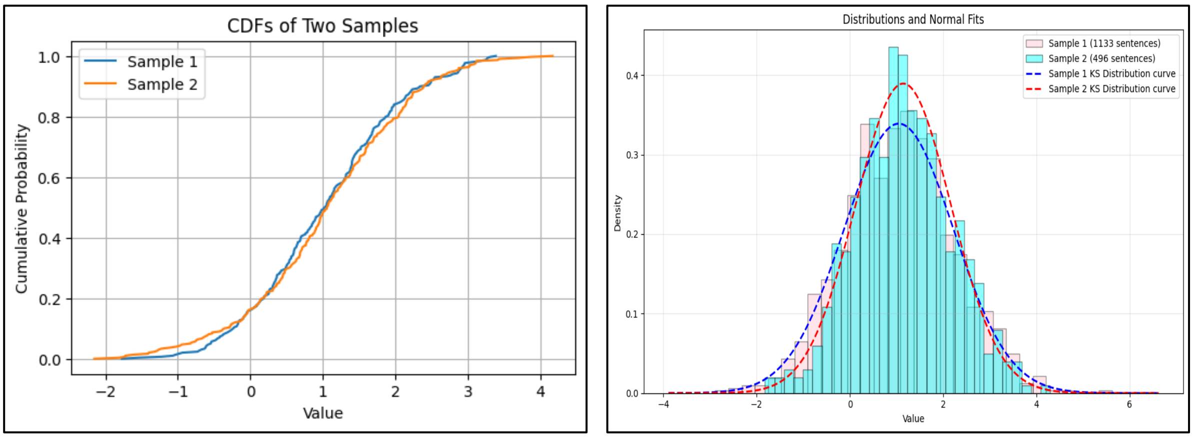
For two population proportions (c_1 and c_2), null hypotheses (*H_0*) and alternative hypotheses (*H_a*) are defined as follows:

$$H_0: c_1 = c_2, \text{ i.e., accuracy of sample 1} = \text{accuracy of sample 2}$$

$$H_\alpha: c_1 \neq c_2, \text{ i.e., the accuracy of sample 1} \neq \text{accuracy of sample 2}$$

According to the Kolmogorov-Smirnov test for two samples, the condition to be satisfied, when D-value (D_{N_1, N_2}) must be less than D_α, i.e., $D_{N_1, N_2} < D_\alpha$. If this condition is followed, the two samples would accept the given null hypothesis. We compute the D_{N_1, N_2}, using *Eqn. (3.17)* (already defined in *Chapter 3*) obtained as 0.8889. Computing the D_α value using *Eqn. (3.18)* (already defined in *Chapter 3*) obtained as 0.8901.

This demonstrated that the given null hypothesis can be accepted without any failure under the given conditions of the Kolmogorov-Smirnov test. The final value in the KS test showed that the given null hypothesis has been accepted. *Fig. 6.13 (a)* represents the cumulative distribution curve for two samples meeting at the D_α point. *Fig. 6.13 (b)* depicts the given D_α value calculated as 0.8901 where the two curves' peaks meet.



(a). Cumulative distribution frequency curve for two samples

(b). Graphical representation using the Kolmogorov-Smirnov test

Fig. 6.13 Statistical validation of the SPOZ-ESD using the Kolmogorov-Smirnov test

This implied that the two samples have identical accuracy scores in terms of distribution, i.e., having a consistent distribution for varied datasets.

6.3.3 Statistical Validation of the Proposed Method Using the Mann-Whitney U Test

The Mann-Whitney U test has been used to determine whether or not two distinct samples originating from different populations had the same distribution. Based on the provided sample data, the computed mean and standard deviation of the number of categorized sentences in each sample are shown in *Table 6.11*.

For two population proportions (c_1 and c_2), null hypotheses (H_0) and alternative hypotheses (H_a) are defined as follows:

H_0: $c_1 = c_2$, i.e., the accuracy distribution of sample 1 = accuracy distribution of sample 2

H_a: $c_1 \neq c_2$, i.e., the accuracy distribution of sample 1 ≠ accuracy distribution of sample 2

In *Fig. 6.12,* the critical region of the hypothesis is depicted graphically. This finding suggested that the consistency of the proposed method's accuracy holds across diverse sample sizes. The mean and standard deviation of the samples are obtained as 0.3087 and 0.6001 using *Eqns. (3.20)* and (3.21), respectively. The mean and standard deviation of the samples are obtained as 0.3087 and 0.6001 using *Eqns. (3.20)* and *(3.21),* respectively.

Therefore, the evidence does not support the claim that the population proportion c_1 significantly differs from c_2 at $\alpha = 0.05$. *Fig. 6.14* shows the critical region of the hypothesis z_B value depicted graphically.

Table 6.11 Two Proportion Mann-Whitney U Test statistics across two samples

	SPOZ-ESD	
Parameters	*Sample 1*	*Sample 2*
Sample size (N)	*1556*	*847*
Sample proportion (c)	*0.7853*	*0.7863*
Favorable cases (X)	*1222*	*666*
U- value	*0.7853*	
$\boldsymbol{mean_{m_1,m_2}}$	*0.3087*	
$\boldsymbol{Std_{m_1,m_2}}$	*0.6001*	
$\boldsymbol{z_B}$	*0.7941*	
Hypothesis status	H_0 *is not rejected*	

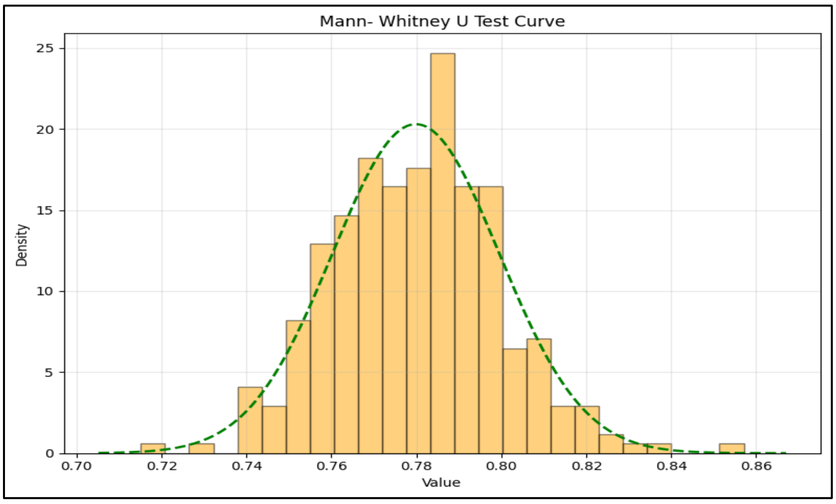


Fig. 6.14 Statistical validation of SPOZ-ESD using Mann- Mann-Whitney U test

This finding showed that the SPOZ-ESD holds a firm accuracy across diverse sample sizes derived from various datasets.

6.3.4 Statistical Validation of SPOZ-ESD using Kruskal-Wallis Test (H-Test)

A non-parametric statistical procedure called the Kruskal-Wallis test has been employed to determine if the medians of three or more distinct groups are considered in a meaningful way.

It is shown with an expansion of the t-test for distinct samples beyond two sets using a non-parametric substitute. We conducted this test using random sentences from the two datasets used in the proposed method, as shown in *Table 6.12*.

Table 6.12 Two Proportion Kruskal-Wallis H-Test statistics across three samples

	SPOZ-ESD		
Parameters	*Sample 1*	*Sample 2*	*Sample 3*
Sample size (N)	*1442*	*936*	*588*
Sample proportion rank (R)	*0.8550*	*0.8547*	*0.8503*
Favorable cases (X)	*1233*	*800*	*500*
$\boldsymbol{H_{N_1,N_2,N_3}}$	*0.8356*		
$\boldsymbol{\chi^2_{0.05}}$	*5.99*		
Hypothesis status	H_0 *is not rejected*		

For three population proportions, null hypotheses (H_0) and alternative hypotheses (H_a) have defined as follows:

H_0: accuracies of sample 1, sample 2, and sample 3 are equivalent

H_a: The accuracies of sample 1, sample 2, and sample 3 are not equivalent

According to the Kruskal-Wallis test, if the samples followed the condition $H_{N_1,N_2,N_3} < \chi^2_\alpha$ (at significance level $\alpha = 0.05$), the null hypothesis is accepted. We computed H_{N_1,N_2,N_3} values using *Eqn. (6.11)*. The value is computed as 0.8356. Here, N denotes the total number of samples taken in the observation, k is the number of groups, and R_i considers the sum of the ranks of the i^{th} group. The total number of samples is computed as $N = N_1 + N_2 + N_3$.

$$H_{N_1,N_2,N_3} = \frac{12}{N\times(N+1)}\left[\frac{R_1^2}{N_1} + \frac{R_2^2}{N_2} + \frac{R_3^2}{N_3} - 3\times(N+1)\right] \tag{6.11}$$

In the case of the SPOZ-ESD, the null hypothesis is formulated as having a consistent accuracy score, and vice versa for the alternative hypothesis.

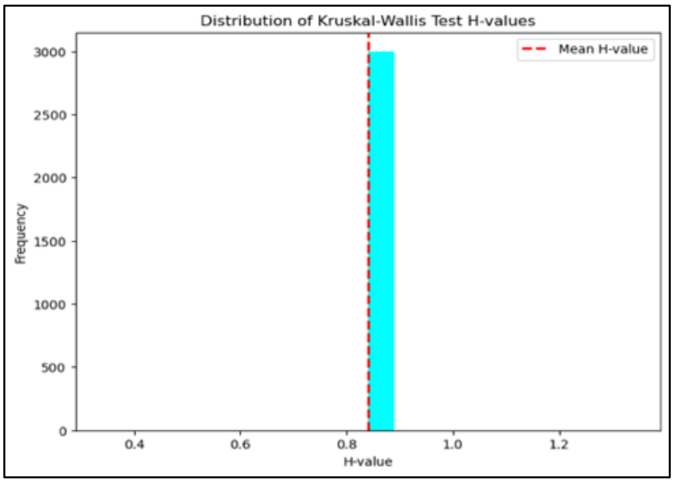


Fig. 6.15 Statistical validation of SPOZ-ESD using the Kruskal-Wallis test

6.3.5 Ablation Study

An ablation study is conducted to evaluate the influence of the sarcasm detection threshold (τ_{10}) on the performance of the proposed SPOZ-ESD method. By analysing the method's behaviour with $\tau_{10} < 0.5, \tau_{10} = 0.5$, and $\tau_{10} > 0.5$ across the two datasets, the results revealed that $\tau_{10} = 0.5$ consistently outperformed other values, yielding the highest accuracy, precision, recall, and F1-score.

Specifically, $\tau_{10} < 0.5$ led to lower recall and accuracy due to increased false negatives, where sarcastic content is misclassified as non-sarcastic. Conversely, $\tau_{10} > 0.5$ caused more false positives, misidentifying non-sarcastic as sarcastic, which lowered precision and accuracy. These results confirmed that $\tau_{10} = 0.5$ provides a balanced threshold that minimizes both false positives and false negatives, ensuring optimal sarcasm detection in SPOZ-ESD. *Table 6.13* shows the results of the ablation study.

Table 6.13 Ablation study of SPOZ-ESD

Dataset	Removed module	Accuracy	Precision	Recall	F1- score
Dataset-1	$\tau_{10} < 0.5$	*0.7306*	*0.8977*	*0.5008*	*0.6429*
	$\tau_{10} > 0.5$	*0.8149*	*0.6891*	*0.9144*	*0.7859*
	SPOZ-ESD with $\boldsymbol{\tau_{10} = 0.5}$	***0.9217***	***0.9270***	***0.9225***	***0.9316***
Dataset-2	$\tau_{10} < 0.5$	*0.6952*	*0.8624*	*0.5256*	*0.6531*
	$\tau_{10} > 0.5$	*0.8576*	*0.7152*	*0.9319*	*0.8092*
	SPOZ-ESD with $\boldsymbol{\tau_{10} = 0.5}$	***0.9206***	***0.9281***	***0.9439***	***0.9359***

6.3.6 Case Study

A case study is conducted to compare the proposed SPOZ-ESD method with the existing state-of-the-art CNN-LSTM method. SPOZ-ESD accurately identified sarcasm through the recognition of implicit expression features, unlike the CNN-LSTM method. The proposed method effectively captured sentiment cues and tonal indicators linked to sarcastic intent. This study demonstrated that sarcasm detection is significantly influenced by the semantic relatedness component.

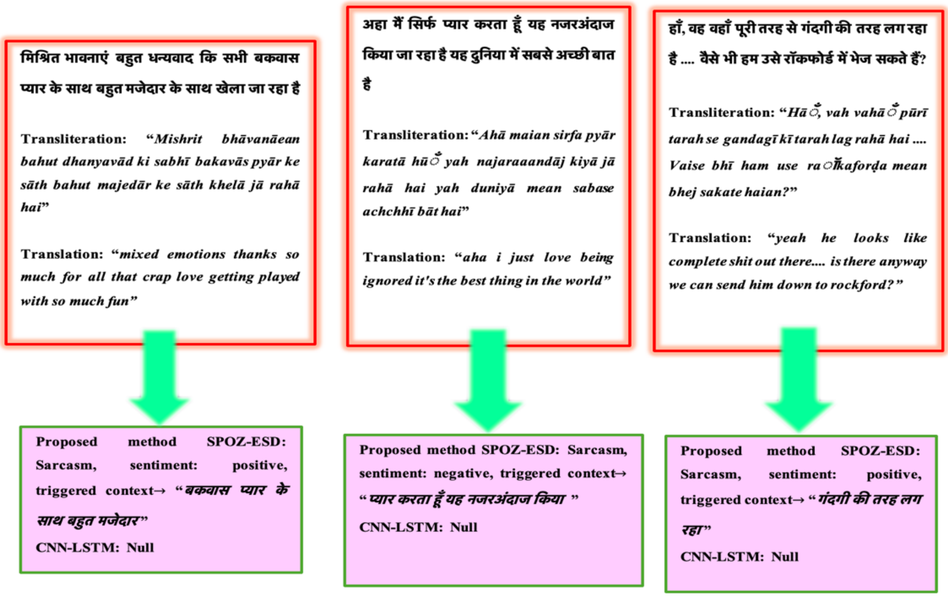


Fig. 6.16 Case study visualisation of sample sentences showing better results by SPOZ-ESD in comparison to CNN-LSTM

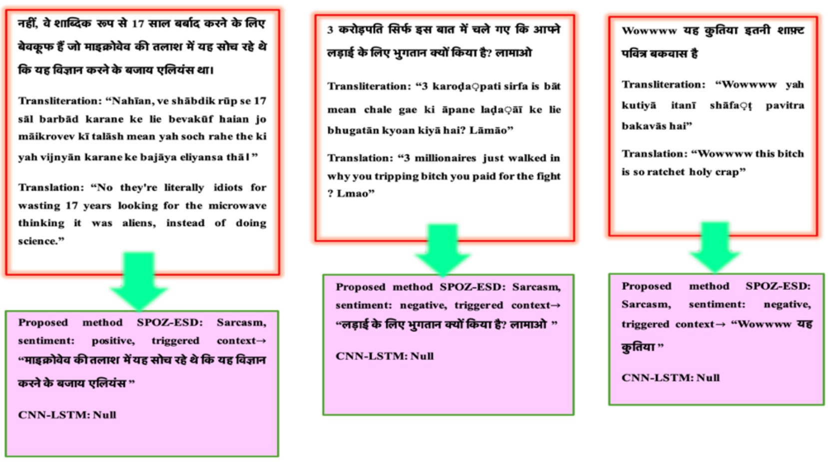


Fig. 6.17 Case study visualization of sample sentences showing better results by SPOZ-ESD in comparison to CNN-LSTM

SPOZ-ESD successfully extracted the sarcasm triggering terms, enhancing recognition of sarcastic signals. This precise extraction supported accurate sarcasm detection. Additionally, SPOZ-ESD preserved psychological cues within sentence structures, aiding in tonal interpretation. It also shows improved contrastive retention by integrating language-based learning features, as illustrated in *Figs. 6.16* and *6.17.*

6.3.8 Challenges of the Proposed Method

A few challenges contributing to the lower performance of SPOZ-ESD are outlined below. The first challenge involves a lack of ethnic and contextual uniqueness in the method. SPOZ-ESD struggles to accurately classify sentences that reflect strong cultural or situational nuances. The absence of such contextual understanding limits the method's ability to interpret sarcasm effectively.

The second challenge is the presence of veiled or polite sarcasm. Sarcasm expressed under the guise of respect is often subtle and difficult to detect. Identifying refined forms of sarcasm requires deeper pragmatic and tonal understanding beyond the current method's capabilities. *Table 6.14* gives the elaborated example in support of this context.

Table 6.14 Sample sentences where the proposed method failed to correctly detect the sarcasm

Sentences	**Proposed Method Tag**	**Actual Tag**
"*वाह! तुमने तो एक ही महीने में शादी, ट्रैक्टर और जमींन सब ले ली— बड़ा ही सादा जीवन है तुम्हारा।*"	Non-sarcasm	Sarcasm
"*आप तो हमेशा समय पर आते हैं आज सिर्फ दो घंटे लेट हुए!*"	Non-sarcasm	Sarcasm

6.4 Summary

This chapter introduced a unique method to learn sentiment cues from sarcastic sentences through optimization-based learning in Hindi text. To the best of our knowledge, it is the first approach to specifically address the tonal nature of sarcasm indicators in Hindi using mathematical modelling. The SPOZ-ESD method captures the nuanced tonal expressions more effectively than existing approaches. The use of a mathematical optimization technique for detecting sarcasm in Hindi text offers a novel and promising direction.

The proposed SPOZ-ESD achieved accuracy scores of 0.9217 and 0.9206 on Dataset-1 and Dataset-2, respectively. It also recorded precision scores of 0.9270 and 0.9281, and recall scores of 0.9225 and 0.9439 on the two datasets. The corresponding F1 scores were 0.9316 and 0.9359, indicating strong performance. These results confirm the method's capability to detect sarcasm with high precision and robustness in Hindi language context.

Chapter 7

Conclusions, Contributions, and Future Work

This chapter brings together the major findings and outcomes of the research work presented in this thesis. The study has been structured across six chapters that introduce, implement, and evaluate various statistical and computational methods to identify sarcasm in textual data. The methods included individual probability distribution-based approaches, hybrid optimization techniques, categorization of sarcasm types, and language-specific modelling.

In *Chapter 3*, we proposed sarcasm detection models using statistical distributions such as the Bessel function, Box-Cox transformation, Cauchy distribution, Logistic distribution, and Weibull distribution as the core functions within optimization frameworks. *Chapter 4* expanded on this by integrating hybrid optimization methods, aiming for better performance in more complex scenarios. *Chapter 5* addressed a novel aspect: detecting and classifying types of sarcasm, which had not been widely explored in previous research. In *Chapter 6*, we implemented sarcasm detection for Hindi language text, presenting a language-specific model that is both innovative and culturally inclusive.

This final chapter summarizes the conclusions drawn from each of these methods, presents the major contributions made through the research, and identifies promising directions for future studies in this domain.

7.1 Conclusions

The proposed models demonstrated that statistical and distribution-based optimization approaches are efficient but also effective in accurately identifying sarcasm across different types of data and languages. This section provides the final thoughts and conclusions of the research work carried out in this thesis. The aim of this conclusion is to bring together all the major insights drawn from the research and to highlight their overall significance. The thesis presents several key findings, which are outlined below:

- This research successfully demonstrated that optimization-based techniques, when combined with statistical distributions, offer a powerful way to detect sarcasm in text data. By treating sarcasm detection as an optimization problem, the models could capture subtle patterns of word usage, sentiment inconsistency, and contextual clues.

- The use of different probability distributions (such as Cauchy, Logistic, Weibull, Box-Cox, and Bessel) proposed in Chapter 3 allowed us to explore how mathematical behaviors of data utilized to frame the sarcasm detection task in a performance-enhancing way. These distributions served as the objective functions for maximization or minimization under specific constraints to get optimal sarcasm score.
- Hybrid optimization methods proposed in Chapter 4 further improved detection accuracy by combining the strengths of multiple algorithms. These hybrid models outperformed basic optimization techniques in handling real-world, noisy social media data.
- Chapter 5 contributed a new perspective by going beyond binary sarcasm classification as categorizing different types of sarcasm, such as callous, humour, offensive, humblebrag, and oxymoron sarcasms. This enriched the understanding of sarcasm as a multi-dimensional linguistic feature.
- In Chapter 6, the work has been extended to Hindi texts, proving that the optimization-based models are not language-dependent and can be customized for regional and under-resourced languages. This addressed a major gap in sarcasm detection for non-English contexts.

The results of this study confirm that optimization methods using mathematical distributions can successfully identify and categorize sarcasm. The developed models showed competitive performance compared to existing deep learning-based approaches, while offering better scalability and reduced computational requirements. This work lays a strong foundation for future research and practical deployment of sarcasm detection systems across diverse applications like social media monitoring, customer feedback analysis, and sentiment tracking.

7.2 Contributions

This section highlights the major contributions of this research work. Throughout the thesis, several new methodologies have been proposed using statistical distributions and optimization techniques to improve sarcasm detection. These contributions not only add value to the field of NLP but also introduce practical frameworks that can be extended to multiple domains and languages.

This thesis presents a novel mathematical optimization-driven framework for sarcasm detection. Within this framework, linguistic cues such as polarity inconsistencies and context-dependent trigger expressions are mapped to quantitative features. These feature values are embedded within a minimised probabilistic objective function define through different statistical distribution functions. Each distribution contributes a distinct tail behaviour, enabling the model to capture varied patterns of linguistic deviation that characterize sarcastic expression. Solving this optimization problem yields an optimal sarcasm score for each text instance.

Entropy measures are further employed to represent uncertainty and information divergence between literal statements and their implied meanings. It helps in capturing the predictability of linguistic patterns within the text. The associated optimization procedures operate under well-defined constraints that regulate threshold boundaries, maintain numerical stability, and delineate feasible regions for optimal sarcasm scores and sarcasm-type indicators.

Collectively, these elements establish a rigorous, unsupervised mathematical framework for deriving optimal sarcasm and sarcasm-type methods across both English and Hindi datasets.

The following points summarize the novel elements developed and implemented during the research.

- **Novel Optimization Frameworks:** Introduced different novel sarcasm detection models where each used a different statistical distribution as the core function (Bessel, Box-Cox, Logistic, Weibull, and Cauchy). These models applied constrained optimization to identify optimal sarcasm scores based on sentiment and linguistic features.
- **Hybrid Optimization Approaches:** Developed new models combining multiple optimization algorithms to achieve higher detection performance, such as Logistic ion motion optimization approaches. These hybrids offered better stability and convergence rates.
- **Categorization for Types of Sarcasm:** Created a novel framework for identifying not just whether a text is sarcastic, but also what kind of sarcasm it represents. This includes detecting subtle sarcasm like humble-bragging or aggressive irony, thus adding semantic depth to analysis.

- **Hindi Sarcasm Detection Model:** Designed a domain-specific sarcasm detection model for the Hindi language using optimization techniques. This filled a critical research gap for sarcasm analysis in regional languages and expanded the model's applicability beyond English.
- **Evaluation and Validation:** Conducted thorough experiments using publicly available datasets such as SARC, Twitter, Semeval, and News Headlines. Employed performance metrics like accuracy, precision, recall, F1-score, and conducted statistical validation using z-tests and other non-parametric tests.
- **Scalable Framework:** Demonstrated that all proposed methods are computationally efficient and can be scaled to large datasets, making them suitable for real-time sarcasm detection in social media applications.

All these contributions reflect a meaningful step forward in the field of sarcasm detection. By integrating mathematical optimization with linguistic analysis, this research has expanded the boundaries of what traditional sarcasm detection systems can achieve.

The work bridges the gap between statistical modelling and machine learning, offering flexible, efficient, and interpretable solutions. These developments can serve as a useful base for further academic exploration and real-world implementation in AI systems.

7.3 Future work

While this thesis presents strong results and methods for sarcasm detection, there are several directions in which the work can be extended in the future. This section outlines potential areas where further research can be done to enhance the current models and expand their applicability. The aim is to improve the system's performance across different languages, platforms, and data types, while also addressing the limitations observed during the study.

- **Multimodal Sarcasm Detection:** Future research can explore sarcasm detection using multimodal data (text, audio, video). Integrating facial expressions, tone of voice, and body language along with text can lead to more accurate sarcasm interpretation, especially in video-based content like vlogs and interviews.
- **Cross-Lingual Models:** While this thesis addressed Hindi, future work could extend sarcasm detection to other regional and global languages (e.g., Tamil, Bengali, Arabic,

Spanish). Developing cross-lingual sarcasm models could facilitate inclusive and multilingual sarcasm detection systems.

- **Explainable AI:** Implementing explainable components into the models will help users and researchers understand why a statement was flagged as sarcastic. This is especially important for building trust in systems used in customer service, sentiment analysis, and online moderation.
- **Integration in Chatbots and Assistants:** The models can be embedded into intelligent systems like chatbots, virtual assistants, or content moderators, enabling them to understand and respond appropriately to sarcastic remarks in real time.
- **Low-Resource and Few-Shot Learning:** Developing few-shot or zero-shot learning frameworks can help in adapting sarcasm detection models to new domains or languages with limited annotated data.
- **Real-Time Detection Engines:** The optimization models developed can be implemented into live platforms such as Twitter sentiment dashboards or content moderation tools for real-time sarcasm detection, which is increasingly important in today's fast-paced online environments.

Moving forward, there is great potential to combine these optimization-based models with transformer architectures like BERT or GPT for hybrid performance. Further research can also explore multilingual sarcasm detection, real-time sarcasm tracking, and multimodal sarcasm analysis involving audio and video cues. With the growing importance of emotional intelligence in AI systems, the proposed models can play a vital role in building smarter, context-aware machines that better understand human communication.

References

[1] S. Poria, E. Cambria, D. Hazarika, and P. Vij, “A deeper look into sarcastic tweets using deep convolutional neural networks,” *arXiv preprint arXiv:1610.08815*, 2016.

[2] T. Yue, R. Mao, H. Wang, Z. Hu, and E. Cambria, “KnowleNet: Knowledge fusion network for multimodal sarcasm detection,” *Information Fusion*, vol. 100, pp. 101921, 2023.

[3] R. Pandey and J. P. Singh, “BERT-LSTM model for sarcasm detection in code-mixed social media post,” *J Intell Inf Syst*, vol. 60, no. 1, pp. 235–254, 2023.

[4] O. Vitman, Y. Kostiuk, G. Sidorov, and A. Gelbukh, “Sarcasm detection framework using context, emotion and sentiment features,” *Expert Syst Appl*, vol. 234, pp. 121068, 2023.

[5] A. Ghosh and T. Veale, “Fracking sarcasm using neural network,” in *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pp. 161–169, 2016.

[6] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep learning--based text classification: a comprehensive review,” *ACM computing surveys (CSUR)*, vol. 54, no. 3, pp. 1–40, 2021.

[7] L. Wu *et al.*, “Graph neural networks for natural language processing: A survey,” *Foundations and Trends® in Machine Learning*, vol. 16, no. 2, pp. 119–328, 2023.

[8] X. Gong, Q. Zhao, J. Zhang, R. Mao, and R. Xu, “The design and construction of a Chinese sarcasm dataset,” in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 5034–5039, 2020.

[9] O. Tsur, D. Davidov, and A. Rappoport, “ICWSM—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews,” in *fourth international AAAI conference on weblogs and social media*, vol. 4, no. 1, 2010.

[10] S. K. Bharti, K. S. Babu, and S. K. Jena, “Parsing-based sarcasm sentiment recognition in twitter data,” in *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, IEEE, pp. 1373–1380, 2015.

[11] J. Meng, Y. Zhu, S. Sun, and D. Zhao, "Sarcasm detection based on BERT and attention mechanism," *Multimed Tools Appl*, vol. 83, pp. 1–20, 2023.

[12] M. A. Galal, A. H. Yousef, H. H. Zayed, and W. Medhat, "Arabic sarcasm detection: An enhanced fine-tuned language model approach," *Ain Shams Engineering Journal*, vol. 15, no. 6, pp. 102736, 2024.

[13] M. E. Hassan, M. Hussain, I. Maab, U. Habib, M. A. Khan, and A. Masood, "Detection of Sarcasm in Urdu Tweets using Deep Learning and Transformer based Hybrid Approaches," *IEEE Access*, vol. 12, pp. 61542-61555, 2024.

[14] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional MKL based multimodal emotion recognition and sentiment analysis," in *2016 IEEE 16th international conference on data mining (ICDM)*, IEEE, pp. 439–448, 2016.

[15] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, and A. Gelbukh, "Sentiment and sarcasm classification with multitask learning," *IEEE Intell Syst*, vol. 34, no. 3, pp. 38–43, 2019.

[16] A. Kumar and V. H. C. Albuquerque, "Sentiment analysis using XLM-R transformer and zero-shot transfer learning on resource-poor indian language," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, pp. 1–13, 2021.

[17] D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words," in *proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1003–1012, 2015.

[18] H. Pan, Z. Lin, P. Fu, and W. Wang, "Modeling the incongruity between sentence snippets for sarcasm detection," in *ECAI 2020*, IOS Press, vol. 325, pp. 2132–2139, 2020.

[19] S. Das, S. Ghosh, A. K. Kolya, and A. Ekbal, "Un paralleled sarcasm: a framework of parallel deep LSTMs with cross activation functions towards detection and generation of sarcastic statements," *Lang Resour Eval*, vol. 57, no. 2, pp. 765–802, 2023.

[20] A. Rahma, S. S. Azab, and A. Mohammed, "A Comprehensive Review on Arabic Sarcasm Detection: Approaches, Challenges and Future Trends," *IEEE Access*, vol. 11, pp. 18261-18280, 2023.

[21] M. S. Akhtar, A. Ekbal, and E. Cambria, “How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes],” *IEEE Comput Intell Mag*, vol. 15, no. 1, pp. 64–75, 2020.

[22] A. Kumar and G. Garg, “Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets,” *J Ambient Intell Humaniz Comput*, vol. 14, no. 5, pp. 5327–5342, 2023.

[23] A. Kumar, S. Dikshit, and V. H. C. Albuquerque, “Explainable artificial intelligence for sarcasm detection in dialogues,” *Wirel Commun Mob Comput*, vol. 2021, pp. 1–13, 2021.

[24] J. Lemmens, B. Burtenshaw, E. Lotfi, I. Markov, and W. Daelemans, “Sarcasm detection using an ensemble approach,” in *proceedings of the second workshop on figurative language processing*, pp. 264–269, 2020.

[25] N. Desai and A. D. Dave, “Sarcasm detection in hindi sentences using support vector machine,” *International Journal*, vol. 4, no. 7, pp. 8–15, 2016.

[26] S. K. Bharti, K. Sathya Babu, and S. K. Jena, “Harnessing online news for sarcasm detection in hindi tweets,” in *International Conference on Pattern Recognition and Machine Intelligence*, Springer, vol. 10597, pp. 679–686, 2017.

[27] T. Xiong, P. Zhang, H. Zhu, and Y. Yang, “Sarcasm detection with self-matching networks and low-rank bilinear pooling,” in *The world wide web conference*, pp. 2115–2124, 2019.

[28] P. Liu, W. Chen, G. Ou, T. Wang, D. Yang, and K. Lei, “Sarcasm detection in social media based on imbalanced classification,” in *Web-Age Information Management: 15th International Conference, WAIM 2014, Macau, China, June 16-18, 2014. Proceedings 15*, Springer, pp. 459–471, 2014.

[29] A. Rajadesingan, R. Zafarani, and H. Liu, “Sarcasm detection on twitter: A behavioral modeling approach,” in *Proceedings of the eighth ACM international conference on web search and data mining*, pp. 97–106, 2015.

[30] A.-C. Băroiu and Ștefan Trăușan-Matu, “Automatic Sarcasm Detection: Systematic Literature Review,” *Information*, vol. 13, no. 8, pp. 399, 2022.

[31] Z. Wen *et al.*, “Sememe knowledge and auxiliary information enhanced approach for sarcasm detection,” *Inf Process Manag*, vol. 59, no. 3, pp. 102883, 2022.

[32] A. Y. Muaad *et al.*, “Artificial intelligence-based approach for misogyny and sarcasm detection from Arabic texts,” *Comput Intell Neurosci*, vol. 2022, 2022.

[33] K. Dave, S. Lawrence, and D. M. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews,” in *Proceedings of the 12th international conference on World Wide Web*, pp. 519–528, 2003.

[34] S. Chatterjee, S. Bhattacharjee, K. Ghosh, A. K. Das, and S. Banerjee, “Class-biased sarcasm detection using BiLSTM variational autoencoder-based synthetic oversampling,” *Soft comput*, vol. 27, no. 9, pp. 5603–5620, 2023.

[35] D. Vinoth and P. Prabhavathy, “An intelligent machine learning-based sarcasm detection and classification model on social networks,” *J Supercomput*, vol. 78, no. 8, pp. 10575–10594, 2022.

[36] L. Huang and A. D. Galinsky, “Was That Sarcastic or Supportive? Why Receiving Sarcasm Improves Perspective-Taking,” *Curr Opin Psychol*, vol. 54, pp. 101709, 2023.

[37] J. Li, H. Pan, Z. Lin, P. Fu, and W. Wang, “Sarcasm Detection with Commonsense Knowledge,” *IEEE/ACM Trans Audio Speech Lang Process*, vol. 29, pp. 3192–3201, 2021.

[38] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, “Sarcasm as contrast between a positive sentiment and negative situation,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 704–714, 2013.

[39] T. Ptáček, I. Habernal, and J. Hong, “Sarcasm detection on czech and english twitter,” in *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pp. 213–223, 2014.

[40] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, “Enhancing deep learning sentiment analysis with ensemble techniques in social applications,” *Expert Syst Appl*, vol. 77, pp. 236–246, 2017.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[42] Y. Ren, D. Ji, and H. Ren, “Context-augmented convolutional neural networks for twitter sarcasm detection,” *Neurocomputing*, vol. 308, pp. 1–7, 2018.

[43] Krishna, Thotakura Venkata Sai, et al. "A novel ensemble approach for Twitter sentiment classification with ML and LSTM algorithms for real-time tweets analysis." *Indonesian Journal of Electrical Engineering and Computer Science* vol. 34, no. 3, 2024.

[44] S. Hiai and K. Shimada, “Sarcasm detection using rnn with relation vector,” *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 15, no. 4, pp. 66–78, 2019.

[45] P. Mehndiratta and D. Soni, “Identification of sarcasm using word embeddings and hyperparameters tuning,” *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 22, no. 4, pp. 465–489, 2019.

[46] Y. Y. Tan, C.-O. Chow, J. Kanesan, J. H. Chuah, and Y. Lim, “Sentiment Analysis and Sarcasm Detection using Deep Multi-Task Learning,” *Wirel Pers Commun*, vol. 129, no. 3, pp. 2213–2237, 2023.

[47] J. Plepi and L. Flek, “Perceived and intended sarcasm detection with graph attention networks,” *arXiv preprint arXiv:2110.04001*, 2021.

[48] X. Mou, W. Xu, Y. Zhu, Q. Li, and Y. Xiao, “A social topic diffusion model based on rumor and anti-rumor and motivation-rumor,” *IEEE Trans Comput Soc Syst*, vol. 10, no. 5, pp. 2644-2659, Oct. 2023.

[49] G. Li, F. Lin, W. Chen, and B. Liu, “Affection Enhanced Relational Graph Attention Network for Sarcasm Detection,” *Applied Sciences*, vol. 12, no. 7, pp. 3639, 2022.

[50] Q. Lu, Y. Long, X. Sun, J. Feng, and H. Zhang, “Fact-sentiment incongruity combination network for multimodal sarcasm detection,” *Information Fusion*, vol. 104, pp. 102203, 2024.

[51] Y. Li *et al.*, “An attention-based, context-aware multimodal fusion method for sarcasm detection using inter-modality inconsistency,” *Knowl Based Syst*, vol. 287, pp. 111457, 2024.

[52] H. Lee, Y. Yu, and G. Kim, “Augmenting data for sarcasm detection with unlabeled conversation context,” *arXiv preprint arXiv:2006.06259*, 2020.

[53] P. Kumar and G. Sarin, “WELMSD–word embedding and language model based sarcasm detection,” *Online Information Review*, vol. 46, no. 7, pp. 1242–1256, 2022.

[54] B. Liang, C. Lou, X. Li, L. Gui, M. Yang, and R. Xu, “Multi-modal sarcasm detection with interactive in-modal and cross-modal graphs,” in *Proceedings of the 29th ACM international conference on multimedia*, pp. 4707–4715, 2021.

[55] N. Punetha and G. Jain, “Bayesian game model based unsupervised sentiment analysis of product reviews,” *Expert Syst Appl*, vol. 214, pp. 119128, 2023.

[56] N. Punetha and G. Jain, “Aspect and orientation-based sentiment analysis of customer feedback using mathematical optimization models,” *Knowl Inf Syst*, vol. 65, no. 6, pp. 2731–2760, 2023.

[57] N. Punetha and G. Jain, “Game theory and MCDM-based unsupervised sentiment analysis of restaurant reviews,” *Applied Intelligence*, pp. 1–22, 2023.

[58] A.-C. Băroiu and Ștefan Trăușan-Matu, “Comparison of Deep Learning Models for Automatic Detection of Sarcasm Context on the MUStARD Dataset,” *Electronics (Basel)*, vol. 12, no. 3, pp. 666, 2023.

[59] A. K. Mishra, M. S. Choudhry, and M. Kumar, “Underwater image enhancement using multiscale decomposition and gamma correction,” *Multimed Tools Appl*, vol. 82, no. 10, pp. 15715–15733, 2023.

[60] K. Mittal, A. Jain, K. S. Vaisla, O. Castillo, and J. Kacprzyk, “A comprehensive review on type 2 fuzzy logic applications: Past, present and future,” *Eng Appl Artif Intell*, vol. 95, pp. 103916, 2020.

[61] A. Chhabra and D. K. Vishwakarma, “Multimodal hate speech detection via multi-scale visual kernels and knowledge distillation architecture,” *Eng Appl Artif Intell*, vol. 126, pp. 106991, 2023.

[62] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” *arXiv preprint arXiv:1404.2188*, 2014.

[63] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, “Disan: Directional self-attention network for rnn/cnn-free language understanding,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[64] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489, 2016.

[65] A. Mohan, A. M. Nair, B. Jayakumar, and S. Muraleedharan, "Sarcasm Detection Using Bidirectional Encoder Representations from Transformers and Graph Convolutional Networks," *Procedia Comput Sci*, vol. 218, pp. 93–102, 2023.

[66] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Adv Neural Inf Process Syst*, vol. 32, 2019.

[67] Y. Cai, H. Cai, and X. Wan, "Multi-modal sarcasm detection in twitter with hierarchical fusion model," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2506–2515, 2019.

[68] S. He, F. Guo, and S. Qin, "Sarcasm Detection Using Graph Convolutional Networks with Bidirectional LSTM," in *Proceedings of the 3rd International Conference on Big Data Technologies*, pp. 97–101, 2020.

[69] P. Tiwari, L. Zhang, Z. Qu, and G. Muhammad, "Quantum Fuzzy Neural Network for multimodal sentiment and sarcasm detection," *Information Fusion*, vol. 103, pp. 102085, 2024.

[70] N. A. Helal, A. Hassan, N. L. Badr, and Y. M. Afify, "A contextual-based approach for sarcasm detection," *Sci Rep*, vol. 14, no. 1, pp. 15415, 2024.

[71] T. Chauhan and H. Palivela, "Optimization and improvement of fake news detection using deep learning approaches for societal benefit," *International Journal of Information Management Data Insights*, vol. 1, no. 2, pp. 100051, 2021.

[72] F. Ilievski, A. Oltramari, K. Ma, B. Zhang, D. L. McGuinness, and P. Szekely, "Dimensions of commonsense knowledge," *Knowl Based Syst*, vol. 229, pp. 107347, 2021.

[73] L. Ren, B. Xu, H. Lin, X. Liu, and L. Yang, "Sarcasm detection with sentiment semantics enhanced multi-level memory network," *Neurocomputing*, vol. 401, pp. 320–326, 2020.

[74] Y. Tay, L. A. Tuan, S. C. Hui, and J. Su, "Reasoning with sarcasm by reading in-between," *arXiv preprint arXiv:1805.02856*, 2018.

[75] R. Akula and I. Garibay, "Interpretable multi-head self-attention architecture for sarcasm detection in social media," *Entropy*, vol. 23, no. 4, pp. 394, 2021.

[76] Y. A. Kolchinski and C. Potts, "Representing social media users for sarcasm detection," *arXiv preprint arXiv:1808.08470*, 2018.

[77] Y. Zhang, Q. Li, D. Song, P. Zhang, and P. Wang, "Quantum-Inspired Interactive Networks for Conversational Sentiment Analysis.," pp. 5436–5442, 2019.

[78] A. Kumar, S. R. Sangwan, A. K. Singh, and G. Wadhwa, "Hybrid deep learning model for sarcasm detection in Indian indigenous language using word-emoji embeddings," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 5, pp. 1–20, 2023.

[79] Z. Keivanlou-Shahrestanaki, M. Kahani, and F. Zarrinkalam, "Interpreting sarcasm on social media using attention-based neural networks," *Knowl Based Syst*, vol. 258, p. 109977, 2022.

[80] Y. Du, T. Li, M. S. Pathan, H. K. Teklehaimanot, and Z. Yang, "An effective sarcasm detection approach based on sentimental context and individual expression habits," *Cognit Comput*, vol. 14, no. 1, pp. 78–90, 2022.

[81] L. Li, O. Levi, P. Hosseini, and D. A. Broniatowski, "A multi-modal method for satire detection using textual and visual cues," *arXiv preprint arXiv:2010.06671*, 2020.

[82] R. Schifanella, P. De Juan, J. Tetreault, and L. Cao, "Detecting sarcasm in multimodal social platforms," in *Proceedings of the 24th ACM international conference on Multimedia*, pp. 1136–1145, 2016.

[83] N. Xu, Z. Zeng, and W. Mao, "Reasoning with multimodal sarcastic tweets via modeling cross-modality contrast and semantic association," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 3777–3786, 2020.

[84] H. Pan, Z. Lin, P. Fu, Y. Qi, and W. Wang, "Modeling intra and inter-modality incongruity for multi-modal sarcasm detection," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1383–1392, 2020.

[85] V. Kolhatkar, H. Wu, L. Cavasso, E. Francis, K. Shukla, and M. Taboada, "The SFU opinion and comments corpus: A corpus for the analysis of online news comments," *Corpus Pragmatics*, vol. 4, pp. 155–190, 2020.

[86] Y. Qiao, L. Jing, X. Song, X. Chen, L. Zhu, and L. Nie, "Mutual-enhanced incongruity learning network for multi-modal sarcasm detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 9507–9515, 2023.

[87] Y. Zhang *et al.*, "Stance-level Sarcasm Detection with BERT and Stance-centered Graph Attention Networks," *ACM Trans Internet Technol*, vol. 23, no. 2, pp. 1–21, 2023.

[88] E. Savini and C. Caragea, "Intermediate-task transfer learning with BERT for sarcasm detection," *Mathematics*, vol. 10, no. 5, p. 844, 2022.

[89] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined CNN and LSTM model for Arabic sentiment analysis," in *Machine Learning and Knowledge Extraction: Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27–30, 2018, Proceedings 2*, Springer, pp. 179–191, 2028.

[90] E. Lunando and A. Purwarianti, "Indonesian social media sentiment analysis with sarcasm detection," in *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, IEEE, pp. 195–198, 2013.

[91] R. Justo Blanco, J. M. Alcaide Salinas, M. I. Torres Barañano, and M. Walker, "Detection of Sarcasm and Nastiness: New Resources for Spanish Language," vol. 10, pp. 1135–1151, 2018.

[92] D. G. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis," in *Lrec 2014 proceedings*, ELRA, 2014.

[93] D. S. Chauhan, G. V. Singh, A. Arora, A. Ekbal, and P. Bhattacharyya, "An emoji-aware multitask framework for multimodal sarcasm detection," *Knowl Based Syst*, vol. 257, pp. 109924, 2022.

[94] B. N. Hiremath and M. M. Patil, "Sarcasm detection using cognitive features of visual data by learning model," *Expert Syst Appl*, vol. 184, pp. 115476, 2021.

[95] S. Ilić, E. Marrese-Taylor, J. A. Balazs, and Y. Matsuo, “Deep contextualized word representations for detecting sarcasm and irony,” *arXiv preprint arXiv:1809.09795*, 2018.

[96] M. Bouazizi and T. O. Ohtsuki, “A pattern-based approach for sarcasm detection on twitter,” *IEEE Access*, vol. 4, pp. 5477–5488, 2016.

[97] T. Chen, R. Xu, Y. He, and X. Wang, “Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN,” *Expert Syst Appl*, vol. 72, pp. 221–230, 2017.

[98] S. K. Bharti and K. S. Babu, “Sarcasm as a contradiction between a tweet and its temporal facts: a pattern-based approach,” *International Journal on Natural Language Computing (IJNLC)*, vol. 7, no. 5, 2018.

[99] P. Katyayan and N. Joshi, “Performance evaluation of machine learning algorithms for detecting Hindi sarcasm,” *Bharatiya Vaigyanik evam Audyogik Anusandhan Patrika (BVAAP)*, vol. 29, no. 1, pp. 43–48, 2022.

[100] D. K. Jain, A. Kumar, and S. R. Sangwan, “TANA: The amalgam neural architecture for sarcasm detection in indian indigenous language combining LSTM and SVM with word-emoji embeddings,” *Pattern Recognit Lett*, vol. 160, pp. 11–18, 2022.

[101] S. K. Bharti, K. S. Babu, and R. Raman, “Context-based sarcasm detection in hindi tweets,” in *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, IEEE, pp. 1–6, 2017.

[102] K. Z. Bousmaha, K. Hamadouche, H. Djouabi, and L. Hadrich-Belguith, “Automatic Algerian Sarcasm Detection from Texts and Images,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 7, pp. 1-25, 2024.

[103] M. Marreddy, S. R. Oota, L. S. Vakada, V. C. Chinni, and R. Mamidi, “Am I a resource-poor language? Data sets, embeddings, models and analysis for four different NLP tasks in telugu language,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 1, pp. 1–34, 2022.

[104] U. Ahmed, J. C.-W. Lin, and G. Srivastava, “Emotional Intelligence Attention Unsupervised Learning Using Lexicon Analysis for Irony-based Advertising,” *ACM*

Transactions on Asian and Low-Resource Language Information Processing, vol. 23, no. 1, pp. 1–19, 2024.

[105] A. Joshi, V. Sharma, and P. Bhattacharyya, "Harnessing context incongruity for sarcasm detection," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 757–762, 2015.

[106] K. Sreelakshmi and P. C. Rafeeque, "An effective approach for detection of sarcasm in tweets," in *2018 International CET Conference on Control, Communication, and Computing (IC4)*, IEEE, pp. 377–382, 2018.

[107] F. Barbieri, H. Saggion, and F. Ronzano, "Modelling sarcasm in twitter, a novel approach," in *proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pp. 50–58, 2014.

[108] A. Ghosh and T. Veale, "Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 482–491, 2017.

[109] M. Bedi, S. Kumar, M. S. Akhtar, and T. Chakraborty, "Multi-modal sarcasm detection and humor classification in code-mixed conversations," *IEEE Trans Affect Comput*, vol. 14, no. 2, pp. 1363-1375, 1 April-June 2023.

[110] R. A. Potamias, G. Siolas, and A.-G. Stafylopatis, "A transformer-based approach to irony and sarcasm detection," *Neural Comput Appl*, vol. 32, no. 23, pp. 17309–17320, 2020.

[111] S. Gupta, R. Singh, and V. Singla, "Emoticon and text sarcasm detection in sentiment analysis," in *First International Conference on Sustainable Technologies for Computational Intelligence: Proceedings of ICTSCI 2019*, Springer, 2020, pp. 1–10, 2020.

[112] Y. Zhang *et al.*, "CFN: a complex-valued fuzzy network for sarcasm detection in conversations," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 12, pp. 3696–3710, 2021.

[113] H. Liu, R. Wei, G. Tu, J. Lin, C. Liu, and D. Jiang, "Sarcasm driven by sentiment: A sentiment-aware hierarchical fusion network for multimodal sarcasm detection," *Information Fusion*, vol. 108, pp. 102353, 2024.

[114] W. Chen, F. Lin, G. Li, and B. Liu, "A survey of automatic sarcasm detection: Fundamental theories, formulation, datasets, detection methods, and opportunities," *Neurocomputing*, vol. 578, pp. 127428, 2024.

[115] H. Wang, Y. Wang, X. Song, B. Zhou, X. Zhao, and F. Xie, "Quantifying controversy from stance, sentiment, offensiveness and sarcasm: a fine-grained controversy intensity measurement framework on a Chinese dataset," *World Wide Web*, vol. 26, no. 5, pp. 3607–3632, 2023.

[116] J. J. Todorov, R. T. Devine, and S. A. De Brito, "Association between childhood maltreatment and callous-unemotional traits in youth: A meta-analysis," *Neurosci Biobehav Rev*, vol. 146, pp. 105049, 2023.

[117] F. K. Lubis and S. Bahri, "Sarcasm in Indonesian Television Show Pesbukers," *Randwick International of Social Science Journal*, vol. 4, no. 1, pp. 91–99, 2023.

[118] X. Jia, D. Wang, D. Nie, X. Luo, and J. Z. Sun, "A new threshold changeable secret sharing scheme based on the Chinese Remainder Theorem," *Inf Sci (N Y)*, vol. 473, pp. 13–30, 2019.

[119] X. Li, X. G. Xia, W. Wang, and W. Wang, "A robust generalized Chinese remainder theorem for two integers," *IEEE Trans Inf Theory*, vol. 62, no. 12, 2016,

[120] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302-3312, 2017.

[121] H. Xiao, Y. Huang, Y. Ye, and G. Xiao, "Robustness in Chinese Remainder Theorem for Multiple Numbers and Remainder Coding," *IEEE Transactions on Signal Processing*, vol. 66, no. 16, pp. 4347-4361, 15 Aug.15, 2018.

[122] X. Li, W. Wang, W. Zhang, and Y. Cao, "Phase-Detection-Based Range Estimation with Robust Chinese Remainder Theorem," *IEEE Trans Veh Technol*, vol. 65, no. 12, 2016.

[123] I. A. Saroit and D. Tarek, "LBCC-Hung: A load balancing protocol for cloud computing based on Hungarian method," *Egyptian Informatics Journal*, vol. 24, no. 3, pp. 100387, 2023.

[124] H. Fei, Y. Ren, Y. Zhang, and D. Ji, "Nonautoregressive Encoder-Decoder Neural Framework for End-to-End Aspect-Based Sentiment Triplet Extraction," *IEEE Trans Neural Netw Learn Syst*, vol. 34, no. 9, 2023.

[125] C. Liu, L. Zheng, and C. Zhang, "Behavior perception-based disruption models for berth allocation and quay crane assignment problems," *Comput Ind Eng*, vol. 97, pp. 258-275, 2016.

[126] H. Zeng *et al.*, "EmotionCues: Emotion-Oriented Visual Summarization of Classroom Videos," *IEEE Trans Vis Comput Graph*, vol. 27, no. 7, pp. 3168-3181, 1 July 2021.

[127] D. Silveira, A. Carvalho, M. Cristo, and M. F. Moens, "Topic Modeling using Variational Auto-Encoders with Gumbel-Softmax and Logistic-Normal Mixture Distributions," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1-8, 2018.

[128] I. A. M. Huijben, W. Kool, M. B. Paulus, and R. J. G. Van Sloun, "A Review of the Gumbel-max Trick and its Extensions for Discrete Stochasticity in Machine Learning," *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 2, pp. 1353-1371, 1 Feb. 2023.

[129] L. Lin, H. Jiang, and Y. Rao, "Copula Guided Neural Topic Modelling for Short Texts," in *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1773 – 1776, 2020.

[130] K. Zheng, W. Liu, J. Liu, Z. J. Zha, and T. Mei, "Hierarchical Gumbel Attention Network for Text-based Person Search," in *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia*, pp. 3441 – 3449, 2020.

[131] P. Mardešić, G. Radunović, and M. Resman, "Fractal zeta functions of orbits of parabolic diffeomorphisms," *Anal Math Phys*, vol. 12, no. 5, pp. 114, 2022.

[132] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," *Augmented Human Research*, vol. 5, no. 1, pp. 12, 2020.

[133] S. Xu, C. Zhang, and D. Hong, "BERT-based NLP techniques for classification and severity modeling in basic warranty data study," *Insur Math Econ*, vol. 107, pp. 57-67, 2022.

[134] T. H. J. Hidayat, Y. Ruldeviyani, A. R. Aditama, G. R. Madya, A. W. Nugraha, and M. W. Adisaputra, "Sentiment analysis of twitter data related to Rinca Island development using Doc2Vec and SVM and logistic regression as classifier," in *Procedia Computer Science*, vol. 197, pp. 660-667, 2021.

[135] F. G. Akgül, B. Şenoılu, and T. Arslan, "An alternative distribution to Weibull for modeling the wind speed data: Inverse Weibull distribution," *Energy Convers Manag*, vol. 114, pp. 234-240, 2016.

[136] A. Abdullah Alahmadi *et al.*, "A New Version of Weighted Weibull Distribution: Modelling to COVID-19 Data," *Discrete Dyn Nat Soc*, vol. 2022, 2022.

[137] D. Rajpathak and S. De, "A data- and ontology-driven text mining-based construction of reliability model to analyze and predict component failures," *Knowl Inf Syst*, vol. 46, no. 1, pp. 87–113, 2016.

[138] S. D. Verifier and A. H. Drive, "Simulink ® Verification and Validation TM Reference," *ReVision*, 2015.

[139] Y. Cao, A. Seth, and C. D. Laird, "An augmented Lagrangian interior-point approach for large-scale NLP problems on graphics processing units," *Comput Chem Eng*, vol. 85, pp. 76-83, 2016.

[140] G. M. Linders and M. M. Louwerse, "Zipf's law revisited: Spoken dialog, linguistic units, parameters, and the principle of least effort," *Psychon Bull Rev*, vol. 30, no. 1, pp. 77–101, 2023.

[141] S. Lakshmi Sridevi and R. Devanathan, "Modification of Zipf-Mandelbrot Law for Text Analysis using Linear Regression," *Indian J Sci Technol*, vol. 10, no. 3, 2017.

[142] G. M. Linders and M. M. Louwerse, "Zipf's Law in Human-Machine Dialog," in *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents, IVA 2020*, no. 36, pp. 1-8, 2020.

[143] C. G. L. de Oliveira, J. G. V. Miranda, H. F. Japyassú, and C. N. El-Hani, "Using Zipf–Mandelbrot law and graph theory to evaluate animal welfare," *Physica A: Statistical Mechanics and its Applications*, vol. 492, pp. 285-295, 2018.

[144] J. Guo, P. Zhang, J. Wang, S. Li, and Z. Guan, "A novel framework for seismic fragility analysis with the combination of Box-Cox transformation and Bayesian inference," *Eng Struct*, vol. 277, pp. 115436, 2023.

[145] E. A. Siddiqui, V. Chaurasia, and M. Shandilya, "Detection and classification of lung cancer computed tomography images using a novel improved deep belief network with Gabor filters," *Chemometrics and Intelligent Laboratory Systems*, vol. 235, pp. 104763, 2023.

[146] V. L. Kuzmin, A. Y. Valkov, and Y. A. Zhavoronkov, "Coherent Backscattering Peak for Radiation with Low Spatial Coherence," *Journal of Experimental and Theoretical Physics*, vol. 137, no. 3, pp. 294–301, 2023.

[147] X. Tang, M. Zheng, J. Feng, J. Huang, and Y. Gong, "Shortcut Enhanced Syntactic and Semantic Dual-channel Network for Aspect-based Sentiment Analysis," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 11, pp. 1–20, 2023.

[148] A. Perti, A. Sinha, and A. Vidyarthi, "Cognitive hybrid deep learning-based multi-modal sentiment analysis for online product reviews," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 8, pp. 1–14, 2024.

[149] M. Khodak, N. Saunshi, and K. Vodrahalli, "A large self-annotated corpus for sarcasm," *arXiv preprint arXiv:1704.05579*, 2017.

[150] T. Dadu and K. Pant, "Sarcasm detection using context separators in online discourse," in *Proceedings of the Second Workshop on Figurative Language Processing*, pp. 51–55, 2020.

[151] S. Oprea and W. Magdy, "isarcasm: A dataset of intended sarcasm," *arXiv preprint arXiv:1911.03123*, 2019.

[152] R. Misra and P. Arora, "Sarcasm detection using news headlines dataset," *AI Open*, vol. 4, pp. 13–18, 2023.

[153] P. Tiwari, L. Zhang, Z. Qu, and G. Muhammad, “Quantum Fuzzy Neural Network for multimodal sentiment and sarcasm detection,” *Information Fusion*, vol. 103, pp. 102085, 2024.

[154] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*, PMLR, pp. 6105–6114, 2019.

[155] C. Lou, B. Liang, L. Gui, Y. He, Y. Dang, and R. Xu, “Affective dependency graph for sarcasm detection,” in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1844–1849, 2021.

[156] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.

[157] Y. Zhang *et al.*, “A Multitask learning model for multimodal sarcasm, sentiment and emotion recognition in conversations,” *Information Fusion*, vol. 93, pp. 282-301, 2023.

[158] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural networks*, vol. 18, no. 5–6, pp. 602–610, 2005.

[159] Y. Liu, Y. Zhang, Q. Li, B. Wang, and D. Song, “What does your smile mean? jointly detecting multi-modal sarcasm and sentiment using quantum probability,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 871–880, 2021.

[160] S. Pramanick, A. Roy, and V. M. Patel, “Multimodal learning using optimal transport for sarcasm and humor detection,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3930–3940, 2022.

[161] B. Javidy, A. Hatamlou, and S. Mirjalili, “Ions motion algorithm for solving optimization problems,” *Appl Soft Comput*, vol. 32, pp. 72–79, 2015.

[162] C. Yin and J. Xi, “Maximum entropy model for mobile text classification in cloud computing using improved information gain algorithm,” *Multimed Tools Appl*, vol. 76, pp. 16875–16891, 2017.

[163] J. Mazarura, A. De Waal, and P. de Villiers, “A Gamma-Poisson Mixture Topic Model for Short Text,” *Math Probl Eng*, vol. 2020, no. 1, pp. 4728095, 2020.

[164] Y. Neuman, Y. Cohen, N. Israeli, and B. Tamir, "A proposed methodology for studying the historical trajectory of words' meaning through tsallis entropy," *Physica A: Statistical Mechanics and its Applications*, vol. 492, pp. 804–813, 2018.

[165] H. Gregory, S. Li, P. Mohammadi, N. Tarn, R. Draelos, and C. Rudin, "A transformer approach to contextual sarcasm detection in twitter," in *Proceedings of the second workshop on figurative language processing*, pp. 270–275, 2020.

[166] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[167] M. S. M. Prasanna, S. G. Shaila, and A. Vadivel, "Polarity classification on twitter data for classifying sarcasm using clause pattern for sentiment analysis," *Multimed Tools Appl*, vol. 82, pp. 32789–32825, 2023.

[168] I. Chaturvedi, C. L. Su, and R. E. Welsch, "Fuzzy aggregated topology evolution for cognitive multi-tasks," *Cognit Comput*, vol. 13, pp. 96–107, 2021.

[169] A.-M. Iddrisu, S. Mensah, F. Boafo, G. R. Yeluripati, and P. Kudjo, "A sentiment analysis framework to classify instances of sarcastic sentiments within the aviation sector," *International Journal of Information Management Data Insights*, vol. 3, no. 2, pp. 100180, 2023.

[170] Y. Zhang *et al.*, "Learning multi-task commonness and uniqueness for multi-modal sarcasm detection and sentiment analysis in conversation," *IEEE Transactions on Artificial Intelligence*, 2023.

[171] B. Singh and D. K. Sharma, "A survey of sarcasm detection techniques in natural language processing," in *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, IEEE, 2023, pp. 1–6, 2023.

[172] R. T. Salekin, H. Andershed, B. D. Batky, and A. P. Bontemps, "Are callous unemotional (CU) traits enough?," *J Psychopathol Behav Assess*, vol. 40, pp. 1–5, 2018.

[173] P. J. Frick and J. V Ray, "Evaluating callous-unemotional traits as a personality construct," *J Pers*, vol. 83, no. 6, pp. 710–722, 2015.

[174] D. K. Sharma, B. Singh, S. Agarwal, H. Kim, and R. Sharma, "Sarcasm detection over social media platforms using hybrid auto-encoder-based model," *Electronics (Basel)*, vol. 11, no. 18, p. 2844, 2022.

[175] J. Tian, S. Chen, X. Zhang, X. Wang, and Z. Feng, "Reducing sentiment bias in pre-trained sentiment classification via adaptive gumbel attack," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 13646–13654, 2023.

[176] B. Mullick, R. Magar, A. Jhunjhunwala, and A. B. Farimani, "Understanding mutation hotspots for the sars-cov-2 spike protein using shannon entropy and k-means clustering," *Comput Biol Med*, vol. 138, pp. 104915, 2021.

[177] I. Naruei, F. Keynia, and A. Sabbagh Molahosseini, "Hunter–prey optimization: Algorithm and applications," *Soft comput*, vol. 26, no. 3, pp. 1279–1314, 2022.

[178] R. D. Parshad, S. Bhowmick, V. Chand, N. Kumari, and N. Sinha, "What is India speaking? Exploring the 'Hinglish' invasion," *Physica A: Statistical Mechanics and its Applications*, vol. 449, pp. 375–389, 2016.

[179] D. K. Sharma, B. Singh, S. Agarwal, N. Pachauri, A. A. Alhussan, and H. A. Abdallah, "Sarcasm Detection over Social Media Platforms Using Hybrid Ensemble Model with Fuzzy Logic," *Electronics (Basel)*, vol. 12, no. 4, pp. 937, 2023.

[180] A. J. Birmicombe and Y. Li, "An Auto-ML Framework for social media domain feature construction to detect Sarcasm," 2023.

[181] Y. Zhang, Y. Yu, M. Wang, M. Huang, and M. S. Hossain, "Self-Adaptive Representation Learning Model for Multi-Modal Sentiment and Sarcasm Joint Analysis," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 5, pp. 1–17, 2024.

[182] N. Punetha and G. Jain, "Integrated shannon entropy and COPRAS optimal model-based recommendation framework," *Evol Intell*, vol. 17, no. 1, pp. 385–397, 2024.

[183] A. Kumar and G. Garg, "The multifaceted concept of context in sentiment analysis," in *Cognitive Informatics and Soft Computing: Proceeding of CISC 2019*, Springer, pp. 413–421, 2020.

[184] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcasm in Twitter and Amazon," in *Proceedings of the fourteenth conference on computational natural language learning*, pp. 107–116, 2010.

[185] S. Swami, A. Khandelwal, V. Singh, S. S. Akhtar, and M. Shrivastava, “A corpus of english-hindi code-mixed tweets for sarcasm detection,” *arXiv preprint arXiv:1805.11869*, 2018.

[186] D. Vijay, A. Bohra, V. Singh, S. S. Akhtar, and M. Shrivastava, “A Dataset for Detecting Irony in Hindi-English Code-Mixed Social Media Text.,” *EMSASW@ ESWC*, vol. 2111, pp. 38–46, 2018.

[187] D. S. Kulkarni and S. S. Rodd, “Sentiment analysis in Hindi—A survey on the state-of-the-art techniques,” *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 1, pp. 1–46, 2021.

[188] R. K. Gupta and Y. Yang, “Crystalnest at semeval-2017 task 4: Using sarcasm detection for enhancing sentiment classification and quantification,” in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 626–633, 2017.

[189] N. Pawar and S. Bhingarkar, “Machine learning based sarcasm detection on Twitter data,” in *2020 5th international conference on communication and electronics systems (ICCES)*, IEEE, pp. 957–961, 2020.