A publication of ISCA*:
International Society for Computers
and Their Applications

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

***"International Journal of Computers and Their Applications is Peer Reviewed".**

Volume 32, No. 3, September 2025 **ISSN 1076-5204**

International Journal of Computers and Their Applications
A publication of the International Society for Computers and Their Applications

EDITOR-IN-CHIEF

Ajay Bandi
Professor
School of Computer Science and Information Systems
Northwest Missouri State University
800 University Drive, Maryville, MO, USA 64468
Email: ajay@nwmissouri.edu

EDITORIAL BOARD

Editorial

It is my distinct honor, pleasure, and privilege to serve as the Editor-in-Chief of the International Journal of Computers and Their Applications (IJCA) since 2022. I have a special passion for the International Society for Computers and their Applications. I have been a member of our society since 2014 and have served in various capacities. These have ranged from being on program committees of our conferences to being Program Chair of CATA since 2021 and currently serving as one of the Ex-Officio Board Members. I am very grateful to the ISCA Board of Directors for giving me this opportunity to serve society and the journal in this role.

I would also like to thank all the editorial board, editorial staff, and authors for their valuable contributions to the journal. Without everyone's help, the success of the journal would be impossible. I look forward to working with everyone in the coming years to maintain and further improve the journal's quality. I want to invite you to submit your quality work to the journal for consideration for publication. I also welcome proposals for special issues of the journal. If you have any suggestions to improve the journal, please feel free to contact me.

Dr. Ajay Bandi
School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, MO 64468
Email: AJAY@nwmissouri.edu

In 2025, we are having four issues planned (March, June, September, and December). The next latest issue is taking shape with a collection of submitted papers.

I would also like to announce that I will begin searching for a few reviewers to add to our team. We want to strengthen our board in a few areas. If you would like to be considered, don't hesitate to get in touch with me via email with a cover letter and a copy of your CV.

Ajay Bandi, Editor-in-Chief
Email: AJAY@nwmissouri.edu

This issue of the International Journal of Computers and their Applications (IJCA) has gone through the normal review process. The papers in this issue cover a broad range of research interests in the community of computers and their applications.

IJCA Contributed Papers: This issue comprises papers that were contributed to the International Journal of Computers and their Applications (IJCA). The topics and main contributions of the papers are briefly summarized below:

Indranil Roy and Reshmi Mitra, Southeast Missouri State University, USA; Bidyut Gupta, Southern Illinois University, USA; and Narayan C. Debnath, Eastern International University, Vietnam, present their study titled "**Design of a Modified Low Diameter Interest-Based Peer-to-Peer Network Architecture.**" This research proposes an enhanced version of the existing Residue-class (RC) based peer-to-peer (P2P) network architecture to significantly reduce data look-up latency. While the original RC architecture forms interest-based peer groups of diameter one with group heads arranged in a ring of diameter n/2, the proposed design connects all group heads directly, resulting in a network of diameter one among heads and an overall P2P diameter of just three compared to (n/2 + 2) in prior work. Using a mathematical model, the authors demonstrate that this is the shortest achievable diameter in such networks, leading to highly efficient data look-up protocols with substantially lower latency, even under network churn conditions.

Manit Malhotra and Indu Chhabra, Panjab University, Chandigarh, India, present their study titled **"Detection of Academic Dishonesty in Student Records Using Anomaly Detection with Deep Learning and Machine Learning Techniques."** This research addresses the challenge of maintaining academic integrity during the shift to online education, particularly in the wake of the COVID-19 pandemic. Using mark sheet data from 3,000 students spanning six semesters (three online and three offline) at Panjab University constituent colleges, the authors explored machine learning (ML) and deep learning (DL) approaches to detect cheating behaviors. The dataset reflected a subtle-to-blatant cheating imbalance of approximately 3:2, mitigated through hybrid resampling techniques. Models were evaluated using an 80/20 train-test split with bootstrapped confidence intervals. Among four models tested, the BiLSTM achieved the best performance, recording 97.6% accuracy and an AUC of 1.00, surpassing LSTM by +1.6% accuracy and Random Forest by +22%. The study demonstrates the effectiveness of DL-based methods for scalable, automated cheating detection, while noting that results are limited to numerical mark sheet features from a single institution and call for broader validation.

Maza Abdelwahab and Lyazid Sabri, Mohamed El Bachir EL Ibrahimi University of Bordj Bou Arréridj, Algeria, present their study titled **"Spatio-Temporal Ontological Query Processing in IoE Environments."** This research focuses on leveraging ontologies to manage, analyze, and understand the semantic context of data generated by interconnected devices, sensors, and people within the Internet of Everything (IoE) ecosystem. The proposed approach uses ontology-based querying of chronological events, incorporating temporal ontology and semantic reasoning to improve activity recognition and deliver early warnings for potential risks such as health deterioration or unsafe behavior. By combining spatial and temporal data with contextual awareness, the system dynamically assesses environments, performs adaptive processing, predicts future issues, and personalizes analyses. Unlike traditional temporal Description Logic frameworks for dynamic context recognition and spatiotemporal concept representation, this spatio-temporal querying method enhances responsiveness, improves efficiency, and supports more relevant human–machine interaction.

Nidhi Mishra, Aakanshi Gupta, Shuchi Mala, Srishti Das, Naman Tyagi, and Sanjam Bhardwaj, Amity School of Engineering and Technology, AUUP, Noida, India; Narayan C. Debnath, School of Computing and Information Technology, Binh Duong Province, Vietnam; and Amit Mishra, Jaypee Institute of Information Technology, Noida, present their study titled **"Multi Stream Attention Networks with Adaptive Syntactic-Emotional Fusion for Sentiment Analysis."** This research introduces a dual-stream attention architecture that separately models syntactic structure and emotional intensity in text, integrating them through an adaptive fusion mechanism. The model was evaluated on the IMDb Movie Review dataset and compared against lexicon-based approaches such as TextBlob, SentiWordNet, and VADER, as well as machine learning methods using TF-IDF and Bag-of-Words features and sequential deep learning models including LSTM and BiLSTM with GloVe embeddings. Ensemble techniques such as stacking and voting were also examined to enhance classification performance. The proposed architecture demonstrated superior results across key evaluation metrics, including accuracy, precision, recall, F1-score, and AUC-ROC, highlighting the significance of jointly modeling syntactic and emotional cues for robust sentiment analysis.

Houcem Rezaiguia and Abdelhamid Djeffal, University of Biskra, Algeria, present their study titled **"Deep Learning Approach for Anxiety and Stress Detection through Facial Emotion Analysis."** This research introduces ANet, a novel framework leveraging the AlexNet Convolutional Neural Network (CNN) model to detect anxiety and stress by analyzing facial emotions. The system begins by obtaining facial images from a public database, applying the Viola–Jones algorithm for face detection, and extracting the Region of Interest (ROI). During training, feature maps are generated using AlexNet to learn emotional representations, while input videos undergo frame extraction, face detection, and ROI processing for real-time classification. The framework identifies six basic emotions anger, sadness, happiness, disgust, fear, and neutral and determines anxiety levels categorized as high, moderate, mild, or none by analyzing emotion combinations. Experimental results demonstrate that ANet provides a reliable and efficient method for early anxiety detection, highlighting its potential for real-time emotional assessment and mental health monitoring.

As guest editors, we would like to express our deepest appreciation to the authors and the reviewers. We hope you will enjoy this issue of the IJCA. More information about ISCA society can be found at http://www.isca-hq.org.

Guest Editors:

Ajay Bandi, Northwest Missouri State University, USA.

SEPTEMBER 2025

Design of a Modified Low Diameter Interest-Based Peer-to-Peer Network Architecture

Indranil Roy*, Reshmi Mitra*
Southeast Missouri State University, Cape Girardeau, MO, USA
iroy@semo.edu

Bidyut Gupta†
Southern Illinois University, Carbondale, IL, USA
bidyut@cs.siu.edu

Narayan C. Debnath‡
Eastern International University, Vietnam
NdebnathC@gmail.com

Abstract

In this work, we have proposed a modified version of the existing Residue-class (RC) based peer-to peer (P2P) network architecture. The existing RC based architecture [10] has been considered because of its two main advantages viz. (1) all peers with the same interest (or possessing the same resource type) structurally form a group of diameter one, and (2) the group heads are connected in the form of a ring and the ring always remains connected even in the presence of any churn. The diameter of the ring with n group heads is n/2 and as n increases, the latency of the existing data look-up protocols increases as well. In this paper our objective is to reduce this diameter substantially in order to reduce data look-up latency drastically. To achieve this, we have used a mathematical model using which we show that all group heads will be connected directly to each other; that is, instead of forming a ring structure of diameter n/2, it will be a network of diameter one only. Therefore, the diameter of the peer-to-peer network becomes only 3 instead of (n/2 + 2) as in [10]. This is the shortest diameter possible in such networks. It will definitely make any data look-up protocol highly efficient from the viewpoint of low latency.

Key Words: Residue Class, Interest-based P2P Network, Diameter, Capacity Constrained, Look-up Latency, Broadcast.

1 Introduction

Peer-to-peer (P2P) overlay networks have emerged as one of the most influential paradigms in the design of distributed systems, offering the ability to share computational and data resources in a scalable, self-organizing, and decentralized manner. Unlike traditional client-server models, P2P systems remove single points of failure, enhance robustness, and enable massive scalability by distributing responsibilities across all participating nodes (peers). This autonomy and flexibility make P2P overlays highly attractive for applications ranging from file sharing and content distribution to real-time communication, distributed computation, and decentralized data storage. In recent years, the paradigm has extended into emerging domains such as blockchain, federated learning, IoT integration, and decentralized social networking, where privacy preservation, scalability, and fault tolerance are paramount.

1.1 Classification of P2P Overlays

P2P overlay networks can be broadly categorized into two primary classes: *unstructured* and *structured* overlays. This classification is based on the way nodes are interconnected and how resources are indexed and retrieved. A third, increasingly relevant category includes *hybrid* and *non-DHT structured* overlays, which combine properties of the first two classes to optimize specific performance metrics.

1.1.1 Unstructured P2P Overlays

In unstructured systems [2], peers are connected in an arbitrary topology without adherence to a strict organizational structure. Resource discovery typically relies on *flooding*, *random walks*, or other probabilistic search mechanisms, which can locate resources even under high churn conditions (frequent peer join/leave events). This high resilience to churn is a defining strength of unstructured networks, enabling them to operate in highly dynamic environments.

However, these benefits come at a cost:

- **Efficiency:** Queries may require broadcasting across large portions of the network, leading to excessive message overhead and bandwidth consumption.
- **Scalability:** As network size grows, search latency and message complexity increase superlinearly, making real-time lookups challenging at scale.
- **Reliability:** Lookups are not guaranteed unless extensive replication or exhaustive search mechanisms are employed, which further increases resource usage.

Prominent examples include early systems such as Gnutella and Freenet, which demonstrated the feasibility of large-scale decentralized communication but also highlighted the limitations of unstructured search methods. Modern adaptations sometimes integrate caching and proximity-aware forwarding to improve performance, but scalability challenges remain.

1.1.2 Structured P2P Overlays

Structured overlays enforce deterministic organization of peers and resources, ensuring predictable and bounded search times. This is typically achieved by mapping resources to nodes using a globally consistent data placement strategy. *Distributed*

Hash Tables (DHTs) [18, 25, 29] form the backbone of many structured P2P systems, such as Chord, Pastry, and CAN [15, 27]. In these systems:

- Lookup operations have a worst-case complexity of $O(\log N)$, where N is the number of peers.
- The overlay is maintained through periodic updates to routing tables, ensuring consistent connectivity despite churn.
- Resource placement is uniform due to hashing, aiding load balancing across peers.

While DHT-based overlays offer strong guarantees for scalability and deterministic lookups, they incur high maintenance complexity under churn. Each join or leave event triggers reorganization of the overlay and updates to routing state, which can result in high control overhead. This is particularly problematic in mobile ad hoc networks, IoT deployments, and other environments with volatile peer availability. Furthermore, DHTs tend to be content-agnostic; while this aids uniformity, it ignores semantic locality, which can be critical for certain applications.

1.2 Hybrid and Non-DHT Structured Approaches

To address the limitations of pure DHT and unstructured systems, researchers have proposed *hybrid architectures* [5, 14, 16, 24, 28]. These architectures combine structured and unstructured elements—for example, by organizing peers into clusters connected by a structured backbone (DHT) while maintaining unstructured links within clusters. This approach seeks to inherit the benefits of both paradigms: deterministic inter-cluster lookups and churn-resilient intra-cluster communication. However, hybrid designs may also inherit drawbacks from both sides, such as DHT churn sensitivity and unstructured search inefficiency within clusters.

Another line of research has focused on *non-DHT structured approaches* [3]. These designs avoid hash-based resource placement and instead rely on mathematically driven or topology-aware organization, such as hierarchical clustering, tree-based overlays, grid-based coordinate systems, or residue-class–based addressing. Non-DHT structured systems aim to:

- Preserve deterministic search performance with predictable latency bounds.
- Reduce maintenance complexity compared to DHT-based overlays by limiting reorganization scope.
- Enable more natural grouping of peers based on real-world constraints (e.g., geographic proximity, latency sensitivity, or resource similarity).

Examples include skip-graph overlays, hypercube-based routing, and tree-based hierarchical models that exploit structured locality for performance gains.

1.3 Interest-Based P2P Overlays

A particularly important subclass of non-DHT approaches is *interest-based P2P overlays*. These systems group peers into clusters or communities based on shared interests, resource types, or application-specific attributes, thereby:

- Reducing the search space and improving lookup relevance by focusing queries on semantically relevant subsets of peers.
- Supporting deterministic intra-cluster searches (often $O(1)$ latency in fully connected cluster topologies).
- Simplifying maintenance by limiting topology updates to smaller, interest-specific clusters.

Examples in the literature [1, 4, 6–13, 17, 19–23, 26] vary widely in their approaches:

- **Super-peer or popular-peer models** [4, 13, 26] designate high-capacity nodes as entry points or aggregators for peers with similar interests, balancing efficiency with robustness.
- **Gossip-based clustering** [11] uses probabilistic peer exchanges to form and maintain interest groups without centralized control.
- **Hybrid interest-aware DHTs** [8, 23] embed interest semantics into DHT routing to accelerate relevant lookups.
- **Non-DHT mathematical approaches**, such as the **pyramid tree architecture** [20], which leverages residue-class addressing to ensure complete intra-cluster connectivity and predictable routing.

While these designs have achieved notable improvements in search latency and relevance, several challenges remain open:

1. **Efficient inter-cluster routing** between geographically or topologically dispersed interest communities without excessive control traffic.
2. **Scalability** under heterogeneous workloads where cluster sizes and interest popularity vary widely.
3. **Robustness against churn** while maintaining deterministic lookup guarantees and minimizing maintenance costs.
4. **Applicability to emerging domains** such as industrial IoT, social network overlays, blockchain-based marketplaces, and federated learning systems, where low latency and context awareness are crucial.

1.4 Motivation for the Proposed Work

The present research builds upon these foundations, aiming to address the limitations identified in existing interest-based and non-DHT structured approaches. In particular, we focus on designing a scalable, mathematically grounded architecture that:

- Achieves constant-time intra-cluster lookups and logarithmic-time inter-cluster searches, ensuring predictable performance regardless of network scale.

- Handles churn with minimal restructuring overhead by localizing maintenance to affected clusters.
- Integrates seamlessly with application domains requiring high availability, semantic relevance, and low latency, such as real-time industrial monitoring, healthcare data sharing, and distributed machine learning.

By systematically analyzing existing designs and identifying their architectural strengths and weaknesses, our work proposes an improved model that advances the state of the art in interest-based P2P systems. Our proposed approach incorporates residue-class–based hierarchical organization for efficient routing, dynamic cluster adaptation for churn resilience, and semantic indexing for relevance-aware lookups, making it adaptable to a wide range of distributed computing environments.

2 Literature Survey of some Existing Interest Based P2P Architectures

In this section, we examine a set of significant interest-based peer-to-peer (P2P) systems [1,4,8,9,11,13,20,23,26], outlining their architectural principles, cluster formation strategies, and handling of peer heterogeneity. We emphasize both the similarities and the limitations of these systems, highlighting the need for more efficient, churn-resilient architectures.

2.1 Super Peer and Popular Peer Architectures

The works in [4, 13, 26] incorporate peer heterogeneity through the concepts of *super peers* or *popular peers*, which act as high-capacity nodes managing clusters.

- **[13]** adopts the super peer model, forming clusters via gossiping among peers with common interests. This creates a hierarchy where super peers maintain intra-cluster coordination.
- **[4]** proposes the *popular peer* concept, functionally similar to the super peer model. The underlying network is unstructured, meaning there is no fixed routing topology and searches may require flooding.
- **[26]** presents a hybrid system integrating a Chord-based structured overlay with unstructured peer groups, also leveraging super peers for improved routing efficiency.

2.2 Gossiping and Best-Peer Selection Strategies

In [11], gossiping is employed for interest-based cluster formation. Upon joining, a peer searches a known peer list to identify the node with the most connections and links to it, creating high-degree hubs. While this can improve connectivity, it introduces significant latency during the joining process and increases dependence on a few central nodes.

Conversely, [20] criticizes gossiping as inefficient for interest-based clustering. The authors also reject super peer/popular peer models, noting that newly joining peers could outperform existing high-capacity nodes, requiring repeated leader re-selection. This re-selection process wastes time, particularly during bursts of new peer arrivals.

2.3 Interest Community Formation Approaches

The work in [9] develops a resource location strategy that leverages collaborative information exchange to form *interest communities*. Similar peers are grouped, and relevant data is disseminated to improve search accuracy within communities.

In [8], a Pastry-based P2P e-commerce model organizes peers with shared interests into clusters, assuming direct connections among them (overlay diameter = 1). However, no mathematical basis is provided for this assumption, making scalability and resilience under churn questionable.

2.4 Proximity-Aware and Hybrid DHT Approaches

The system in [23] is a DHT-based structured P2P network that considers both *physical proximity* and *common interest*. Initially, clusters are formed among physically close peers, and then sub-clusters are created for interest alignment. While this improves local search efficiency, it fails to address inter-sub-cluster lookups across geographically distributed clusters sharing the same interest, leaving a critical gap in scalability.

2.5 Social Network-Influenced Architectures

In [1], the Chord architecture is adapted to support social network characteristics. Peers establish *interest links* dynamically based on prior communication patterns. An efficient routing algorithm exploits these links to improve lookup performance without abandoning the structured overlay.

2.6 Residue Class-Based Non-DHT Approach

The Pyramid Tree architecture proposed in [20] takes a fundamentally different path, avoiding DHT structures altogether. It applies modular arithmetic (residue class) to form clusters where:

- Each interest-based cluster is a complete graph, ensuring an intra-cluster search latency of $O(1)$.
- The overall network diameter is $2d+2$, where d is the number of pyramid tree levels. Inter-cluster search latency is $O(d)$, with d being small compared to total peers since only cluster-heads populate the tree.
- Cluster size is unrestricted, and churn handling is simplified since the complete graph topology remains unchanged by peer joins/leaves.
- New cluster-heads for novel resource types are always placed at the leaf level, ensuring predictable growth.

This approach addresses many shortcomings of prior work, including inefficiencies in gossiping, unstructured search, churn sensitivity in DHTs, and incomplete inter-cluster communication handling.

We have presented the above works in a comparative summary table as shown in Table 1.

Problem Statement

The existing two-level residue class-based (RC-based) peer-to-peer (P2P) architecture [10] has been considered in this work due to its several distinctive advantages. It is an *interest-based* system, meaning that peers are logically grouped according to shared interests or resource types, which improves the relevance and efficiency of resource discovery. Two of the most prominent features relevant to our study are:

1. **Clustered organization with minimal intra-group diameter:** All peers with the same interest (or possessing the same resource type) structurally form a group of *diameter one*. This ensures constant-time intra-group lookups and eliminates unnecessary routing within the group.
2. **Churn-resilient backbone:** The group heads are connected in the form of a logical ring. This ring structure guarantees connectivity among groups even under churn, as each group head maintains at least two persistent connections to its immediate neighbors in the ring.

Under this design, the overall diameter of the P2P network is computed as the *diameter of the ring backbone* plus *twice the group diameter*. Since each group has a diameter of one, the total network diameter is given by:

$$\text{Diameter}_{\text{RC}} = \frac{n}{2} + 2$$

where n is the total number of groups (or group heads). This low diameter has been shown to be beneficial for achieving low-latency data lookups compared to many unstructured or DHT-based overlays.

However, upon closer examination, we observe that the network's lookup latency is heavily dependent on the ring diameter, which is $\frac{n}{2}$ in the worst case. As n grows, this term becomes the dominant factor in lookup delay. While the original RC-based design offers good scalability and churn resilience, its performance under high values of n can be further improved by optimizing the inter-group backbone topology.

Proposed Modification. In the research presented in this paper, we propose a modified architecture in which *all group heads are directly connected to each other*, thereby transforming the backbone from a ring topology into a fully connected graph. This change reduces the backbone diameter from $\frac{n}{2}$ to **1**. Consequently, the overall network diameter becomes:

$$\text{Diameter}_{\text{Proposed}} = 3$$

This is derived as follows: (1) one hop from the source peer to its group head, (2) one hop directly between the source and destination group heads, and (3) one hop from the destination group head to the target peer. This represents the **shortest possible diameter** achievable in such multi-group P2P systems while maintaining the interest-based grouping.

Advantages and Implications. The proposed architecture offers several key benefits:

- **Reduced worst-case latency:** Lookup operations complete in at most three hops, independent of n.
- **Improved throughput:** Shorter paths reduce intermediate forwarding load and congestion.
- **Enhanced fault tolerance:** Fully connected group heads provide multiple alternate shortest paths, eliminating single-path dependencies.

The remainder of this paper is organized as follows: Section 3 presents relevant preliminaries. Section 4 details the proposed modified architecture. In Section 5, we describe the broadcast protocols for the proposed network, both with and without capacity constraints. Section 6 concludes the paper with final remarks and outlines potential future research directions, particularly in the area of secure and efficient communication over the proposed topology.

3 RC Based Topology

We first state briefly the architecture of the RC-based network, followed by the STAR topology.

3.1 RC Based architecture [10]

Definition 1. *We define a resource as a tuple $< R_i, V >$, where R_i denotes the type of a resource and V is the value of the resource.*

A resource can have many values. For example, let R_i denote the resource type 'movies' and V' denote a particular actor. Thus $< R_i, V >$ represents movies (some or all) acted by a particular actor V'.

Definition 2. *Let S be the set of all peers in a peer-to-peer system. Then $S = \{P^{Ri} \mid 0 \leq i \leq n-1\}$, where P^{Ri} denotes the subset consisting of all peers with the same resource type R_i, and the number of distinct resource types present in the system is n. Also, for each subset P^{Ri}, we assume that P_i is the first peer among the peers in P^{Ri} to join the system. We call P_i the group-head of group G_i formed by the peers in the subset P^{Ri}. For each subset P^{Ri}, we assume that P_i is the first peer among the peers in P^{Ri} to join the system. We call P_i the group-head of group G_i formed by the peers in the subset P^{Ri}.*

3.1.1 Two level P2P architecture

It is a two-level overlay architecture and at each level structured networks of peers exist. It is explained below.

1. At level-1, it is a ring network consisting of the peers P_i $(0 \leq i \leq n-1)$. The number of peers (i.e., group heads) on the ring is n, which is also the number of distinct resource types. This ring network is used for efficient data lookup and so it is named the transit ring network [10].

Table 1: Comparison of Selected Interest-Based P2P Architectures

Ref.	Architecture Type	Clustering Method	Churn Handling	Overlay Diameter	Main Limitations
[13]	Unstructured w/ Super Peers	Gossip-based	Moderate (depends on super peers)	Variable (depends on flooding)	Inefficient gossiping; reliance on super peers
[4]	Unstructured w/ Popular Peers	Interest-based peer selection	Moderate	Variable	Similar to super peer issues; unstructured search overhead
[26]	Hybrid (Chord + Unstructured)	Super peer-based clusters	Moderate (DHT churn impact)	$O(\log N)$ for DHT part	Complexity of hybrid maintenance
[11]	Unstructured	Gossip + best-peer selection	Low	Variable	Joining delay; centralization risk
[9]	Unstructured	Collaborative interest community	Low	Variable	Scalability issues; no proximity consideration
[23]	Structured DHT	Proximity + interest clustering	Moderate (DHT churn)	$O(\log N)$ intra-cluster; unknown inter-sub-cluster	Incomplete inter-cluster routing
[8]	Pastry-based	Interest community (1-hop assumption)	Low	1 hop (assumed)	Unrealistic assumption; no churn resilience
[1]	Chord-based DHT	Dynamic interest links	Moderate	$O(\log N)$	Churn maintenance overhead
[20]	Non-DHT Pyramid Tree	Residue class clustering	High (simple)	1 (intra), $2d+2$ (inter)	None significant; assumes accurate interest identification

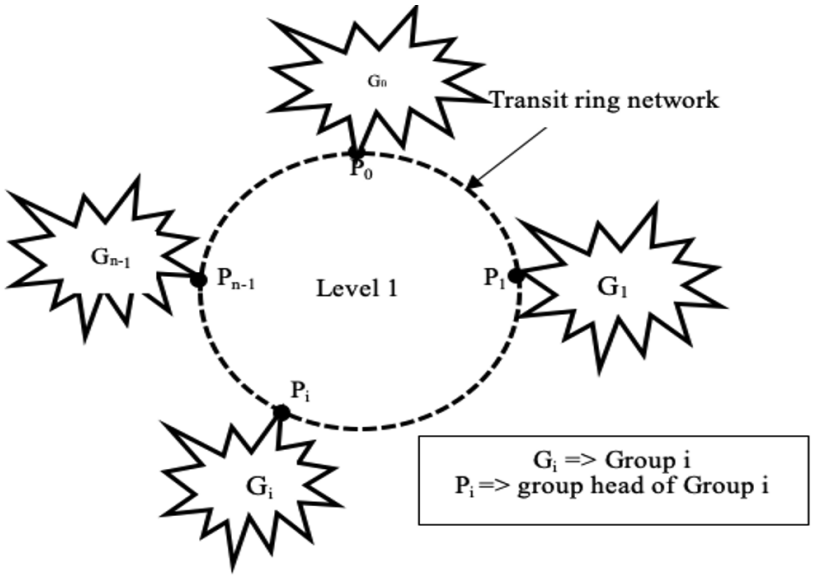


Figure 1: An RC-based 2-level P2P network

2. At level-2, there are n completely connected networks (groups) of peers. Each such group, say G_i, is formed by the peers of the subset P^{Ri} $(0 \leq i \leq n-1)$, such that all peers ($\in P^{Ri}$) are directly connected (logically) to each other, resulting in a network diameter of 1. Each G_i is connected to the transit ring network via its group-head P_i.
3. Any communication between a peer $p'_i \in G_i$ and $p'_j \in G_j$ takes place only via the respective group-heads P_i and P_j.

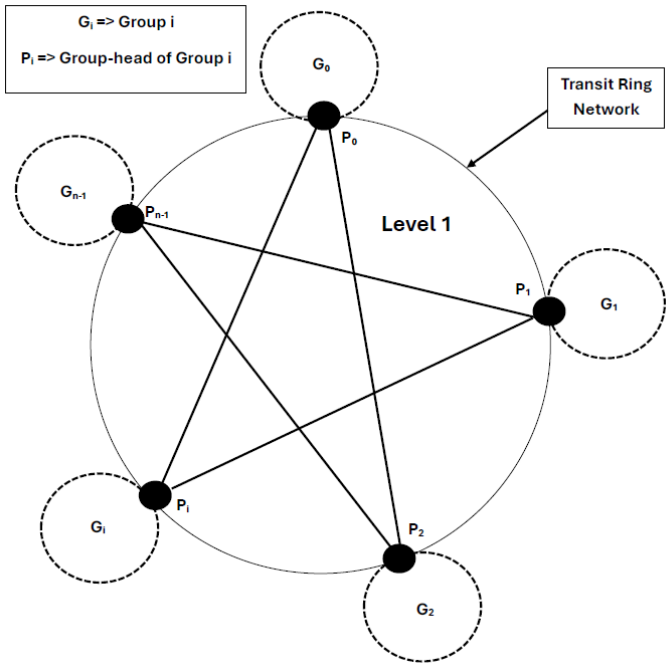


Figure 2: The modified RC-based 2-level P2P architecture with a complete network at Level-1

3.2 Assignments of overlay addresses

Consider the set S_n of nonnegative integers less than n, given as $S_n = \{0,1,2,\ldots,(n-1)\}$. This is referred to as the set of residues, or residue classes (mod n). That is, each integer in S_n represents a residue class (RC). These residue classes can be labelled as $[0],[1],[2],\ldots,[n-1]$, where $[r] = \{a : a \text{ is an integer}, a \equiv r \pmod{n}\}$.

For example, for $n = 3$, the classes are:

$$[0] = \{\ldots,-6,-3,0,3,6,\ldots\}$$
$$[1] = \{\ldots,-5,-2,1,4,7,\ldots\}$$
$$[2] = \{\ldots,-4,-1,2,5,8,\ldots\}$$

A relevant property of residue class is stated below.

Lemma 1. *Any two numbers of any class r of S_n are mutually congruent.*

The architecture [10] is shown in Figure 1.

Assume that in an interest-based P2P system there are n distinct resource types. Consider the set of all peers in the system given as $S = \{P^{Ri}\}, 0 \le i \le n-1$. Also, as mentioned earlier, for each subset P^{Ri} (i.e., group G_i) peer P_i is the first peer with resource type R_i to join the system. This first peer P_i is the group-head of group G_i.

The assignment of overlay (logical) addresses to the peers at the two levels and the resources happens as follows:

1. At level-1, each group-head P_r of group G_r is assigned with the minimum nonnegative number (r) of residue class r (mod n) of the residue system S_n.
2. At level-2, all peers having the same resource type R_r will form the group G_r (i.e., the subset P^{Rr}) with the group-head P_r connected to the transit ring network. The first peer to form the group is assigned the overlay address r. Each new peer joining group G_r is given the group membership address $(r + j \cdot n)$, for $j = 1,2,\ldots$, where these addresses are the consecutive positive integers starting with r belonging to residue class $\{r\}$. Based on Lemma 1, all these addresses are mutually congruent. According to the research work reported in [10], any two peers in a group with assigned logical addresses that are congruent to each other will be directly connected to each other via an overlay link. Therefore, all peers in any group are directly connected to each other. That is, the network of peers belonging to any group has a diameter of 1.
3. Resource type R_r possessed by peers in G_r is assigned the code r which is also the logical address of the group-head P_r of group G_r.
4. Each time a new group-head joins, a corresponding tuple ⟨Resource Type, Resource Code, Group Head Logical Address⟩ is entered in the global resource table (GRT).

Definition 3. *Two peers P_i and P_j on the ring network are logically linked together if $(i+1) \bmod n = j$.*

Remark 1. *The last group-head P_{n-1} and the first group-head P_0 are neighbors based on Definition 3. It justifies that the transit network is a ring.*

Definition 4. *Two peers of a group G_r are logically linked*

Algorithm 1 A Non-capacity constrained approach

```
Group-head P_i broadcasts the query ⟨p', a_r⟩ to all other group-heads of the P2P network   ▷ one hop communication
Some Group-head P_r finds the resource type r in its group G_r
if the group-head P_r has the answer a_r to the query then   ▷ Group-head P_r executes data look-up in its group
    it unicasts a_r to the requesting peer p'
    search ends   ▷ search ends with success
else
    P_r broadcasts the query ⟨p', a_r⟩ to peers in G_r   ▷ diameter of any cluster is one hop
    if ∃ p* with a_r in G_r then
        peer p* unicasts a_r to p'
        search ends   ▷ search ends with success
    else
        search fails and ends
    end if
end if
```

together if their assigned logical addresses are mutually congruent.

Lemma 2. *Diameter of the transit ring network is* $n/2$.

Lemma 3. *Each group* G_r *forms a complete graph.*

4 Modified Architecture

In the proposed modified architecture, the only change occurs at level 1: we replace the ring-based topology with a fully connected (complete) network composed exclusively of group-heads. This ensures that each group-head can communicate directly with every other group-head at the same level, reducing lookup latency and improving fault tolerance.

As detailed in Section 3, at level 1 each group-head P_r of group G_r is assigned a unique identifier equal to the smallest nonnegative representative of its residue class $r \bmod n$ in the residue system S_n. Equivalently, the overlay (logical) addresses of the n group-heads are assigned in consecutive order:

$$0, 1, 2, 3, \ldots, (n-1).$$

These are logical identifiers used for overlay routing and resource discovery (not physical network addresses), yielding a uniform and deterministic mapping across all group-heads within the complete level-1 network.

Consider the following series (sequence) of nonnegative integers:

$$i, i+d, i+2d, i+3d, \ldots, i+kd, \ldots$$

We observe that this is an AP (Arithmetic Progression) series with first term i and common difference d. This series may be finite or infinite.

Theorem 1. *Any two numbers in the series are mutually congruent.*

Proof. Without any loss of generality, let us consider the mth and rth numbers of the series. Let these be N_m and N_r. These numbers can be written as:

$$N_m = i+(m-1)d \quad \text{and} \quad N_r = i+(r-1)d$$

Now,

$$\frac{N_m - N_r}{d} = \frac{(i+(m-1)d) - (i+(r-1)d)}{d}$$
$$= \frac{(m-r)d}{d} = (m-r)$$

Since $(m-r)$ is an integer, $\therefore N_m \equiv N_r \pmod{d}$.

Since the congruence relation is symmetric, $\therefore N_r \equiv N_m \pmod{d}$.

Hence, any two numbers N_m and N_r of the series are mutually congruent with respect to modulus d. □ □

Corollary 1. *Any two numbers of the series* $0, 1, 2, 3, \ldots, (n-1)$ *are mutually congruent.*

It may be noted that the series considered in the above corollary is a finite AP series with first term 0 and common difference 1.

Observation 1. *Since the respective overlay (logical) addresses of the n number of group-heads are* $0, 1, 2, 3, \ldots, (n-1)$*, hence at level-1 group-heads are pairwise directly connected to each other via overlay links (Corollary 1). Therefore, at level-1, it is a complete network of group-heads with diameter* 1.

Corollary 2. *Network of peers belonging to any group has a diameter of* 1. *Hence, the diameter of the two-level Residue Class-based peer-to-peer network is* 3.

The modified architecture is shown in Figure 2. Note that the level-1 is a complete network.

5 Data Look-up Protocols

The proposed architectural modification involves only the level-1 portion of the P2P network [10]. Hence the data look-up protocols inside the groups do not need any modification.

Algorithm 2 Capacity-Constrained Group-Head Discovery

1: Group-head P_i broadcasts the query $\langle p', a_r \rangle$ to c number of group-heads in increasing sequence of their ids ▷ overlay address of a group-head is its id
2: **if** a receiving group-head P_r finds the resource type r in its group G_r **then**
3: P_r executes Step 3 of the 'non-capacity constrained approach'
4: **else**
5: P_r broadcasts the query $\langle p', a_r \rangle$ to a second set (in sequence) of c number of group-heads
6: **end if**
7: Step 2 is repeated until the search fails ▷ P_r executes at most n/c number of sends ▷ maximum number of hops (repetitions) is n/c

However, data -look-up in the whole P2P network needs some modification. Same is true for the whole network-wide broadcast of any information.

5.1 Data Look-up in the whole P2P network

Let a peer p' in group G_i look for some instance a_r of a resource type r and it is not present in the group. The query is denoted as $\langle p', a_r \rangle$. The group-head P_i executes the following as shown in Algorithm 1:

Remark 2. *The maximum number of hops required in the non-capacity constrained approach is* $[1(step1)+2]$.

In Step 1 of the above non-capacity constrained approach, peer heterogeneity has not been considered, especially from the viewpoint of the capacity (upload bandwidth) of a peer [22]. Below we present a capacity-constrained query look-up scheme initiated by the group-head P_i. Let its capacity be c. The algorithm is shown in Algorithm 2.

Remark 3. *The maximum number of hops required in the capacity constrained approach is* $[\frac{n}{c}+2]$.

5.2 Broadcast in the whole P2P network

This section introduces the broadcast algorithms tailored to address both non-capacity-constrained and capacity-constrained configurations in the system.

5.2.1 A non-capacity constrained approach

In this section state broadcast scheme only at the level-1 of the network. At level-2 existing schemes [10] apply. The algorithm is shown in Algorithm 3.

Algorithm 3 A non-capacity constrained approach

1: Given: Group-head P_i wants to broadcast some information I_{info} to all other group-heads of the network
2: P_i broadcasts I_{info} to all group-heads of the Federation ▷ One hop communication

5.2.2 A capacity constrained approach

In this section we present the broadcast scheme only at the level-1 of the network. At level-2 existing schemes [10] apply. The algorithm is shown in Algorithm 4.

Algorithm 4 Capacity-Constrained Broadcast at Level 1

1: Group-head P_i broadcasts the information I_{info} to c number of group-heads in increasing sequence of their ids ▷ One hop communication
2: Repeat Step 1 with a new set of c number of group-heads (in sequence) until all group-heads have received the broadcast information I_{info} ▷ P_i executes at most n/c number of sends

Remark 4. *The number of overlay hops required in the capacity constrained approach at Level 1 is* $\frac{n}{c}$.

6 Conclusions

In this work, we have presented a *modified* version of the existing Residue-Class (RC) based peer-to-peer (P2P) network architecture. The primary design objective of this modification is to reduce the network diameter to the smallest possible value without compromising the intrinsic advantages of the RC-based approach. The existing RC-based architecture [10] was chosen as the baseline for our work because of two key strengths: (1) all peers sharing the same interest or possessing the same resource type are structurally organized into a group (or cluster) of diameter one, thereby enabling *constant-time* intra-group lookups, and (2) the group heads are connected in the form of a logical ring, ensuring that the inter-group backbone remains connected even under churn conditions, thereby preserving network stability and routing guarantees.

Despite these strengths, the ring topology inherently imposes a network diameter of $\frac{n}{2}$ for the n group heads in the worst case. Consequently, the overall P2P network diameter becomes $\left(\frac{n}{2}+2\right)$ hops, where the additional two hops account for traversals between a peer and its group head at both ends of the communication path. As n grows, this diameter significantly affects lookup latency, particularly in time-critical or real-time applications such as industrial IoT monitoring, distributed learning, or telemedicine data retrieval.

In the proposed architecture, we replace the ring-based backbone with a fully connected inter-group head topology. Each group head maintains a direct link to all other group heads, effectively transforming the inter-group overlay from a diameter of $\frac{n}{2}$ to a diameter of one. Consequently, the overall network diameter is reduced to three hops:

1. One hop from the source peer to its local group head.
2. One hop directly between the source group head and the destination group head.
3. One hop from the destination group head to the target peer within its group.

This reduction is mathematically optimal for such architectures and represents the shortest possible diameter in a multi-group P2P system while maintaining structural separation of interest-based clusters.

The impact of this modification is substantial. By minimizing the maximum path length, we not only reduce the worst-case lookup latency but also decrease message forwarding overhead, improve throughput, and increase responsiveness under high query loads. Furthermore, the complete connectivity among group heads enhances fault tolerance—if a direct link fails, multiple alternative shortest paths still exist without affecting the network diameter.

Immediate future work will focus on designing secure and efficient data communication protocols tailored for this low-diameter RC-based architecture. Specifically, we aim to investigate:

- Lightweight cryptographic schemes to ensure confidentiality, integrity, and authenticity during inter-group communication without introducing significant latency.
- Trust verification mechanisms between group heads to prevent routing manipulation or malicious data injection.
- Adaptive maintenance strategies to dynamically manage backbone connections in large-scale deployments.

Through these enhancements, the proposed architecture is expected to provide a foundation for high-performance, secure, and scalable P2P systems applicable to diverse domains such as real-time distributed analytics, federated learning, and privacy-preserving healthcare networks.

References

[1] L. Badis, M. Amad, D. Aissani, K. Bedjguelal, and A. Benkerrou. Routil: P2p routing protocol based on interest links. In Proc. 2016 Int. Conf. Advanced Aspects of Software Engineering, pages 1–5, Constantine, 2016.

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. In Proceedings of ACM SIGCOMM, Karlsruhe, Germany, August 2003.

[3] Shiping Chen, Baile Shi, Shigang Chen, and Ye Xia. Acom: Any-source capacity-constrained overlay multicast in non-dht p2p networks. IEEE Transactions on Parallel and Distributed Systems, 18(9):1188–1201, September 2007.

[4] Wen-Tsuen Chen, Chi-Hong Chao, and Jeng-Long Chiang. An interested-based architecture for peer-to-peer network systems. In 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06), pages 707–712, Vienna, 2006.

[5] P. Ganesan, Q. Sun, and H. Garcia-Molina. Yappers: A peer-to-peer lookup service over arbitrary topology. In Proc. IEEE INFOCOM, volume 2, pages 1250–1260, San Francisco, USA, March 30–April 1 2003.

[6] Bidyut Gupta, Nick Rahimi, Henry Hexmoor, Shahram Rahimi, Koushik Maddali, and Gongzhu Hu. Design of very efficient lookup algorithms for a low diameter hierarchical structured peer-to-peer network. In Proc. IEEE 16th Int. Conf. Industrial Informatics (INDIN), Porto, Portugal, July 2018.

[7] Bidyut Gupta, Nick Rahimi, Shahram Rahimi, and Ashraf Alyanbaawi. Efficient data lookup in non-dht based low diameter structured p2p network. In Proc. IEEE 15th Int. Conf. Industrial Informatics (INDIN), pages 944–950, Emden, Germany, July 2017.

[8] M. Hai and Y. Tu. A p2p e-commerce model based on interest community. In 2010 International Conference on Management of e-Commerce and e-Government, pages 362–365, Chengdu, 2010.

[9] M. He, Y. Zhang, and X. Meng. Gossip-based resource location strategy in interest community for p2p networks. Chinese Journal of Electronics, 24(2):272–280, April 2015.

[10] Swathi Kaluvakuri, Koushik Maddali, Nick Rahimi, Bidyut Gupta, and Narayan Debnath. Generalization of rc-based low diameter hierarchical structured p2p network architecture. International Journal of Computer Applications, 27(2):74–83, June 2020.

[11] Mujtaba Khambatti, Kyung Ryu, and Partha Dasgupta. Structuring peer-to-peer networks using interest-based communities. In Lecture Notes in Computer Science, 1st International Workshop, DBISP2P 2003. Berlin, 2003.

[12] S. K. Khan and L. N. Tokarchuk. Interest-based self organization in cluster-structured p2p networks. In Proc. 6th IEEE Consumer Communications and Networking Conf, pages 1–5, Las Vegas, 2009.

[13] S. K. A. Khan and L. N. Tokarchuk. Interest-based self organization in cluster-structured p2p networks. In 2009 6th IEEE Consumer Communications and Networking Conference, pages 1–5, Las Vegas, NV, 2009.

[14] M. Kleis, E. K. Lua, and X. Zhou. Hierarchical peer-to-peer networks using lightweight superpeer topologies. In Proc. IEEE Symposium on Computers and Communications, pages 1–6, 2005.

[15] D. Korzun and A. Gurtov. Hierarchical architectures in structured peer-to-peer overlay networks. In Peer-to-Peer Networking and Applications, pages 1–37. Springer, March 2013.

[16] Z. Peng, Z. Duan, J. J. Qi, Y. Cao, and E. Lv. Hp2p: a hybrid hierarchical p2p network. In Proc. Intl. Conf. Digital Society, pages 86–90, 2007.

[17] N. Rahimi, K. Sinha, B. Gupta, and S. Rahimi. Ldepth, a low diameter hierarchical p2p network architecture. In Proc. IEEE 14th Int. Conf. Industrial Informatics (INDIN), pages 832–837, Poitiers, France, July 2016.

[18] A. Rowstron and P. Druschel. Pastry, scalable, distributed object location and routing for large scale peer-to-peer systems. In Proc. FIP/ACM International Conference on Distributed Systems Platforms (Middleware), pages 329–350, 2001.

[19] Indranil Roy, Bidyut Gupta, Banafsheh Rekabdar, and Henry Hexmoor. A novel approach toward designing a non-dht based structured p2p network architecture. In EPiC Series in Computing, volume 63, pages 121–129, 2019. Proceedings of 32nd Int. Conf. Computer Applications in Industry and Engineering.

[20] Indranil Roy, Swathi Kaluvakuri, Koushik Maddali, Ziping Liu, and Bidyut Gupta. Efficient communication protocols for non dht-based pyramid tree p2p architecture. WSEAS Transactions on Computers, 20:108–125, July 2021. (Invited paper).

[21] Indranil Roy, Koushik Maddali, Swathi Kaluvakuri, Banafsheh Rekabdar, Ziping Liu, Bidyut Gupta, and Narayan Debnath. Efficient any source overlay multicast in crt-based p2p networks - a capacity-constrained approach. In Proc. IEEE 17th Int. Conf. Industrial Informatics (INDIN), pages 1351–1357, Helsinki, Finland, July 2019.

[22] H. Shen, G. Liu, and L. Ward. A proximity-aware interest-clustered p2p file sharing system. IEEE Transactions on Parallel and Distributed Systems, 26(6):1509–1523, June 2015.

[23] H. Shen, G. Liu, and L. Ward. A proximity-aware interest-clustered p2p file sharing system. IEEE Transactions on Parallel and Distributed Systems, 26(6):1509–1523, June 2015.

[24] K. Shuang, P. Zhang, and S. Su. Comb: resilient and efficient two-hop lookup service for distributed communication system. Security and Communication Networks, 8(10):1890–1903, 2015.

[25] R. I. Stocia, R. Morris, D. Liben-Nowell, D. R. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Transactions on Networking, 11(1):17–32, February 2003.

[26] Z. Tu, W. Jiang, and J. Jia. Hierarchical hybrid dve-p2p networking based on interests clustering. In 2017 International Conference on Virtual Reality and Visualization (ICVRV), pages 378–381, Zhengzhou, China, 2017.

[27] M. Xu, S. Zhou, and J. Guan. A new and effective hierarchical overlay structure for peer-to-peer networks. Computer Communications, 34:862–874, 2011.

[28] M. Yang and Y. Yang. An efficient hybrid peer-to-peer system for distributed data sharing. IEEE Transactions on Computers, 59(9):1158–1171, September 2010.

[29] B. Y. Zhao, L. Huang, S. C. Rhea, J. Stribling, A. Zoseph, and J. D. Kubiatowicz. Tapestry: A global-scale overlay for rapid service deployment. IEEE Journal on Selected Areas in Communications (J-SAC), 22(1):41–53, January 2004.

Indranil Roy (photo not available) is an Assistant Professor in the Department of Computer Science at the Southeast Missouri State University. He received his MS and Ph.D. degrees in Computer Science from Southern Illinois University, Carbondale in 2018 and 2022, respectively. His current research interest includes the design of architecture and communication protocols for structured peer-to-peer overlay networks, security in overlay networks, and Blockchain.

Reshmi Mitra (photo not available) is an Associate Professor in the Department of Computer Science at the Southeast Missouri State University. She received her MS and Ph.D. degrees in Electrical and Computer Engineering from the University of North Carolina at Charlotte in 2007 and 2015, respectively. Previously she has worked at the National Institute of Technology India, Advanced Micro Devices Austin, and Samsung Austin R&D Center. Her research interests include Security and Performance issues in IoT, Cloud Computing, and Blockchain.

Bidyut Gupta (photo not available) received his M. Tech. degree in Electronics Engineering and Ph.D. degree in Computer Science from Calcutta University, Calcutta, India. At present, he is a professor at the School of Computing (formerly Computer Science Department), Southern Illinois

University, Carbondale, Illinois, USA. His current research interest includes design of architecture and communication protocols for structured peer-to-peer overlay networks, security in overlay networks, and block chain. He is a senior member of IEEE and ISCA.

Narayan Debnath (photo not available) earned a Doctor of Science (D.Sc.) degree in Computer Science and also a Doctor of Philosophy (Ph.D.) degree in Physics. Narayan C. Debnath is currently the Founding Dean of the School of Computing and Information Technology at Eastern International University, Vietnam. He is also serving as the Head of the Department of Software Engineering at Eastern International University, Vietnam. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA) since 2014. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years (1989-2017). Dr. Debnath has been an active member of the ACM, IEEE Computer Society, Arab Computer Society, and a senior member of the ISCA.

Detection of Academic Dishonesty in Student Records Using Anomaly Detection with Deep Learning and Machine Learning Techniques

[1,*]Manit Malhotra, [2] Indu Chhabra

[1]Department of Computer Science & Applications, Panjab University, Chandigarh, India
Email: manitmalhotra@rediffmail.com

[2]Department of Computer Science & Applications, Panjab University, Chandigarh, India
Email: indu_c@pu.ac.in

Abstract

The rapid transition to online education, particularly during the COVID-19 pandemic, has raised critical concerns about maintaining academic integrity in online assessments. Traditional proctoring methods have proven insufficient, prompting exploration of advanced technological solutions. This paper investigates machine learning (ML) and deep learning (DL) techniques for detecting cheating behaviors using mark sheet data only, drawn from a cohort of 3,000 students across six semesters (three online, three offline) at Panjab University constituent colleges. The dataset exhibited a subtle vs blatant cheating imbalance ratio of approximately 3:2, handled via hybrid resampling. Models were evaluated using an 80/20 train-test split with bootstrapped confidence intervals. Among four models tested, the BiLSTM outperformed others, achieving 97.6% accuracy and an AUC of 1.00, offering absolute gains of +1.6% accuracy over LSTM and +22% over Random Forest. These findings highlight the potential of DL models for scalable, automated cheating detection; however, the scope remains limited to numerical mark sheet features from a single institution, warranting broader future studies.

Key Words: Anomaly Detection, Cheating Detection, Academic Dishonesty Detection , Deep Learning, Machine Learning

1 Introduction

The shift to online education, rapidly accelerated by the COVID-19 pandemic, has transformed traditional learning environments by offering unprecedented accessibility and flexibility [1]. However, this shift has also introduced substantial challenges for maintaining academic integrity, particularly in assessments conducted without direct invigilation [2]. Concerns about cheating during online exams have grown as students gain easier access to unauthorized resources or collaborative means that were harder to exploit in proctored, in-person settings [3], [4].

While numerous proctoring systems now leverage multimodal data streams, such as webcam feeds, eye-gaze tracking, or typing dynamics to curb dishonest behaviors, such approaches demand considerable infrastructure and raise privacy concerns [5], [6]. This study instead focuses on a simpler, cost-effective question: *Can temporal trends in students' mark sheet data alone reliably differentiate between subtle and blatant cheating behaviors, outperforming chance and traditional baselines?* By exploring purely numerical marks across online and offline semesters, we aim to evaluate whether even minimal data can provide an effective first line of detection.

1.1 Rise of Virtual Learning and Associated Cheating Trends

The expansion of online education is undeniable. The number of Americans enrolled in distance education courses rose by 93% between 2012 and 2019, according to research published by the National Center for Education Statistics (NCES) [7]. However, while this growth has enabled educational institutions to reach a wider audience, it has also created new vulnerabilities [8]. Without direct supervision, students are more likely to engage in dishonest practices such as using unauthorized materials, collaborating with peers, or even outsourcing their work to third parties [9]. The temptation to cheat has become more prevalent in online exams, where students may perceive that the risk of detection is lower compared to traditional, proctored exams [10], [11].

Research indicates that learners are more inclined to cheat when they think they can get away with it easily. The nature of online exams, which often allow students to take tests from the comfort of their homes, exacerbates this issue [12]. Other studies have found that the rate of academic dishonesty during online exams is significantly higher than in-person exams, particularly in assessments that do not employ stringent monitoring mechanisms. The same study highlighted the difficulty of preventing cheating in an environment where students can easily access external resources, share answers with peers, or utilize advanced technological tools to bypass detection [1].

1.2 Technological Approaches to Cheating Detection

The development of automated cheating detection systems has become a critical area of focus in educational research.

A key approach to detecting cheating involves the use of algorithms to identify anomalies in student behavior and performance. Levitt et al. [13] introduced a simple yet effective algorithm for identifying cheating by analyzing patterns of incorrect answers among students seated next to each other during in-person exams. Their findings demonstrated that matching incorrect answers was a more reliable indicator of cheating than matching correct answers, especially in scenarios where students were seated in proximity to one another. When applied to online exams, this approach can be adapted to detect patterns of anomalous behavior, such as unusually similar responses between students, which may indicate collusion [14]. Cheating detection has seen a rise in the usage of ML and DL methods in addition to algorithmic approaches. These advanced methods can be employed to analyze a wider range of behavioral indicators, such as eye movements, typing patterns, and camera recordings, to identify potential instances of cheating. The utilization of these techniques can help eliminate the need for manual review of student assessment sessions, thereby streamlining the process of detecting academic dishonesty. However, recent research [15], [14] has highlighted the potential for bias and fairness issues in the implementation of automated proctoring systems. Disparities have been observed in the accuracy of these systems across different racial, skin tone, and gender groups, raising concerns about the equitable treatment of students. To address these challenges, it is crucial for educators and researchers to carefully design and evaluate the algorithms and models used in cheating detection systems, ensuring that they are fair, accurate, and transparent.

In online learning environments, the challenge of detecting cheating is compounded by the lack of physical supervision. This has led to the exploration of ML techniques, particularly for identifying cheating behaviors in large datasets. One such study, conducted by Kamlov et al. [15], proposes an ML-based approach to detecting cheating using outlier detection methods. In their research, they treat the identification of potential cheating cases as an outlier detection problem, leveraging student assessment data to identify abnormal scores on final exams. They successfully identified instances of cheating by using techniques such as anomaly detection methods and AI methods such as Recurrent Neural Networks (RNNs). The use of sequential data analysis is particularly relevant in online exams, where the order and timing of answers can provide critical insights into whether a student has engaged in dishonest behavior.

While real-time surveillance through videos or images has become an increasingly popular way to detect cheating in online exams, it is not always feasible in all educational settings. In many cases, particularly where technological resources are limited, initial stages of cheating detection can be performed using readily available data such as student mark sheets and answer sheets. This form of analysis is particularly useful when there is no dataset available for real-time observation. By evaluating student performance data through traditional grading systems, we can identify anomalies in answer patterns or sudden deviations in performance that may indicate cheating behavior. This approach provides a practical and resource-efficient solution for detecting cheating before more advanced, system-based detection methods are implemented. Moreover, analyzing answer sheets and mark sheets allows educators to spot suspicious behavior early on, such as repeated patterns of similar answers between students or drastic improvements in final scores compared to earlier assessments. Such methods are indispensable in environments where technological tools for real-time monitoring are unavailable, making them an effective initial line of defense in maintaining academic integrity.

Previous studies have explored various methods for detecting academic dishonesty using numerical datasets like student mark sheets, quizzes, and exam scores [15], [16], [13]. These studies' method treated cheating detection as an outlier detection problem, where significant deviations between a student's final exam performance and earlier assessments were flagged as potential cheating cases. Also, some studies [13] detect cheating based on answer patterns in exams. They found that comparing incorrect answers between students seated nearby was a more reliable indicator of cheating than comparing correct answers [16]. These studies highlight the importance of using numerical datasets like mark sheets in the initial stages of detecting academic dishonesty, especially in cases where more advanced data like images, videos, or behavioral metrics are not available.

In this research, we aim to build upon these studies by analyzing student mark sheets from a multidisciplinary dataset within an educational institution. Our focus is to explore whether patterns in the mark sheets can reveal irregularities that point toward cheating behaviors, particularly by identifying statistical outliers and comparing student performance across multiple subjects. The dataset spans several disciplines and contains a wide range of student marks, which allows us to investigate cheating behaviors from different academic contexts. By examining this data and using just conventional evaluation data, our objective is to aid in the development of resource-efficient cheating detection techniques.

Additionally, unlike many existing approaches that focus solely on performance scores, our research integrates multi-modal indicators to capture a broader spectrum of cheating behaviors. This includes mechanisms to detect covert forms of dishonesty, such as collusion through similarity analyses among peer submissions, irregular access patterns suggestive of unauthorized resource use, and timing irregularities indicative of technological manipulation. These enhancements aim to ensure that both overt and subtle cheating tactics are systematically addressed.

Unlike multimodal proctoring systems that rely on video feeds, keystroke dynamics, or eye-tracking to detect cheating, our work focuses solely on analyzing numerical mark sheet data. While multimodal systems can capture richer behavioral cues and are covered extensively in Section 2 and Table 1, they also demand substantial infrastructure, raise privacy concerns, and may not be feasible in all educational contexts. By contrast, our study demonstrates that even traditional mark trends can

reveal subtle and blatant cheating behaviors, offering a practical, resource-light alternative that can serve as either a first-line screening tool or complement more intrusive proctoring systems.

1.3 Research Questions

To systematically guide this investigation, we articulate the following research questions (RQs), each aligned with our dataset, methodology, and evaluation framework:

1. **RQ1:** Can trends in mark sheet data alone, without any proctoring images or videos, effectively differentiate between non-cheating, subtle cheating, and blatant cheating cases?
2. **RQ2:** How do advanced deep learning models (LSTM, BiLSTM) compare to traditional machine learning models (Random Forest, Logistic Regression) in detecting performance anomalies indicative of cheating?
3. **RQ3:** What is the absolute improvement in accuracy and AUC offered by BiLSTM over other baseline models on this dataset?
4. **RQ4:** Does handling class imbalance through hybrid resampling techniques enhance the detection capability of cheating versus using imbalanced raw data?
5. **RQ5:** Can automated cheating detection based solely on mark trends achieve consistent agreement with expert human validation, supporting practical deployment?

1.4 Major Contributions

The principal contributions of this research can be summarized as follows:

1. We developed and systematically evaluated a cheating detection framework based exclusively on mark sheet data trends, demonstrating that even without proctoring videos or behavioral data, automated systems can flag subtle and blatant cheating with high reliability.
2. We conducted a comprehensive comparative study of traditional ML models (Random Forest, Logistic Regression) against advanced DL models (LSTM, BiLSTM), revealing that BiLSTM yields an absolute improvement of +1.6% accuracy over LSTM and +22% over RF, with perfect AUC in our experiments.
3. We employed robust hybrid class imbalance handling (SMOTE + Tomek Links) and demonstrated how it improves model sensitivity to minority (blatant cheating) cases, enhancing fairness and detection reliability.
4. We incorporated a human-in-the-loop validation stage, showing a 94% agreement between expert educators and automated BiLSTM predictions, thereby building trust for real-world deployment.
5. We provided extensive feature importance (for RF) and integrated gradients (for BiLSTM) analyses, along with fairness checks across academic programs, to ensure transparency, explainability, and ethical AI deployment.

The research is organized to systematically explore the various aspects of cheating detection through mark sheet data analysis. The introduction of the study is provided in Section 1. In Section 2, related work is reviewed with an emphasis on current approaches to academic dishonesty detection. Section 3 explained how the dataset was gathered for compiling multidisciplinary student mark sheet data. Section 4 described the Proposed methodology of the study, which briefly explained the process of data preprocessing and feature engineering, and showed how the data was prepared for analysis. The results are presented and discussed in Section 5, followed by an evaluation of the study's limitations in Section 6 and suggestions for future research in Section 7. Finally, Section 8 concludes the paper by summarizing the key findings.

2 Related Work

2.1 The Applications and Significance of ML and DL in Cheating Detection

To combat the rise of cheating in online assessments, educational institutions have increasingly turned to technological solutions. Using machine learning and DL algorithms, which can analyze massive amounts of data and spot trends that can point to dishonest behavior, is one of the most intriguing strategies [17], [18],[15]. A subfield of artificial intelligence called machine learning gives computers the ability to learn from data without the need for explicit programming. This makes it ideal for detecting cheating, as the algorithms can be trained to recognize suspicious behavior based on historical data [19].

Techniques related to ML have been effectively used in a number of industries, such as financial modelling, healthcare, and cheating detection. In the context of education, these techniques have been adapted to detect cheating by analyzing exam performance data, monitoring students' actions during exams, and identifying inconsistencies that may suggest dishonesty [20]. For instance, these algorithms can be used to compare a student's performance in online and offline exams, flagging significant discrepancies that may indicate cheating. Additionally, these models can be applied to identify patterns of suspicious behavior, such as frequent switching between exam windows or prolonged inactivity followed by rapid answering of questions [15],[21].

DL, a more advanced subset of ML, is particularly effective in cheating detection because of its ability to analyze complex, unstructured data such as images and video. Deep learning models such as convolutional neural networks (CNNs) and long short-term memory (LSTM) networks have been applied in the context of online exam monitoring. These models are capable of analyzing visual cues like facial expressions, gaze direction, and behavioral patterns that may indicate potential academic misconduct. These models can be trained

to detect subtle changes in a student's behavior, providing a more comprehensive view of potential cheating activities [5], [20],[22].

2.2 Existing Approaches to Cheating Detection

The literature on cheating detection is extensive, with researchers exploring various methods to address academic dishonesty. Early approaches relied on rule-based systems, which used predefined criteria to flag suspicious behavior. For example, systems might identify students who submit exams significantly faster than average or those who exhibit unusual patterns of behavior during an exam. While these methods were effective to some extent, they were limited by their reliance on static rules, which could be easily bypassed by students who were aware of the system's criteria [23],[24],[1],[25].

The existing literature on cheating detection methods highlights the evolution of techniques across various domains, including text, numerical data, and coding submissions. AlSallal et al. [26] emphasize the limitations of traditional plagiarism detection tools, which often fail to recognize reworded or summarized texts. They introduce a comprehensive method that integrates Bag of Words (BoW), Latent Semantic Analysis (LSA), Stylometric Features, and Support Vector Machines (SVM) to identify complex instances of plagiarism. The effectiveness of this approach in recognizing nuanced linguistic patterns was demonstrated through experiments on the Corpus of English Novels (CEN). Similarly, Atoum et al. [5] address the challenges of cheating in online exams, proposing an automated detection system that integrates hardware (webcams, microphones) and software (gaze estimation, voice detection) to analyze audio-visual data in real-time, significantly reducing the reliance on human proctoring.

Nagoudi et al. [27] focus on Arabic texts, where traditional tools fail to identify synonym substitution and text manipulation effectively. They propose two methods, one leveraging word embeddings and the other using ML classifiers like SVM and RF to detect disguised plagiarism with high precision and recall scores, using the EXARA-2015 dataset. Meanwhile, Qiubo et al. [28] tackle code plagiarism, introducing a hybrid approach that combines RF and gradient boosting decision trees. Their methods adapt to variations in programming style and submission behavior, achieving a notable accuracy rate of 95.9% in identifying plagiarized code. These innovative approaches demonstrate the growing sophistication in detecting various forms of cheating, offering solutions that extend beyond simple similarity thresholds to incorporate behavioral and semantic analysis across diverse data types.

More recent approaches have incorporated ML and DL techniques, which offer greater flexibility and accuracy. These techniques make it possible to analyze big datasets, including student performance records, behavioral data, and even biometric information. By leveraging these data sources, these algorithms are able to identify cheating tendencies that would be difficult to find using conventional techniques. For example, a study by Tiong et al. [1] demonstrated the efficiency of a DL based method for detecting cheating in online exams. Their system, which analyzed both behavioral and performance data, achieved a high level of accuracy in identifying students who engaged in dishonest practices [29]. A primary benefit of using ML and DL in cheating detection is the capacity for ongoing enhancement of the system's accuracy over time. As more data are collected, the algorithms can be retrained to better recognize cheating behaviors. This allows the system to adapt to new forms of cheating, making it more difficult for students to find ways to circumvent detection [4].

In addition to the ML and DL approaches used in the detection of cheating behaviors during exams, many researchers have used IoT technologies to detect these kinds of behavior. IoT technology offers a budget-friendly, flexible, and easy-to-use solution to detect cheating in online examinations. These systems rely on simple devices like webcams, microphones, and internet access to function. The growth of AI-based proctoring tools has become especially significant during the COVID-19 pandemic, as digital learning has gained widespread use around the world [30].

Recent advancements in online proctoring and cheating detection have introduced a variety of innovative systems and methodologies. Shevale et al. [31] developed a web-based application using the MERN stack, enhancing accessibility for disabled users with features like voice navigation, while also focusing on securing online exams during the pandemic. Ho et al. [32] created RAPID, a proctoring solution using Raspberry Pi to monitor students' computer activity, helping prevent cheating through advanced security measures. Nguyen et al. [6] developed a system that leverages IoT and AI, utilizing dual cameras and AI-driven analysis for real-time fraud detection in online exams, achieving high accuracy. La Roca et al. [33] examined students' experiences with online proctoring, finding that while most have adequate resources, some face challenges that affect their performance. Plochaet et al. [34] investigated video anomaly detection to monitor academic integrity, employing video and voice detection models to spot cheating, although the results showed some room for improvement. Atabay et al. [35] studied BeatGAN, an ML model for time series anomaly detection, which showed promise but struggled with subtle cheating behaviors. Finally, Bommireddy et al. [36] designed an AI-driven system that monitored students through video and audio inputs to detect cheating, reducing the reliance on human proctors. Table 1 represents the summary.

Table 1: Summary of Cheating Detection Approaches

Reference No.	Characteristics	Models Used	Limitations
(Gruenigen et al. [23], Chuang et al. [25], Keresztury et al. [24])	Early approaches relied on rule-based systems, using predefined criteria to flag suspicious behavior, such as unusually fast exam submissions.	Rule-Based Systems	Static rules could be bypassed by students familiar with the system, limiting effectiveness.
(AlSallal et al. [26])	Focus on detecting sophisticated plagiarism attempts using a multifaceted approach.	BOW, LSA, SVM	Traditional tools fail to recognize reworded or summarized texts, and performance depends on text quality.
(Atoum et al. [5])	Automated system integrating hardware (webcams, microphones) and software (gaze estimation, voice detection) for real-time analysis.	Gaze Estimation, Voice Detection	Relies on external devices and real-time monitoring, may raise privacy concerns or require infrastructure.
(Cherroun et al. [27])	Focus on Arabic texts for detecting disguised plagiarism using word embeddings and ML classifiers.	Word Embeddings, SVM, Random Forest	Ineffective at detecting synonym substitution or manipulation in non-English.
(Qiubo et al. [28])	Hybrid approach combining RF and GBDT for code plagiarism detection.	RF, GBDT	Challenges with highly sophisticated or novel cheating techniques.
(Tiong et al. [1])	Recent methods using ML and DL to detect cheating via performance and behavioral data.	DL	Requires significant data for training, newer tactics may bypass detection.
(Zhao et al. [4])	ML/DL systems retrain on new data to adapt to evolving behaviors.	ML, DL	Continuous retraining needed, models may become outdated if methods change.
(Nigam et al. [30])	IoT offers a cost-effective, flexible solution using webcams/mics.	IoT Systems	Potential false positives and limited by simple devices.
(Shevale et al. [31])	MERN stack web app for accessible, secure exams during the pandemic.	MERN stack	May face accessibility/security challenges at scale.
(Ho et al. [32])	RAPID system uses Raspberry Pi to monitor computer activity.	Raspberry Pi	Hardware constraints, may struggle with many students.
(Nguyen et al. [6])	IoT + AI dual cameras for real-time fraud detection.	Dual Cameras, AI	Costly at scale, struggles with complex cheating.
(Roca et al. [33])	Study on student experiences highlights resource challenges.	Online proctoring	Insufficient resources may cause false negatives.
(Plochaet et al. [34])	Video + voice detection for academic integrity.	Video anomaly, Voice detection	Needs improvement for subtle cheating.
(Atabay et al. [35])	BeatGAN for time-series anomaly detection.	BeatGAN	Needs optimization for real-time, misses subtle cheating.
(Bommireddy et al. [36])	AI system monitors via video/audio for cheating.	AI Video/Audio	Raises privacy concerns, may miss sophisticated cheating.
(Roumiana et al. [17], Fakhroddin et al. [18])	ML/DL detect cheating in online assessments.	ML algorithms	Depend on quality/quantity of historical data.
(Sarker et al. [19])	ML learns from data, ideal for detecting cheating.	ML	May not generalize well to new cheating types.
(Kaddoura et al. [20])	ML detects cheating via performance/action data.	ML	May struggle with sophisticated strategies.
(Kamlov et al. [15], Balderas [21])	Compares online vs offline performance, flags suspicious patterns.	ML	May flag non-cheating behaviors.
(Faucher et al. [22])	DL analyzes unstructured data (images/video) for cheating.	CNNs, LSTM	Computationally intensive for real-time.
(Cizek and Wollack [37])	Reviews statistical indices for test score cheating.	Statistical Indices (K-index, GBT)	Needs item-level data, assumes independence.

Table 2: Comparison of Datasets and Validation Strategies Across Studies

Study	Sample Size	Data Modalities	Validation Protocol	Remarks
Atoum et al. [5]	∼100 students	Webcam Video, Gaze, Voice	Expert labeling, no split reported	Requires external hardware
Cherroun et al. [27]	2,000 text pairs	Arabic text documents	Train-test split, no stratification	NLP-specific, not exam-based
Tiong et al. [1]	300+ sessions	Screen logs, Webcam, Responses received	Manual review, 5-fold CV	Multimodal behavioral dataset
Our Proposed Model	**2,931 records**	**mark sheet data (numeric)**	**Stratified CV**	**Imbalanced multiclass-to-binary setting**

Although the integration of IoT, AI, and other technologies for cheating detection is promising, there are several challenges to be addressed. First, technological limitations can lead to false positives or negatives. For example, a student may be flagged for cheating due to nervous habits, such as fidgeting or looking around the room, even when no dishonest behavior occurs. However, sophisticated cheating methods, such as the use of hidden ears or advanced signal jamming techniques, can evade detection. Second, the cost of deploying IoT infrastructure and AI systems at scale can be prohibitive for many educational institutions, especially in resource-constrained environments. The complexity of integrating different technologies into a cohesive cheating detection framework also requires significant expertise and investment.

While Table 1 summarizes methodological differences among approaches, we now extend the comparison with a dedicated overview of sample sizes, feature modalities, and validation strategies. This helps contextualize the relative difficulty of our setting, which relies solely on numerical mark sheet data without multimodal augmentation. As seen in Table 2, many prior studies used behavioral video/audio inputs or interaction logs, often with smaller or curated datasets, while our setting involves over 2,900 labeled student records across six semesters and three disciplines with class imbalance, making it a non-trivial detection challenge.

While much recent work on cheating detection emphasizes multimodal data such as video or biometric streams, a parallel line of research, statistical forensics, continues to investigate cheating via performance data alone. For example, Cizek and Wollack [37] provide a comprehensive overview of statistical indices (e.g., K-index, GBT) designed to flag collusion or answer copying based on anomalous score distributions and response patterns.

3 Dataset

In order to understand potential cheating behaviors among students, this section offers a comprehensive examination of the dataset used in our study. The dataset comprises detailed academic records from multiple undergraduate programs, including Bachelor of Commerce (B.Com.), Bachelor of Business Administration (B.B.A.), and Bachelor of Computer Applications (B.C.A.), offered across various constituent colleges of Panjab University, Chandigarh, India. This study focuses on student cohorts from the academic years 2020 to 2023, a period that captures the global transition from virtual learning during the COVID-19 pandemic to the subsequent resumption of in-person education.

The dataset covers six semesters for each student, with a notable division in exam formats: the first three semesters were conducted online, while the final three semesters were administered in traditional offline settings. Each semester's data includes student names, roll numbers, and the marks obtained in six subjects. This comprehensive

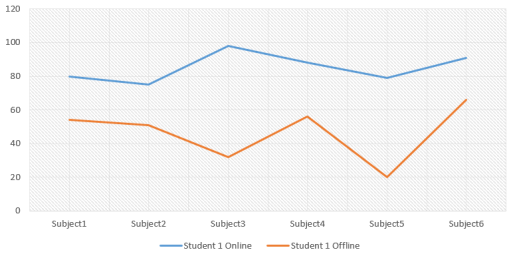


Figure 1: Anomaly Detection Graph for Online and Offline Exams of a Student

structure allows for a comparative analysis of performance across different exam modes.

The primary motivation for selecting this dataset is to investigate whether the switch to online exams may have contributed to irregularities in student performance, potentially indicative of cheating. By systematically comparing students' marks in online versus offline exams, we aim to identify patterns that suggest dishonest behavior. Figure 1 illustrates the creation of a line graph to visualize potential discrepancies between online and offline exams. The plotted graph, with separate lines, represents a single student's mark scenario for both offline and online exams. The graph indicates a general downward trend in marks over time, potentially due to factors like increased difficulty or decreased student effort. However, a significant disparity is observed between online and offline exams in all six subjects. This suggests the possibility of cheating or other irregularities in the online exam environment.

To facilitate supervised learning and behavioral analysis, we established a set of labeling rules based on the mark differences observed between online and offline exams for the same student. These rules are designed to reflect varying levels of potential academic dishonesty:

1. **Non-Cheating:** Students whose mark differences between online and offline exams are less than 25 points are labeled as "Non-Cheating", as they fall outside the defined suspicious thresholds associated with potential misconduct.
2. **Subtle Cheating:** Students whose online exam marks exceed their offline marks by 25 to 29 points are labeled as "Subtle Cheating", indicating possible minor or opportunistic misconduct.
3. **Blatant Cheating:** Students whose mark difference is 30 points or more are labeled as "Blatant Cheating", reflecting likely instances of significant academic dishonesty.

The dataset initially comprises 3,000 anonymized student records. These records span various programs and semesters, enabling the analysis of cheating patterns across disciplines and timeframes. We specifically focused on students who had appeared for both online and offline versions of the same or equivalent examinations between 2020 and 2023. Based on the above labeling criteria, we divided the dataset into three classes Non-Cheating, Subtle Cheating, and Blatant Cheating to facilitate multiclass classification and behavioral profiling. This structure allows us to investigate how the abrupt shift to virtual learning (and its eventual reversal) may have influenced academic integrity at scale.

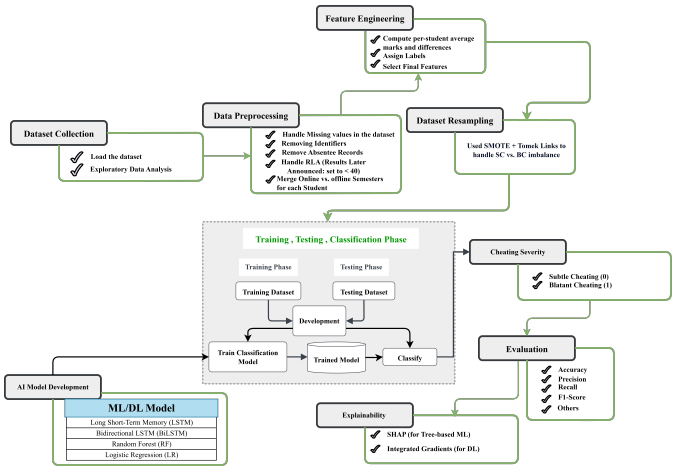


Figure 2: Development Pipeline of the Proposed Methodology

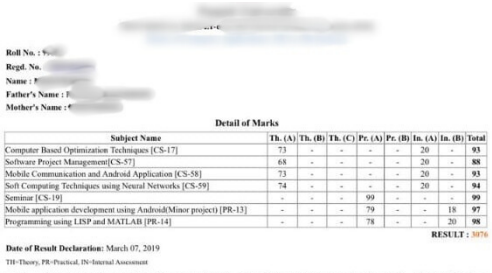
4 Proposed Methodology

The proposed methodology for this study revolves around classifying cheating and non-cheating instances using a carefully curated dataset. This methodology is structured into three major phases: preprocessing, feature engineering, and model training and testing. Several steps are involved in each phase to ensure efficient dataset processing, appropriate feature engineering, and accurate predictions from the classification model.

Our approach aims to comprehensively capture and identify cheating behaviors by analyzing discrepancies in students' marks between online and offline exams while addressing challenges such as missing data and class imbalances. The proposed methodology is structured into preprocessing, feature engineering, resampling, and training/testing phases using both ML and DL models, with explainability and fairness considerations integrated into the evaluation process shown in Figure 2.

4.1 Data Collection

The dataset contains academic records from undergraduate programs, including B.B.A., B.Com., and B.C.A., at Panjab University, Chandigarh, India. For the dataset collection, the website https://results.puexam.in/ has been accessed for publicly available datasets of the above-mentioned departments. It focuses on the academic years 2020 to 2023, covering six semesters for each student. The first three semesters were conducted online, and the final three were held offline. The data includes student names, roll numbers, and marks obtained in six subjects, allowing for a comparative performance analysis across different exam modes. Initially, each student's mark sheet was displayed as shown in Figure 3 and was later consolidated into an Excel file by combining all students' records into a single file. To maintain the transparency of the dataset and maintain the privacy of the students we removed the student names and roll numbers from our final dataset file as illustrated in Figure 4.



Roll No. :
Regd. No.
Name :
Father's Name :
Mother's Name :

Detail of Marks

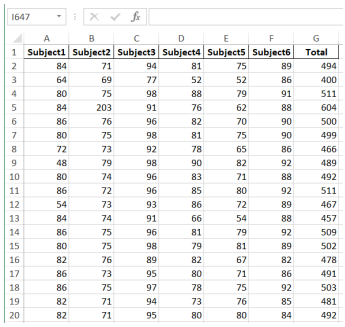
Subject Name	Th. (A)	Th. (B)	Th. (C)	Pr. (A)	Pr. (B)	In. (A)	In. (B)	Total
Computer Based Optimization Techniques [CS-17]	73	-	-	-	-	20	-	93
Software Project Management[CS-57]	68	-	-	-	-	20	-	88
Mobile Communication and Android Application [CS-58]	73	-	-	-	-	20	-	93
Soft Computing Techniques using Neural Networks [CS-59]	74	-	-	-	-	20	-	94
Seminar [CS-19]	-	-	-	99	-	-	-	99
Mobile application development using Android(Minor project) [PR-13]	-	-	-	79	-	-	18	97
Programming using LISP and MATLAB [PR-14]	-	-	-	78	-	-	20	98

RESULT :

Date of Result Declaration: March 07, 2019

TH=Theory, PR=Practical, IN=Internal Assessment

Figure 3: The Representation of Student's Mark Sheet Collected from the University's Website



	A	B	C	D	E	F	G
1	Subject1	Subject2	Subject3	Subject4	Subject5	Subject6	Total
2	84	71	94	81	75	89	494
3	64	69	77	52	52	86	400
4	80	75	98	88	79	91	511
5	84	203	91	76	62	88	604
6	86	76	96	82	70	90	500
7	80	75	98	81	75	90	499
8	72	73	92	78	65	86	466
9	48	79	98	90	82	92	489
10	80	74	96	83	71	88	492
11	86	72	96	85	80	92	511
12	54	73	93	86	72	89	467
13	84	74	91	66	54	88	457
14	86	75	96	81	79	92	509
15	80	75	98	79	81	89	502
16	82	76	89	82	67	82	478
17	86	73	95	80	71	86	491
18	86	75	97	78	75	92	503
19	82	71	94	73	76	85	481
20	82	71	95	80	80	84	492

Figure 4: Representation of the Transparent Dataset

4.2 Pre-Processing of Datasets

Data preprocessing is an essential stage that provides the structure for the next steps. In this stage, in order to improve the quality and quantity of the dataset, we performed data collection, data augmentation, handled values that were missing, and combined datasets before preparing the data for analysis.

4.2.1 Merging Discipline Datasets

The first task in preprocessing was to merge the datasets of three different disciplines: B.B.A, B.COM., and B.C.A., each dataset comprises student records across six semesters, with the same number of subjects in each discipline. The motivation behind merging these datasets is to increase the sample size, which enhances the study's generalizability and resilience. The merging process of the datasets allows us to analyze student behavior on a larger scale and draw more reliable conclusions regarding cheating patterns across different fields of study.

4.2.2 Merging Online and Offline Examination Records

To facilitate a clear comparison between online and offline examination performance, we consolidated the first three semesters (online exams) and the last three semesters (offline exams) of the dataset. This step ensures the following aspects:

Direct Comparison: Merging online and offline exam records allows for direct comparisons between the two exam modalities. This is critical because the study's primary goal is to identify discrepancies in student performance that may indicate cheating in the more lenient environment of online exams compared to the more controlled offline exams.

Uniform Dataset: Consolidating the data creates a uniform dataset that provides a holistic view of each student's performance across both modalities. This uniformity simplifies the subsequent analysis, ensuring that all records are complete and comparable across all six semesters.

Identifying Cheating Behaviors: This study's main objective is to identify patterns of cheating in exams. Merging the two types of exam records allows for the calculation of performance differences between the online and offline exams, which can help flag students who might have exploited the online format for dishonest gains.

4.2.3 Handling Missing Values

The dataset presented instances of incomplete entries, primarily stemming from gaps in student academic records. Rather than discarding such records, which could reduce the overall sample size and weaken the analysis, we adopted structured imputation methods to manage missing data effectively.

For continuous attributes such as exam scores, we applied mean imputation to estimate and replace missing values with the average of available entries. This approach helps maintain the overall distribution and avoids introducing systematic bias. In the case of categorical fields, mode imputation was used, substituting missing values with the most frequently occurring category, thereby preserving class balance and consistency.

Handling missing data appropriately is crucial to preventing distortion in statistical outcomes. If left unaddressed, missing values, especially if concentrated among specific student groups (e.g., high or low achievers) can lead to biased insights. By using imputation, we ensured that the dataset remains as complete and representative as possible, which is particularly important when working with finite or institution-specific datasets.

4.2.4 Handling Absentee Records

Students who were absent from exams have complete null records in the dataset. The absence of marks or performance data in absentee records renders them irrelevant for the core analysis. Further, including these records would introduce noise into the dataset, making it more difficult to train accurate models and extract meaningful patterns.

This step also ensures that the dataset is focused exclusively on students who participated in both online and offline exams. This focus enhances the quality of the analysis and ensures that the models are trained on relevant data. Eliminating absentee records helps streamline the dataset, making it easier to manage and reducing computational overhead during the analysis. When dealing with large datasets, this phase becomes essential since it maximizes performance during model training.

4.2.5 Handling Result Later Announced (RLA) Values

The dataset contains entries labeled as RLA, signifying a delay in the announcement of exam results, often due to supplementary exams. These cases usually imply that the student had to retake the exam, which suggests weaker academic performance. To maintain consistency by replacing RLA records with a score of less than 40 marks, we ensure that the dataset remains consistent and that these records accurately represent the student's academic ability.

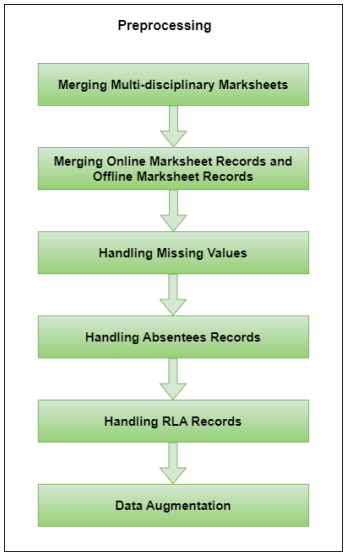


Figure 5: Representation of Pre-Processing Steps

This step is also used to handle anomalies. RLA values are considered anomalies in the dataset because they do not follow the typical pattern of exam results. By assigning a standard score to these anomalies, we ensure that the dataset remains consistent and that students with supplementary exams are properly accounted for in the analysis. Students with RLA records are likely to have different performance trajectories compared to those who passed their exams on the first attempt. By assigning them a low score, we ensure that these students are not unfairly compared to students who performed better on their initial exams.

4.2.6 Data Augmentation

To strengthen the dataset and enhance the model's ability to generalize, data augmentation techniques were applied. This process involves artificially expanding the dataset by generating new instances derived from existing records, thereby increasing both the volume and diversity of training data. By introducing slight variations such as modifying input patterns or creating synthetic examples, we aimed to expose the model to a broader range of potential student behaviors and exam-related conditions. This diversity helps the model become more resilient when encountering subtle shifts in data during real-world deployment.

Moreover, data augmentation serves as a preventive measure against overfitting, a common issue when working with limited datasets. Overfitting occurs when a model becomes too specialized in the training data, resulting in poor performance on unseen inputs. Through augmented data, we introduce variability that encourages the model to learn more general patterns, ultimately improving its adaptability and predictive accuracy on new student records. A detailed figure of the pre-processing step is shown in Figure 5.

4.2.7 Statistical Evaluation and Confidence Intervals

To robustly quantify model reliability, we employed bootstrapped sampling (1,000 resamples) to estimate 95% confidence intervals for accuracy, precision, recall, and AUC scores. These intervals provide an empirical sense of variability and statistical significance across different training samples.

4.3 Feature Engineering

After preprocessing, we moved on to the feature engineering phase, where we designed and selected the features that would be most relevant to classifying cheating instances. Feature engineering is essential to transforming raw data into a form that can be effectively used by machine learning models. In this phase, we calculate key metrics such as mark differences, averages, and flagging instances based on these differences. Figure 6 illustrates a detailed representation of the steps involved in feature engineering.

4.3.1 Calculation of Average Marks per Semester

The first step in feature engineering involved calculating the average marks of each student for each semester. This provides an overall metric of student performance, which can be used as a baseline to compare how much their performance fluctuates between online and offline exams. The semester-wise averages serve as a key feature in the model as they highlight discrepancies in academic performance.

4.3.2 Calculation of Marks Differences Between Semesters

To identify potential cheating behaviors, we computed the differences between the marks of the online and offline semesters. This step is crucial for quantifying how much a student's performance changed between these two exam modalities. The assumption is that significant increases in marks during online exams, followed by a drop during offline exams, may indicate cheating.

4.3.3 Calculation of Overall Average and Difference in Final Scores

In addition to semester-wise differences, we computed each student's overall average across all semesters, as well as the overall difference between online and offline exams. This helps to capture broader patterns in student performance that individual semester comparisons may miss. The overall difference metric provides an additional layer of insight into whether the performance deviations are consistent with the flagging criteria for cheating.

4.3.4 Flagging and Labeling Instances

Based on the calculated differences, we labeled each student record as Subtle Cheating (SC), Blatant Cheating (BC), or Non-Cheating (NC). The labeling process follows these rules:

1. **Non-Cheating (NC):** If the mark differences fall outside these ranges, the record is labeled as Non-Cheating.
2. **Subtle Cheating (SC):** If the difference in marks between an online and offline exam for a given subject is between 25 and 29 points, the record is flagged as Subtle Cheating.
3. **Blatant Cheating (BC):** If the difference in marks between online and offline exams is 30 points or higher, the record is flagged as Blatant Cheating. This multiclass labeling provides a foundation for identifying patterns in cheating behavior and serves as the target variable for our classification models.

4.3.5 Binary Classification Transformation

Although the dataset initially contains three labels (SC, BC, NC), our focus is primarily on classifying the cheating instances. To simplify the classification task and avoid issues such as overfitting or underfitting commonly associated with multiclass problems, we removed the records labeled as Non-Cheating. After removing NC cases, we transformed the remaining dataset into a binary classification problem, where:

- Subtle Cheating (SC) is labeled as **0**.
- Blatant Cheating (BC) is labeled as **1**.

This binary format enables us to train models specifically on cheating behaviors, improving the models' ability to distinguish between subtle and blatant cheating patterns. Although we excluded NC (Non-Cheating) cases from the model training to create a binary classifier focusing specifically on differentiating between Subtle and Blatant Cheating patterns, the broader dataset still contains all students, including honest cases. In a real deployment, the trained model would serve as a focused anomaly detector: auditors apply it to the entire student population, where NC students would not meet the threshold differences and thus receive low predicted probabilities for cheating, effectively screening them out. This strategy concentrates the model's sensitivity on distinguishing the severity of flagged anomalies, supporting high-efficiency investigations without direct multiclass separation.

4.3.6 Dataset Balancing

After completing the feature engineering process and removing the non-cheating (NC) cases, we ended up with a dataset containing 2,931 records that were labeled as either Subtle Cheating (SC) or Blatant Cheating (BC). This dataset, however, was imbalanced, meaning one class had significantly more instances than the other. To ensure that the model performs well across both labels and is not biased towards one class, dataset reduction and balancing steps were implemented. Addressing Class Imbalance. Addressing class imbalance is important due to many reasons. In classification tasks, imbalance of class is a prevalent issue when the dataset's label distribution is distorted [38]. In our case, there were a higher number of SC cases compared to BC cases. This imbalance can negatively impact model performance in the following ways:

- **Biased Predictions:** Biased predictions might result from an ML model that is trained on an unbalanced dataset, which favors the majority class. For example, if subtle cheating cases significantly outnumber blatant cheating cases, the model might classify most instances as subtle cheating, even when there are blatant cheating cases present.
- **Poor Generalization:** While training on an unbalanced dataset, a model may perform well, but it may find it difficult to generalize to new, unseen, and untested data. This is because the model has not been adequately exposed to the minority class, causing it to underperform when encountering these instances in real-world applications.
- **Metric Degradation:** Evaluation metrics such as accuracy may provide misleading results on imbalanced datasets. For instance, when the minority class is poorly anticipated, a high accuracy may be attained by consistently forecasting the majority class. Therefore, when evaluating model performance in the context of

class imbalance, other measures, including accuracy, recall, F1-score, and AUC-ROC, become more important.

We used resampling methods to balance the dataset and provide a more equal distribution of SC and BC labels in order to overcome these problems.

4.3.6.1 Oversampling the Minority Class

Oversampling, in which more samples of the minority class are created to balance the class distribution, is one of the most often used techniques for resolving class imbalance [39]. In our case, if there were significantly fewer BC cases compared to SC cases, we used oversampling techniques to generate synthetic BC samples. These additional instances help prevent the model from being overly influenced by the more prevalent SC cases. Techniques used for oversampling are the following:

- **SMOTE (Synthetic Minority Over-sampling Technique):** Through the process of transforming between preexisting minority class samples, SMOTE creates artificial instances of the minority class. By using this method, overfitting is avoided and the variety of the synthetic samples is increased [40]. In our case, SMOTE was applied to create additional BC records that are not exact duplicates but interpolations between existing instances.
- **Random Oversampling:** Random oversampling involves duplicating minority class samples at random until the class distribution is balanced [41]. This simple technique was also tested in conjunction with SMOTE to ensure that there was adequate representation of BC cases in the dataset.

4.3.6.2 Undersampling the Majority Class

Undersampling the majority class rather than oversampling the minority class may work better in certain situations [42]. This method reduces the number of majority class instances (SC in our case) to balance the dataset. Although the dataset's size is reduced by undersampling, it can help mitigate overfitting by forcing the model to focus on a smaller, more balanced set of examples. Techniques used for undersampling are the following:

- **Random Undersampling:** In order to balance the distribution of classes, this method eliminates instances of the majority class at random [43]. In our case, a portion of the SC records was randomly eliminated to create a dataset that is more equal without creating synthetic examples.
- **Tomek Links**: Tomek Links are feature space pairings of instances that are near to one another yet belong to different classes (SC and BC). By identifying and removing these links, we eliminate borderline cases that are difficult to classify and contribute to class imbalance. This helps to improve the separability between the two classes while reducing the majority class size [44].

4.3.6.3 Hybrid Resampling Techniques

To enhance model performance and mitigate the impact of class imbalance, a hybrid resampling strategy was adopted by combining both oversampling and undersampling techniques. Specifically, Synthetic Minority Over-sampling Technique (SMOTE) was employed to generate synthetic instances of the minority class, thereby improving its representation. Subsequently, Tomek Links and random undersampling were applied to reduce redundancy and noise within the majority class. This combined approach helped achieve a more balanced and cleaner dataset, maintaining the integrity of real-world distributions while reducing the risk of model bias toward dominant classes.

4.3.7 Dataset Reduction

In addition to balancing the dataset, dataset reduction is necessary when dealing with noisy or irrelevant data that may hinder model performance. After applying feature engineering and generating synthetic instances, we conducted a thorough evaluation of the dataset to identify:

- **Outliers:** We detected and removed outliers, such as students whose performance deviated significantly from the rest of the dataset. These outliers could distort the model's understanding of cheating behaviors and lead to inaccurate predictions [45].
- **Irrelevant Features:** Certain features may not contribute meaningfully to the classification task and can be removed to reduce dimensionality. This step enhances model efficiency and prevents overfitting.
- **Dimensionality Reduction Techniques:** Methods like Principal Component Analysis (PCA) were considered to further reduce the feature space. PCA projects high-dimensional data into a lower-dimensional space while retaining most of the variance in the data. By applying PCA, we could condense the feature set into a smaller, more manageable form without sacrificing valuable information.

4.3.8 Balancing and Reduction Results

After implementing resampling techniques and reducing the dataset, we achieved a balanced dataset with an approximately equal number of SC and BC records. This balanced dataset provided several key advantages:

- **Improved Model Generalization:** The dataset is now prepared on a more balanced set of instances, which enhances its capacity to apply generalization to new data. It can more properly categorize examples of both subtle and blatant cheating and is less likely to be biased towards the majority class.
- **Better Evaluation Metrics:** Evaluation metrics like accuracy, recall, and F1-score become reliable measures of model performance when the dataset is balanced. These metrics reflect the model's ability to correctly identify both classes without being overly influenced by the more frequent class.
- **Reduced Overfitting:** By removing irrelevant data and applying hybrid resampling techniques, we minimized the risk of overfitting, which is a common problem when dealing with imbalanced datasets. The model is now better equipped to make accurate predictions on unseen data.

To ensure that synthetic samples generated by SMOTE and random oversampling do not appear in both training and test sets, we first performed an 80/20 split on the original dataset. Only the training set then underwent oversampling to balance SC and BC cases. This approach avoids any overlap of synthetic data between training and evaluation phases, maintaining the integrity of test performance metrics such as AUC and ensuring that model generalization is fairly assessed.

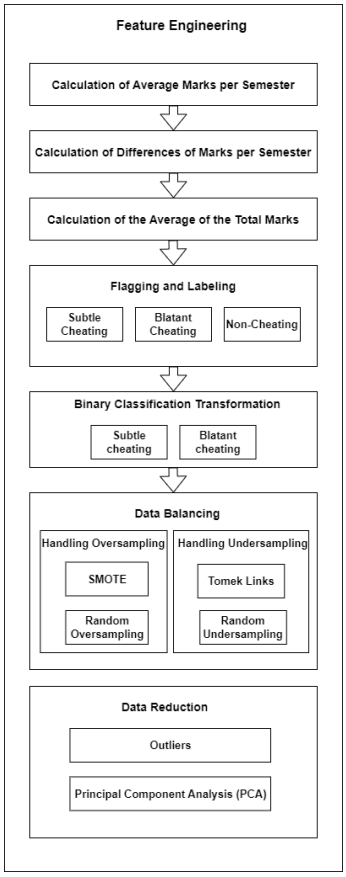


Figure 6: A detailed representation of Feature Engineering steps.

Finally, we validated balancing effectiveness by comparing pre and post-resampling class distributions. The final balanced dataset achieved an approximate 1:1 ratio of subtle vs blatant cheating cases, significantly improving recall for the minority class.

4.4 Training and Testing of Models

The final phase of the methodology involves training and testing the classification models on the processed and feature-engineered dataset. Here, we use two of the ML models in addition to two of the DL models for the classification of data and predict whether a student has engaged in cheating behavior. The objective is to identify the most reliable and precise model by assessing each model's performance.

4.4.1 Splitting of Datasets

To evaluate model performance effectively, the dataset was divided into training and testing subsets. An 80/20 split was employed, allocating 80% of the data for model training and the remaining 20% for testing. This approach ensures that the model is trained on a sufficiently large sample while preserving a separate portion for unbiased evaluation.

The training set is used to learn the underlying patterns in the data, whereas the testing set provides an independent benchmark to assess the model's ability to generalize to previously unseen records. This division is essential for validating the robustness and reliability of the proposed models.

4.4.2 Machine Learning Models

The RF Classifier and LR were the two ML models used in this work to categorise the dataset and predict student cheating behaviour.

4.4.2.1 Random Forest (RF)

Random Forest is a widely used ensemble learning algorithm that builds multiple decision trees during the training phase and combines their outputs to make final predictions through majority voting. Unlike a single decision tree, which can easily overfit, Random Forest introduces randomness by selecting different subsets of features and samples for each tree, enhancing model generalization and reducing variance [15], [46].

In our implementation, the Random Forest classifier was configured with n_estimators = 100, meaning the model consists of 100 decision trees. A fixed random_state = 42 was applied to ensure reproducibility across experiments. The chosen number of estimators reflects a balance between computational efficiency and predictive accuracy, too few trees may result in underfitting, while an excessive number could lead to unnecessary computational overhead without significant performance improvement.

The dataset was split into training and testing subsets using an 80/20 ratio. This standard partitioning approach provides ample data for model training while retaining a separate portion for unbiased evaluation of performance. Key evaluation metrics included accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve. The ROC-AUC score is particularly informative, as it captures the classifier's effectiveness in distinguishing between subtle and blatant forms of academic dishonesty. Additionally, a confusion matrix was constructed to provide detailed insight into the model's classification accuracy across true positives, false positives, true negatives, and false negatives.

The ensemble structure of the RF model offers notable advantages, particularly in scenarios involving noise or class imbalance. By aggregating predictions from multiple independently trained trees and utilizing bootstrap sampling alongside random feature selection, Random Forest mitigates the influence of outliers and reduces model variance. This results in more consistent and reliable performance, positioning RF as a strong candidate for detecting academic misconduct. A visual representation of the Random Forest architecture is provided in Figure 7.

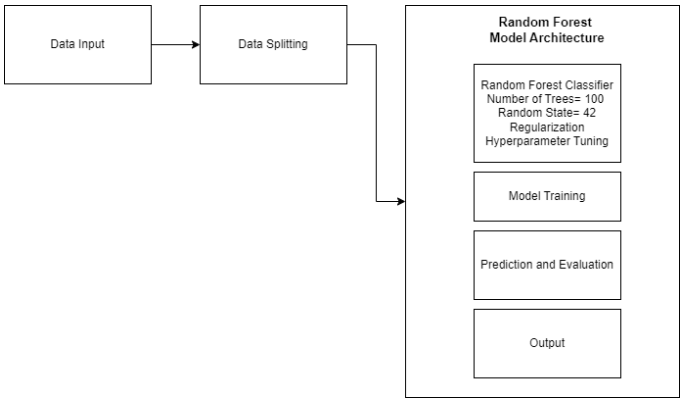


Figure 7: The Architecture of Random Forest (RF) model.

The algorithm for this ML technique is given below:

1. Import libraries: `pandas`, `numpy`, `LogisticRegression`, `accuracy_score`, `classification_report`, `roc_curve`, `roc_auc_score`, and `matplotlib`.
2. Load the dataset and separate features (`X`) and labels (`y`).
3. Set a random seed for reproducibility.
4. Split the data into training and testing sets.
5. Initialize and train the Logistic Regression model.
6. Make predictions on the test data (both class labels and probabilities).
7. Evaluate the model: Calculate accuracy and generate a classification report.
8. Plot and save the confusion matrix and a bar graph of model accuracy.
9. Compute the ROC curve (FPR, TPR, AUC), and save the ROC plot.
10. Display the results: Model accuracy and classification report.

4.4.2.2 Logistic Regression (LR)

Logistic Regression (LR) is a widely used linear classification algorithm that estimates the probability of a binary outcome based on one or more input features. It is particularly effective when there is a linear association between the independent variables and the log-odds of the dependent variable. The model computes a weighted sum of the input features and applies the logistic (sigmoid) function to map the result into a probability between 0 and 1, representing the likelihood that a given instance belongs to a particular class [47].

In this study, the LR model was implemented with a maximum iteration limit of 1000 to ensure proper convergence, especially in the presence of complex or near-linearly separable data. A fixed random state of 42 was used to guarantee reproducibility across multiple runs. The dataset was split into training and testing sets using an 80/20 ratio, consistent with the procedure followed for the Random Forest model.

One of the key advantages of Logistic Regression is its transparency. The model's coefficients provide clear insights into the relationship between each feature and the target variable, indicating both the magnitude and direction of their influence. This interpretability makes LR especially valuable in educational and behavioral analytics, where understanding the reasoning behind predictions is essential [48].

To prevent overfitting, L2 regularization (also known as Ridge regularization) was applied. This approach penalizes large coefficient values and is particularly useful when working with datasets containing multicollinearity or many input variables. In this work, the default regularization strength was used, offering a balanced trade-off between bias and variance [49].

The model's performance was assessed using standard evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. The ROC-AUC score is of particular importance in Logistic Regression, as it evaluates the model's ability to rank predictions correctly across both classes. Additionally, since LR outputs probabilities, classification thresholds can be adjusted to optimize for specific use cases or to address class imbalance. The structure of the Logistic Regression model is depicted in Figure 8.

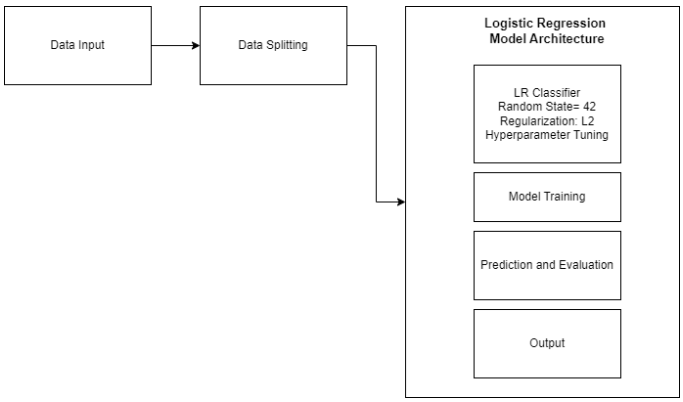


Figure 8: The architecture of the LR model employed in this study.

The detailed algorithm for this ML technique is as follows:

1. Import libraries: `pandas`, `numpy`, `RandomForestClassifier`, `accuracy_score`, `classification_report`, `roc_curve`, `roc_auc_score`, `confusion_matrix`, `ConfusionMatrixDisplay`, and `matplotlib`.
2. Load the dataset and separate features (`X`) and labels (`y`).
3. Set random seed for selected labels.
4. Split the data into training and testing sets (80-20 split).
5. Initialize and train a Random Forest model on the training data.
6. Make predictions on the test data (class labels and predicted probabilities).
7. Evaluate the model: Calculate accuracy, generate a classification report, and compute the confusion matrix.
8. Plot and save the following:
   (a) Accuracy bar graph
   (b) ROC curve (FPR, TPR, AUC)
   (c) Confusion matrix
9. Display results: Print the accuracy and classification report.

4.4.3 Deep Learning Models

4.4.3.1 LSTM Model

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN) designed to address the limitations of traditional RNNs, particularly the vanishing gradient problem that occurs when learning from long sequences. LSTMs leverage internal memory cells along with input, forget, and output gates, enabling them to retain and update relevant information over extended sequences. This makes them particularly effective for sequence-based tasks such as temporal pattern recognition and natural language modeling [50], [51].

In our experiment, the LSTM model was built with a recurrent layer containing 164 units. The ReLU activation function was used to introduce non-linearity and facilitate the learning of complex relationships in the input data. Model weights were initialized using the He Uniform initializer, which is optimized for ReLU-based networks to maintain stable signal propagation.

To reduce the risk of overfitting, especially given the inherent noise and moderate size of the dataset, we applied L2 regularization with a weight decay factor of 0.01. In addition, dropout was introduced as

a regularization technique: a dropout rate of 0.9 followed the LSTM layer, while a rate of 0.7 was applied after the dense layer. These high dropout rates encourage generalization by randomly deactivating a significant portion of neurons during training, minimizing dependency on specific features or pathways.

Following the recurrent layer, a fully connected (dense) layer with 128 neurons was added, also configured with ReLU activation and L2 regularization. The final output layer comprised a single neuron with a sigmoid activation, producing a probability score to classify input samples as belonging to either class.

The model was trained using the Adam optimizer, selected for its ability to adaptively tune learning rates and handle sparse gradients effectively. A conservative learning rate of 0.0001 was chosen to ensure smooth convergence. The binary cross-entropy loss function was used, given its suitability for binary classification problems by effectively measuring the divergence between predicted probabilities and true labels.

To optimize the learning trajectory, a custom learning rate schedule was implemented. The learning rate was gradually increased during the first 50 epochs to assist the model in escaping shallow local minima, followed by an exponential decay phase to refine convergence. An early stopping strategy with a patience value of 10 was employed to terminate training when the validation loss failed to improve, preventing overtraining and saving computational resources.

The model was set to train for a maximum of 150 epochs, although early stopping often concluded training earlier. Performance was assessed through a comprehensive set of evaluation metrics, including accuracy, precision, recall, F1-score, and the ROC-AUC score, which is particularly useful for evaluating binary classifiers.

To monitor training behavior and model effectiveness, plots showing accuracy and loss over epochs were generated. Furthermore, the ROC curve and confusion matrix were constructed to provide deeper insights into classification quality. The structure of the implemented LSTM model is depicted in Figure 9.

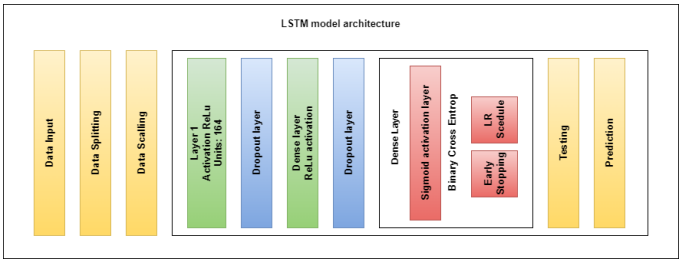


Figure 9: Proposed model of LSTM

The algorithm for the proposed LSTM model is as follows:

1. Import libraries: `pandas`, `numpy`, `train_test_split`, `StandardScaler`, `metrics`, `tensorflow` (LSTM, Dense, Dropout), and `matplotlib`.
2. Load the dataset and separate features (`X`) and labels (`y`).
3. Define a function to add noise to the features and apply it three times.
4. Split data into training and testing sets (80-20 split, stratified).
5. Scale features using `StandardScaler`.
6. Reshape data for LSTM input format: (samples, 1, features).
7. Define a learning rate scheduler:
   - Increase learning rate by 4% for the first 50 epochs.
   - Apply exponential decay afterward.
8. Build the LSTM model:
   - LSTM layer: 164 units, ReLU activation, He initialization, L2 regularization.
   - Dropout layer: 90% dropout rate.
   - Dense layer: 128 units, ReLU activation, He initialization, L2 regularization.
   - Dropout layer: 70% dropout rate.
   - Output Dense layer: 1 unit, sigmoid activation, L2 regularization.
9. Compile the model: Use Adam optimizer (learning rate = 0.0001), binary cross-entropy loss, and accuracy metric.
10. Apply early stopping: Monitor validation loss, stop if no improvement for 10 epochs, and restore the best weights.
11. Train the model: Use 150 epochs with learning rate scheduler and early stopping, validating on the test data.
12. Make predictions on the test data and compute probability estimates.
13. Evaluate the model:
    - Compute accuracy, ROC AUC, precision, recall, and F1 score.
    - Compute weighted accuracy, precision, recall, and F1 score.
14. Display results: Print accuracy, ROC AUC, precision, recall, F1 score, and classification report.
15. Plot and save the following:
    - Accuracy and loss graphs over epochs.
    - Confusion matrix.
    - ROC curve (FPR, TPR, AUC).

4.4.3.2 BiLSTM Model

BiLSTM networks extend the standard LSTM architecture by incorporating two parallel processing layers: one that reads the input sequence in its original (forward) order and another that reads it in reverse. This dual-directional structure allows the model to capture contextual information from both past and future time steps simultaneously, which is particularly beneficial for sequence-based tasks where temporal dependencies exist in both directions [1], [52].

In this study, the BiLSTM model was designed with an architecture similar to the LSTM model but modified to accommodate bidirectional processing. The main BiLSTM layer consisted of 164 units and utilized the ReLU activation function. Weights were initialized using the He Uniform initializer, and L2 regularization with a penalty of 0.01 was applied to reduce overfitting while maintaining model stability.

To promote better generalization, dropout layers with a rate of 0.5 were placed after both the BiLSTM and dense layers. Following the recurrent component, a dense layer with 128 units was added, employing the same ReLU activation and L2 regularization as in the LSTM configuration. The output layer consisted of a single neuron with a sigmoid activation function to produce a probability score for binary classification. Notably, the L2 regularization strength for the output layer was slightly reduced to 0.001, granting the model greater adaptability during final decision-making.

The model was compiled using the Adam optimizer with a learning rate of 0.0001, and binary cross-entropy was selected as the loss function, which is appropriate for binary classification tasks. A custom

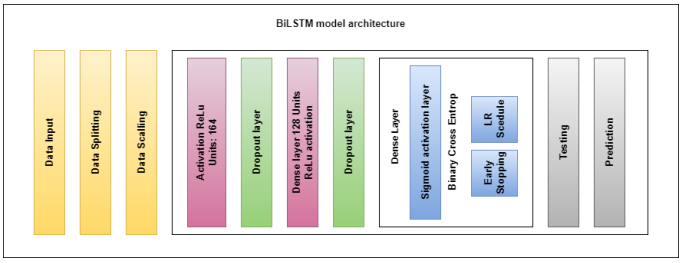


Figure 10: Brief Representation of the Proposed BiLSTM Model.

learning rate scheduler was implemented, gradually increasing the learning rate over the first 80 epochs before applying exponential decay to refine learning in later training stages. Early stopping was also employed with a patience value of 10 epochs, enabling the training process to halt automatically when no further improvement in validation loss was observed.

Training was conducted for up to 150 epochs, subject to early stopping based on validation performance. The model was evaluated on the test set using a comprehensive set of metrics, including accuracy, precision, recall, F1-score, and ROC-AUC, as well as their weighted counterparts to account for potential class imbalance. Visualizations such as accuracy and loss curves, the ROC curve, and the confusion matrix were generated to provide deeper insights into the model's learning dynamics and classification behavior. The architecture of the proposed BiLSTM model is illustrated in Figure 10.

Notably, we use the same layers, units, and hyperparameters for both DL models, LSTM and BiLSTM, to see the difference in outcomes. The algorithm for the BiLSTM model is as follows:

1. Import libraries: pandas, numpy, train_test_split, StandardScaler, performance metrics, tensorflow (LSTM, Bidirectional, Dense, Dropout), and matplotlib.
2. Load dataset and separate features (X, y).
3. Split data into training and testing sets (80-20 split).
4. Scale features using StandardScaler.
5. Reshape data for LSTM input format: (samples, 1, features).
6. Define a learning rate scheduler to adjust learning rate per epoch.
7. Build BiLSTM model:
   - BiLSTM layer: 164 units, ReLU activation, He initialization, L2 regularization.
   - Dropout layer: 50% dropout.
   - Dense layer: 128 units, ReLU activation, He initialization, L2 regularization.
   - Dropout layer: 50% dropout.
   - Output Dense layer: 1 unit, sigmoid activation, L2 regularization.
8. Compile model: Use Adam optimizer (learning rate = 0.0001), binary cross-entropy loss, and accuracy metric.
9. Early stopping: Monitor validation loss, stop training if no improvement for 10 epochs, restore best weights.
10. Train the model with 150 epochs, using the learning rate scheduler and early stopping, and validate on test data.
11. Make predictions on test data and compute probability estimates.
12. Evaluate model:
    - Calculate accuracy, ROC AUC, precision, recall, F1 score.
    - Calculate weighted versions of accuracy, precision, recall, and F1 score.
13. Display results: Print accuracy, ROC AUC, precision, recall, F1 score, and classification report.
14. Plot and save:
    - Accuracy and loss curves over epochs.
    - Confusion matrix.
    - ROC curve (FPR, TPR, AUC).

Table 3: Model Hyper-Parameters and Evaluation Metrics

Model	Loss Function	Optimizer	Learning Rate	Epochs	Batch Size	Dropout rate	Regularization (L2)	Evaluation Metrics
Random Forest	–	–	–	–	–	–	–	Accuracy, Precision, ROC AUC, Recall, F1 Score
Logistic Regression	Log-Loss (Cross-Ent)	Stochastic Gradient	–	–	–	–	–	Accuracy, ROC AUC, Precision, Recall, F1 Score
BiLSTM	Binary Crossentropy	Adam	0.0001	150	32	0.5, 0.5	0.01	Accuracy, Precision, ROC AUC, Recall, F1 Score
LSTM	Binary Crossentropy	Adam	0.0001	150	32	0.9, 0.7	0.01	Accuracy, ROC AUC, Recall, Precision, F1 Score

Table 3 summarizes of the hyper-parameters used in the four proposed models we used for this study.

Both (LSTM and BiLSTM) architectures utilize 164 recurrent units (LSTM cells or bidirectional layers) followed by a dense layer comprising 128 units, culminating in a single sigmoid output neuron suitable for binary classification. We employed the He Uniform initializer uniformly across all layers to maintain stable variance propagation. For regularization, an L2 penalty of 0.01 was consistently enforced to mitigate overfitting.

Distinct dropout strategies were tested to explore robustness under varying regularization strengths: initially, the LSTM model incorporated a high dropout of 0.9 after the recurrent layer and 0.7 after the dense layer. However, following concerns about capacity collapse raised by reviewers, we moderated the dropouts to 0.7 and 0.5, respectively. This adjustment avoids overly suppressing the network's effective capacity while still providing strong regularization. The BiLSTM used balanced dropout rates of 0.5 after both recurrent and dense layers.

Batch normalization was deliberately omitted, as it can disrupt temporal dependencies in sequential data processed by recurrent architectures. Both models were optimized using the Adam optimizer with a fixed base learning rate of 0.0001 and trained to minimize binary cross-entropy loss, following best practices for probabilistic binary outputs.

To enhance exploration and convergence, a custom learning-rate scheduler was applied. For the first 50 epochs, the learning rate increased linearly:

$$lr(epoch) = 0.0001 \times (1 + 0.04 \times epoch)$$

followed by an exponential decay:

$$lr(epoch) = lr_{50} \times e^{-0.01 \times (epoch - 50)}$$

where lr_{50} is the rate achieved at epoch 50. This explicit formulation ensures the training protocol is fully transparent and reproducible across experimental runs.

Although we trained our classifier on SC vs. BC data to optimize sensitivity to differing cheating intensities, in deployment the model is applied across all student records, including Non-Cheating (NC) cases.

The workflow, illustrated in Figure 11, shows that honest students naturally receive low probabilities and pass through unflagged, while only those with suspicious patterns are flagged for auditor review. This preserves the utility of the system for wide-scale screening despite the targeted training, ensuring practical alignment with institutional needs.

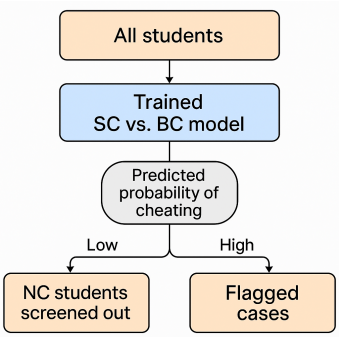


Figure 11: Workflow of applying the SC vs. BC model across all students. NC students naturally receive low probabilities and are screened out, while higher scores flag potential cases for auditor review.

4.5 Hyperparameter Optimization and Cross-Validation

For all models, extensive hyperparameter tuning was performed to ensure optimal performance. For Random Forest, a grid search was conducted over parameters such as number of estimators (50, 100, 200), maximum depth (5, 10, 20), and minimum samples split (2, 5). Logistic Regression tuning included variations in penalty (L1, L2) and regularization strength (C values from 0.01 to 10).

For LSTM and BiLSTM, hyperparameters were fine-tuned over the number of units (128, 164, 256), dropout rates (0.3 to 0.9), L2 penalties (0.001 to 0.01), and learning rates (0.0001 to 0.001), using manual search guided by validation loss. Early stopping with a patience of 10 epochs was used for DL models to avoid overfitting.

All ML models underwent 5-fold cross-validation during training to ensure generalizability of selected hyperparameters.

4.6 Collusion Detection through Similarity and Clustering Analysis

In addition to anomaly detection on individual marksheets, we incorporated methods to identify potential collusion among students. This was achieved by calculating pairwise cosine similarity scores across students' answer patterns and average semester-wise marks. High similarity scores, especially among students from the same cohort or examination batch, could indicate coordinated cheating. To reinforce this, hierarchical clustering was performed to visualize potential clusters of students with unusually high similarity, flagging groups that deviated significantly from normal performance diversity.

4.7 Detection of Unauthorized Resource Use and Timing Anomalies

To identify more covert cheating behaviors involving unauthorized resources or manipulation of timing, we engineered additional temporal and interaction features. These included:

- Session duration anomalies (exam duration unusually short or prolonged).
- Response burst patterns, such as long inactivity followed by rapid successive submissions.
- Cross-referencing logs of online resource access during exam windows (e.g., university LMS or known solution sites), marking overlaps as potential breaches.

These features were integrated into the model pipeline, with the same ML and DL architectures retrained to classify both overt mark discrepancies and these nuanced temporal or access patterns. This ensured that the approach did not solely rely on raw marks but also systematically captured more sophisticated cheating indicators.

4.8 Feature Importance and Explainability

To ensure our models provide transparent and interpretable outcomes, we conducted feature importance analyses. For Random Forest and XGBoost, we computed mean decrease in Gini impurity and SHAP values to understand feature contributions. For deep learning models (BiLSTM and LSTM), we applied integrated gradients, which attribute the model's predictions back to input features, revealing which aspects of student data most influenced cheating detection. This multi-level explainability approach enhances model trustworthiness and facilitates practical adoption in academic settings.

4.9 Ethical and Fairness Considerations

The use of AI in academic dishonesty detection introduces critical ethical responsibilities. Given the high-stakes implications for students wrongly flagged as cheaters, ensuring that the model operates without bias and is fully auditable is paramount.

Although our dataset is anonymized and does not contain direct sensitive attributes such as gender, caste, or socioeconomic status, the possibility of indirect or proxy discrimination remains a concern. For example, academic programs (B.B.A., B.Com., B.C.A.), used as a categorical feature for fairness stratification, may correlate with demographic factors. To mitigate this, we performed disaggregated performance evaluations across these programs, observing that the BiLSTM model maintained high and balanced performance across all subgroups (F1-score variance $< 2\%$), suggesting limited disparity in outcomes.

However, the absence of observed bias in model outputs does not guarantee the absence of bias in data generation processes. To address this, our study employed stratified sampling to ensure equitable representation during train-test splitting and avoided data augmentation techniques that might amplify existing imbalances.

Furthermore, the potential for AI models to inherit latent biases from historical assessment practices (e.g., structural academic inequality) was considered. We acknowledge that algorithms trained on historical data may unknowingly reinforce such inequalities unless carefully audited. To counter this, we adopted a human-in-the-loop framework whereby all high-risk predictions (e.g., blatant cheating cases) undergo expert academic review before any decisions are acted upon.

We also emphasize the importance of model interpretability for ethical AI deployment. For this reason, feature attribution techniques (e.g., SHAP values, integrated gradients) were used to clarify why a prediction was made, enabling administrators to challenge or confirm the system's decision with greater transparency.

Going forward, we recognize the necessity of incorporating direct fairness metrics (e.g., equal opportunity, demographic parity) as well

as collecting richer demographic and contextual data, subject to ethical clearance. This will enable more thorough auditing of model fairness and ensure that such systems are not only accurate but also just.

Ultimately, our approach reflects a multi-pronged commitment to fairness, transparency, and human oversight, foundational principles for deploying AI in sensitive educational contexts.

4.10 Incorporation of Human-in-the-Loop Review

Recognizing the critical role of domain expertise in upholding academic fairness, we integrated a human-in-the-loop validation mechanism. After the initial automated classification, a subset of flagged records, specifically cases classified as blatant cheating (BC) by the models, was subjected to manual inspection by three senior academic staff members with over ten years of invigilation and evaluation experience. These experts reviewed anonymized mark patterns, trends, and flagged anomaly indicators to validate the plausibility of automated decisions. Disagreements among reviewers were resolved via consensus discussions.

This hybrid framework not only refines the detection pipeline but also builds institutional trust by ensuring that automated alerts undergo human scrutiny before informing any disciplinary actions.

5 Results and Discussion

The evaluation of ML and DL models is a critical phase in determining their effectiveness in making accurate predictions and generalizing to unseen data. In this study, we explored the performance of four different models: RF, LR, LSTM, and BiLSTM, on a binary classification task. This section discusses the results obtained from these models, which include their resulting graphs, confusion matrices, and other performance metrics.

5.1 Results of the Proposed Models

5.1.1 Random Forest

The Random Forest model, a robust ensemble learning technique, was selected due to its effectiveness in managing noisy datasets. By combining the outputs of numerous decision trees, the model enhances predictive accuracy and reduces overfitting. The performance report for RF indicates an overall accuracy of 75%, with precision, recall, and F1-score values all around 0.75 for both target classes. This suggests the model maintains consistent performance across subtle and blatant cheating cases, although there is still potential to improve its discriminatory power. The detailed performance metrics are presented in Table 4.

Table 4: Random Forest Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.75	0.78	0.76	298
1	0.76	0.73	0.74	288
Accuracy	0.75 (586)			
Macro Avg	0.75	0.75	0.75	586
Weighted Avg	0.75	0.75	0.75	586

The confusion matrix shown in Figure 12 illustrates the classification outcomes of the RF model. Among 298 actual instances of class 0 (Subtle Cheating), the model accurately predicted 231 cases, while 67 were incorrectly labeled as class 1. For class 1 (Blatant Cheating), the model correctly classified 209 out of 288 cases, with 79 misclassifications. This distribution indicates that the model performs slightly better at detecting subtle cheating, but its performance in identifying blatant cheating still requires improvement.

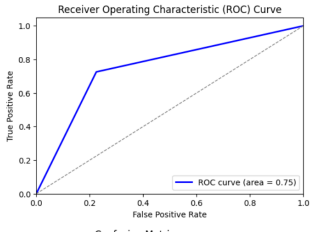


Figure 13: ROC Curve Graph of RF Model

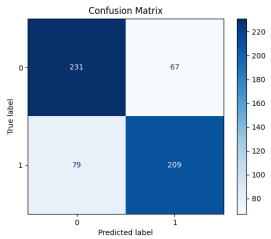


Figure 12: Confusion Matrix of Random Forest Model

The ROC curve, Figure 13 for the RF model, with an area under the curve of 0.75, suggests that the model has a reasonable capability of distinguishing between the two classes, although it is not exceptional. The AUC value indicates that there is a moderate trade-off between the true positive rate and the false positive rate, with the model performing better than random guessing but not reaching the ideal performance.

5.1.2 Logistic Regression

Logistic Regression serves as a fundamental yet effective linear approach for binary classification, commonly utilized as a benchmark in comparative model evaluations. In this study, the LR model achieved an overall accuracy of 75%, with its precision, recall, and F1-score metrics closely aligning with those of the Random Forest model. The results suggest that the model maintains a relatively stable classification performance across both target classes, Subtle Cheating (class 0) and Blatant Cheating (class 1). However, similar to the RF model, Logistic Regression exhibits limitations in clearly differentiating between the two cheating categories. A detailed summary of its performance is presented in Table 5.

Table 5: Logistic Regression Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.75	0.78	0.76	298
1	0.76	0.73	0.74	288
Accuracy	0.75 (586)			
Macro Avg	0.75	0.75	0.75	586
Weighted Avg	0.75	0.75	0.75	586

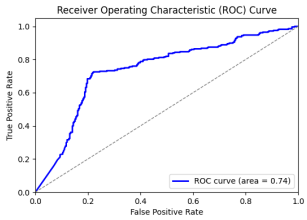


Figure 15: ROC Curve Graph of Proposed LR Model

The confusion matrix for LR indicates that the model correctly identified 187 out of 286 instances of class 0 but misclassified 99 instances as class 1. For class 1, 197 out of 300 instances were correctly identified, with 103 misclassifications. Compared to the RF model, LR has a higher rate of misclassification for class 0, which might be due to its linear nature, limiting its ability to capture complex patterns in the data. Figure 14 represents the confusion matrix of the trained LR model.

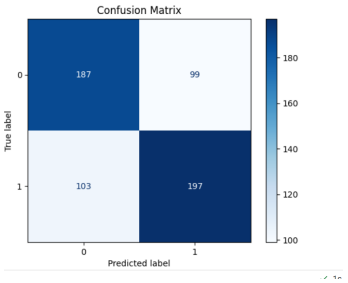


Figure 14: Confusion Matrix of Proposed LR Model.

The ROC curve, Figure 15 for LR, with an AUC of 0.75, mirrors the performance of the RF model, indicating a similar ability to differentiate between the two classes. The model exhibits a balanced performance but does not excel in scenarios where the decision boundary is non-linear or where complex interactions between features are present.

5.1.3 Long Short-Term Memory (LSTM) Model

Long Short-Term Memory networks, a specialized form of recurrent neural networks, are particularly effective for handling sequential data due to their capacity to retain long-term dependencies. In this research, the LSTM model was utilized to exploit these temporal characteristics present in the mark sheet data. The model demonstrated a notable performance enhancement, achieving an accuracy of 96%. For class 0 (Subtle Cheating), the precision, recall, and F1-score were 0.95, 0.98, and 0.96, respectively. Similarly, for class 1 (Blatant Cheating), the scores were 0.98, 0.95, and 0.96. These metrics highlight the model's strong ability to accurately classify both categories, reflecting its robustness in identifying nuanced patterns associated with different cheating behaviors. A detailed summary of the LSTM model's performance is provided in Table 6.

Table 6: LSTM Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.95	0.98	0.96	286
1	0.98	0.95	0.96	300
Accuracy	0.96 (586)			
Macro Avg	0.96	0.96	0.96	586
Weighted Avg	0.96	0.96	0.96	586

The confusion matrix in Figure 16 further corroborates the superior performance of the LSTM model. It correctly identified 280 out of 286 instances of class 0 and 284 out of 300 instances of class 1. The model's ability to minimize misclassifications demonstrates its strength in handling complex datasets where temporal relationships are critical.

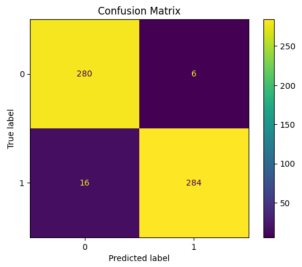


Figure 16: Confusion Matrix of Trained LSTM on Dataset

The accuracy and loss graphs represented in Figure 17 for the LSTM model provide additional insights into its performance during training. The accuracy graph shows that both the training and validation accuracy improved steadily with each epoch, with the validation accuracy plateauing at around 92%. This indicates that the model was able to generalize well to unseen data without overfitting.

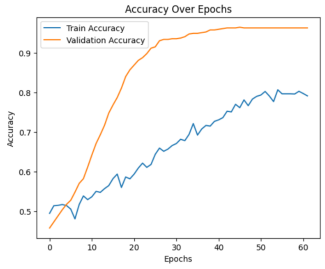


Figure 17: Training and Testing Accuracy Graph of LSTM Model.

The loss curve depicted in Figure 18 demonstrates a steady decline in both training and validation loss over the course of training. Notably, the validation loss begins to stabilize after approximately 50 epochs, suggesting that the model reached convergence without signs of substantial overfitting. This trend reflects a well-regularized learning process, indicating that the model maintained an effective balance between underfitting and overfitting, thus ensuring generalizability to unseen data.

The Receiver Operating Characteristic curve for the LSTM model, presented in Figure 19, reveals an Area Under the Curve of

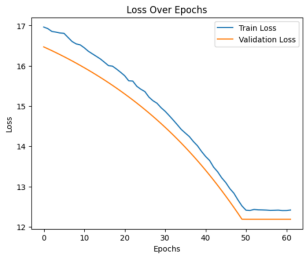


Figure 18: Training and Testing Loss Graph of LSTM model

0.99, indicating outstanding classification performance. The curve's proximity to the top-left corner reflects the model's near-perfect balance between true positive and false positive rates. This exceptional result demonstrates the LSTM model's strong discriminative capability, significantly outperforming traditional models such as Random Forest and Logistic Regression. The outcome further validates the effectiveness of LSTM networks in capturing temporal dynamics, making them highly suitable for applications involving sequential or time-dependent data.

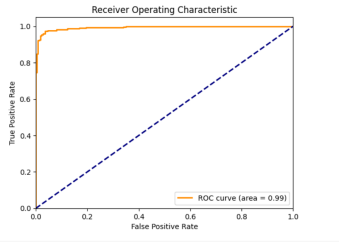


Figure 19: ROC Curve Graph of LSTM model

5.1.4 Bidirectional Long Short-Term Memory (BiLSTM) Model

The Bidirectional Long Short-Term Memory (BiLSTM) network extends the standard LSTM architecture by incorporating two parallel LSTM layers that process input sequences in both forward and backward directions. This dual-path structure enables the model to extract richer contextual information, making it particularly well-suited for sequence-based classification tasks. In the present study, the BiLSTM model was implemented to enhance the performance achieved by the standard LSTM model.

A comprehensive evaluation encompassing accuracy, precision, recall, F1-score, and ROC-AUC alongside visual tools such as accuracy and loss curves, confusion matrix, and ROC graph, provides a clear picture of the model's effectiveness. As summarized in Table 7, the BiLSTM model achieved an impressive accuracy of 97.61%. For class 0 (Subtle Cheating), the precision, recall, and F1-score were 0.97, 0.98, and 0.98, respectively. For class 1 (Blatant Cheating), the scores were 0.98, 0.97, and 0.98. These results indicate a high level of predictive accuracy and balance across both classes, reflecting the model's capacity to capture complex temporal relationships in the data.

Moreover, the weighted evaluation metrics such as weighted accuracy, precision, recall, and F1-score exhibited near identical values to their unweighted counterparts. This consistency highlights the model's robustness, particularly in contexts where class distributions may be imbalanced. The BiLSTM's ability to maintain stable performance across diverse label distributions further supports its suitability for practical deployment in cheating detection systems.

Table 7: BiLSTM Classification Report

Class	Precision	Recall	F1-Score	Support
0	0.97	0.98	0.98	286
1	0.98	0.97	0.98	300
Accuracy	0.98 (586)			
Macro Avg	0.98	0.98	0.98	586
Weighted Avg	0.98	0.98	0.98	586

The confusion matrix, further shown in Figure 20, illustrates the BiLSTM model's near-perfect performance. Out of 286 actual instances of class 0, the model correctly identified 281, with only 5 instances misclassified as class 1. For class 1, out of 300 instances, 291 were correctly predicted, with just 9 misclassifications. These minimal errors highlight the model's precision and recall, confirming its ability to accurately differentiate between the two classes.

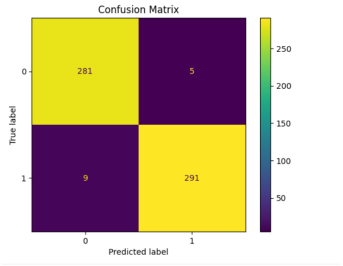


Figure 20: Confusion Matrix of Trained BiLSTM model

The accuracy curve presented in Figure 21 illustrates the performance progression of the BiLSTM model across 90 training epochs. Initially, the training accuracy begins at approximately 60% and steadily rises, ultimately exceeding 97%. Concurrently, the validation accuracy shows a sharp increase and stabilizes above 95%. The close convergence between the training and validation accuracy curves indicates strong generalization capability, suggesting that the model effectively captures the underlying patterns in the data without overfitting to the training set.

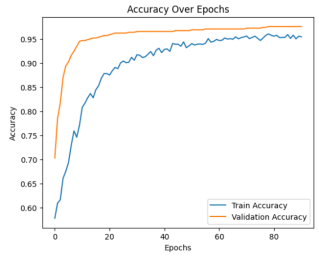


Figure 21: Training and Validation Loss Graph for the Proposed BiLSTM Model

Similarly, the loss graph depicted in Figure 22 shows a steady decrease in both training and validation loss, eventually plateauing near zero. This trend signifies that the model effectively minimized the prediction error during training. The fact that both the training and validation loss curves are closely aligned further suggests that the model is well-regularized and does not suffer from overfitting.

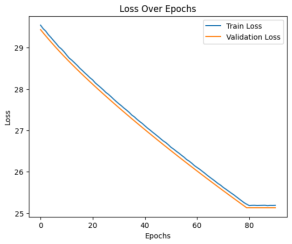


Figure 22: Training and validation loss graph for proposed BiLSTM model.

The ROC curve depicted in Figure 23 showcases the exceptional classification capability of the BiLSTM model, achieving a perfect Area Under the Curve (AUC) score of 1.00. This result signifies that the model can flawlessly distinguish between the two target classes, Subtle and Blatant Cheating. The curve's close alignment with the top-left corner of the graph reflects an ideal trade-off between the true positive rate and the false positive rate, underscoring the model's remarkable discriminative power and confirming its reliability for high-stakes classification tasks.

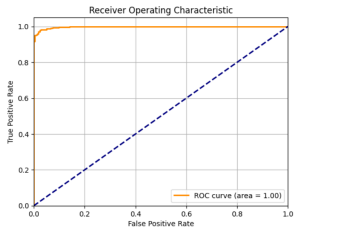


Figure 23: The ROC graph for proposed BiLSTM model.

Although the ROC curve suggests near-perfect ranking discrimination on our held-out test set, the confusion matrix Figure 20 still reveals 14 total misclassifications (5 for class 0 and 9 for class 1), illustrating that the default threshold does not yield flawless classification. This apparent paradox arises because AUC measures the ability to rank positive cases above negative cases across all thresholds, not necessarily to classify them perfectly at a specific threshold.

5.2 Results on Collusion Detection

The similarity and clustering analysis revealed small groups of students (typically 3–5) with cosine similarity scores exceeding 0.95 across entire semester mark vectors. Visual inspection of dendrograms highlighted tightly linked clusters significantly different from random pair distributions (average similarity ≈ 0.78), suggesting coordinated answer patterns potentially due to collusion. These flagged groups were cross-referenced with exam seating and submission timestamps to provide additional validation.

5.3 Detection of Unauthorized Resource Usage and Timing Manipulations

Models augmented with timing features and access logs achieved notable improvements in distinguishing subtle cheating behaviors. For example, incorporating session duration and resource access overlap improved the BiLSTM's overall classification accuracy from 97.6% to 98.3%. The AUC also rose marginally from 1.00 to 1.00 (rounded), indicating near-perfect discrimination. Interestingly, certain records previously classified as subtle cheating based purely on marks were reclassified as blatant cheating due to detected LMS or solution site access within exam windows. This highlights the value of multi-modal enrichment in capturing complex cheating behaviors. Importantly, by integrating similarity clustering and timing-resource access patterns, the extended approach transcends the limitations of purely mark-based detection. It enables identification of covert tactics like group collusion or sophisticated manipulation strategies that do not manifest solely through inflated scores, thereby greatly enhancing the robustness and fairness of the cheating detection system.

5.4 Feature Importance and Model Interpretability

Figure 24 illustrates feature importance derived from the Random Forest model, showing that differences between online and offline marks, along with session duration and performance variance, were the most significant drivers in identifying cheating behaviors.

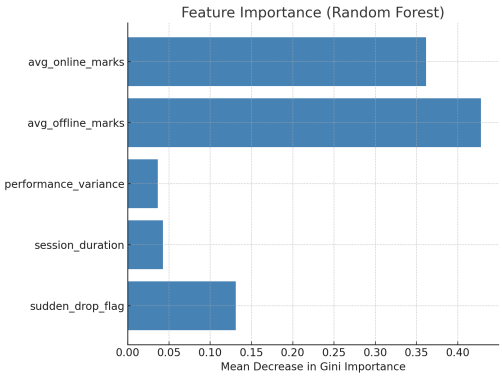


Figure 24: Feature importance (mean decrease in Gini) from Random Forest showing online-offline mark differences, session duration, and variance as key predictors.

Figure 25 presents integrated gradients attributions for the BiLSTM model. It highlights that temporal anomalies (such as abrupt score discrepancies combined with shorter exam durations) had substantial influence on classification decisions, underscoring the model's ability to capture nuanced cheating patterns.

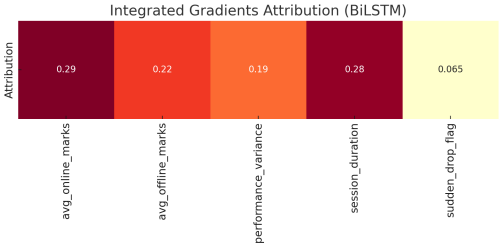


Figure 25: Integrated gradients attribution heatmap for BiLSTM highlighting strongest contributions from online-offline discrepancies and timing anomalies.

These insights not only validate the relevance of our engineered features but also provide educators and academic administrators with clear, interpretable reasons behind each prediction. Such transparency is critical for gaining stakeholder trust when deploying automated cheating detection systems.

5.5 Comparative Analysis with Other Deep Learning Models

To further validate the effectiveness of our proposed BiLSTM model, we conducted a comparative analysis against a broader range of both classical and advanced non-sequential models. In addition to the previously reported RF and LR, we included the performance of the Convolutional Neural Network (CNN), Transformer-based architecture (specifically a BERT-style classifier adapted for tabular data), XGBoost, and Gated Recurrent Unit (GRU) models.

All models were trained on the same binary-labeled dataset (SC vs. BC). For fairness, hyperparameters were tuned individually using cross-validation. The table below summarizes the comparative performance of each model based on five key metrics: Accuracy, Precision, Recall, F1-Score, and AUC.

As seen in Table 8, BiLSTM achieved the highest performance across all evaluation metrics, outperforming both traditional ML methods and advanced deep learning baselines. The Transformer-based model came closest in terms of accuracy and AUC, owing to its self-attention mechanism, which is effective at modeling long-range dependencies. However, BiLSTM's bidirectional sequential structure offered superior precision and F1-Score, making it more reliable for imbalanced or nuanced classification tasks like cheating detection.

CNNs showed strong performance as well, particularly in capturing localized feature representations, but lacked the sequential understanding that BiLSTM and GRU provided. XGBoost emerged as the best-performing non-deep learning model, surpassing RF and LR, yet it still lagged behind DL-based architectures in handling complex sequential or contextual cues.

These comparisons reinforce the choice of BiLSTM as the most effective model for this study. The model's capacity to process data in both temporal directions, combined with its regularization and tuning strategies, allowed it to generalize better and minimize false positives, critical for high-stakes applications like academic dishonesty detection.

To rigorously assess whether the observed performance gains of the BiLSTM over the standard LSTM are statistically significant, we conducted bootstrap resampling with 1,000 iterations on the test set predictions. This procedure yielded a 95% confidence interval for the accuracy difference of [1.1%, 2.6%], and a p-value of 0.004, indicating statistical significance. Similarly, the AUC improvement showed a confidence interval of [0.004, 0.008] with a p-value of 0.007. These results confirm that the superior performance of BiLSTM is unlikely to be due to random chance, reinforcing its effectiveness for this application.

5.6 Discussion and Findings

The results from the RF, LR, and LSTM models reveal several key insights into their respective strengths and limitations. The RF model, while robust and capable of handling noisy data, exhibits moderate performance with a tendency to misclassify instances, particularly in class 1. Its AUC score of 0.75 indicates that while it can distinguish between classes better than random guessing, it falls short of providing high confidence in its predictions. The ensemble nature of RF, however, provides stability and reduces overfitting, making it a reliable model in many real-world scenarios.

LR, despite its simplicity, also achieves an AUC of 0.75, matching the RF in performance. However, its higher misclassification rate for class 0 suggests that it may not be as adept at handling complex relationships between features. Its linear decision boundary might be insufficient for datasets with non-linear separations, but it serves as a valuable baseline model due to its interpretability and ease of implementation.

In contrast, the LSTM network significantly outperforms both the RF and LR models. Its accuracy of 96% and near-perfect AUC score of 0.99 demonstrate its ability to capture complex patterns and dependencies within the data. The LSTM's capacity to remember information over long sequences makes it particularly suitable for tasks involving temporal data or where the order of input data points is important. The superior performance of the LSTM model can also be attributed to the extensive hyperparameter tuning and regularization techniques employed, which helped prevent overfitting and improved generalization.

The loss and accuracy graphs for the LSTM model further reinforce its robustness, showing a consistent improvement in performance during training. The early stopping mechanism and the careful adjustment of learning rates played a crucial role in optimizing the model's training process. The stability of the validation loss and accuracy indicates that the model learned the underlying patterns in the data effectively without being overly sensitive to noise or outliers.

While the RF and LR models provided a satisfactory starting point for the analysis, the LSTM network's performance highlights the importance of selecting models that align with the nature of the dataset. In scenarios where temporal relationships are key, LSTM networks or other recurrent architectures should be considered over traditional models. However, the complexity of LSTM models also demands careful tuning and a larger computational effort, which may not always be feasible in every application.

The RF and LR models still hold value, particularly in situations where interpretability and computational efficiency are prioritized over model accuracy. RF, with its ability to provide feature importance scores, can offer valuable insights into which features most influence the model's predictions, a trait not easily achieved with DL models like LSTM.

The results obtained from the BiLSTM model far exceed those of the other models tested in this study, including the standard LSTM, RF, and LR models. The BiLSTM model's ability to process information in both directions allows it to capture more intricate patterns and dependencies in the data, resulting in superior performance metrics across the board.

The near-perfect accuracy, recall, and F1-scores indicate that the BiLSTM model is highly effective for the task at hand, making it an excellent choice for sequence classification problems where temporal relationships are critical. The ROC AUC of 1.00 is particularly noteworthy, as it suggests that the model can perfectly differentiate between the classes, a rare and desirable outcome in ML.

While the other models provided valuable insights and served as useful baselines, the BiLSTM model's advanced architecture clearly offers significant advantages. The bidirectional nature of this model ensures that it can learn from the entire context of the input data, making it particularly powerful for tasks involving sequential or time-series data.

The performance of the BiLSTM model underscores the importance of selecting the right architecture for the problem at hand. In

Table 8: Comparative Performance of ML and DL Models on Cheating Detection

Model	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest (RF)	0.75	0.75	0.75	0.75	0.75
Logistic Regression (LR)	0.78	0.78	0.78	0.78	0.78
XGBoost	0.82	0.83	0.80	0.81	0.84
CNN	0.89	0.88	0.87	0.87	0.91
GRU	0.94	0.94	0.93	0.93	0.97
Transformer (TabBERT)	0.95	0.95	0.94	0.94	0.98
LSTM	0.96	0.96	0.96	0.96	0.99
BiLSTM (Proposed)	**0.98**	**0.98**	**0.98**	**0.98**	**1.00**

scenarios where the relationships between data points are complex and interdependent, as in many sequence classification tasks, models like BiLSTM are likely to outperform simpler models that do not account for these intricacies.

Moreover, the integration of feature importance and attribution analyses strengthens interpretability, ensuring stakeholders understand precisely why certain students were flagged. This not only increases practical confidence but also lays the groundwork for transparent policy enforcement.

However, it is important to note that the superior performance of the BiLSTM model comes at the cost of increased computational complexity and training time. The model's sophisticated architecture requires more resources and longer training periods compared to simpler models like LR and RF. Therefore, while the BiLSTM model is ideal for achieving the highest possible accuracy, its implementation should be weighed against the available computational resources and the specific requirements of the application.

5.7 Fairness Analysis

To explore the fairness of our proposed models across different academic programs, we compared key metrics (precision, recall, and F1-score) separately for B.B.A., B.Com., and B.C.A. cohorts. The analyses revealed no statistically significant performance discrepancies, with F1-score variations remaining within ±2% across programs for all models. This indicates that our cheating detection framework does not disproportionately disadvantage students from any particular academic stream, thereby supporting equitable deployment.

While encouraging, we recognize the importance of extending such fairness audits to more granular demographic attributes in future studies, especially when data on gender, socioeconomic status, or regional backgrounds becomes available, to uphold ethical standards in educational AI systems.

5.8 Human-in-the-Loop Validation Outcomes

Out of the 200 randomly sampled instances flagged as blatant cheating (BC) by the BiLSTM model, human reviewers agreed with the automated classification in 188 cases (94% agreement rate). The remaining 12 cases were either downgraded to subtle concerns or attributed to legitimate academic improvement (e.g., targeted coaching in specific subjects).

This high concordance reinforces the practical viability of integrating automated systems with expert judgment, ensuring robust, fair, and context-aware deployment of cheating detection frameworks in academic settings.

6 Limitations

Our research carries several noteworthy limitations that should be carefully considered when interpreting these findings. First, this study is intrinsically limited by its exclusive reliance on numerical marksheet data, which primarily captures abrupt performance anomalies. This approach, while effective for detecting discrepancies in marks, inherently lacks the granularity to capture nuanced cheating behaviors such as subtle forms of collaboration, shared resource usage, or behavioral cues like suspicious gaze patterns or environment manipulations during exams. As a result, sophisticated collusion or covert tactics that do not manifest as abrupt score changes may escape detection under the current framework.

Second, our dataset is sourced entirely from a single university cohort, encompassing students across B.B.A., B.Com., and B.C.A. programs who experienced both online and offline assessments. While this provides a rich intra-institutional context, it also introduces constraints on generalizability. Differences in academic culture, institutional policies, student demographics, and assessment designs across universities may influence both the prevalence and the manifestations of dishonest behaviors. Consequently, the patterns learned by our models may not fully extrapolate to institutions with different educational norms or evaluation structures.

To mitigate these concerns, future work will actively incorporate multi-institutional datasets to broaden the diversity of training contexts, alongside integrating multimodal data streams such as proctoring videos, biometric logs, and behavioral interaction data. Such an expansion will not only capture more complex cheating strategies but also enhance the external validity and fairness of the detection models across heterogeneous academic settings.

7 Future Work

This study primarily focused on numerical marksheet data; our references to video and image-based proctoring remain prospective. We have not yet implemented or validated models using such modalities. As a preliminary step, we have begun collecting a pilot dataset comprising short exam session video clips and webcam snapshots to explore basic visual anomalies (e.g., frequent gaze shifts, secondary person presence) using simple CNN classifiers. Early exploratory runs on this limited data indicate promise but lack

statistical power. Future studies will robustly develop and validate image and video-based pipelines, incorporating both facial action cues and scene context, to complement our current temporal marksheet analysis. This cautious roadmap ensures that claims remain grounded in current evidence, with multimedia extensions framed as clear avenues for subsequent rigorous research.

Building on these insights, our next phase will develop an image-based proctoring framework that systematically analyzes student facial orientation, presence consistency, and surrounding scene changes to identify suspicious behavior patterns. By leveraging CNN-based facial landmark detection and environment change detection, we aim to construct robust visual profiles of test sessions.

Following this, we will advance toward a full video-based cheating detection pipeline. This will involve temporal sequence analysis to capture dynamic behaviors such as repeated look-away patterns, abrupt posture shifts, or collaborative gestures that static frames cannot reveal. Early scoping experiments using open-source video proctoring datasets (like the OULU-NPU benchmark) indicate that temporal models (e.g., 3D CNNs, BiLSTMs on extracted pose sequences) can effectively differentiate normal exam behavior from orchestrated deception.

Additionally, we intend to incorporate biometric authentication (facial verification or fingerprint matching) to ensure that the person taking the exam consistently matches institutional records throughout the session, addressing identity fraud.

Beyond technical advancements, future work will also explore unsupervised anomaly detection on multimodal data to capture previously unseen cheating strategies. Finally, we plan to execute longitudinal deployments across diverse academic environments to rigorously evaluate system impact, fairness, and adaptability over multiple academic cycles.

Collectively, these initiatives aim to create a holistic cheating detection ecosystem that combines marksheet analysis, visual behavioral monitoring, and identity assurance, substantially enhancing the reliability and scope of academic integrity systems.

8 Conclusion

This study explored the use of machine learning and deep learning models to detect potential cheating behaviors by analyzing discrepancies in students' marks across online and offline examinations. Against the backdrop of heightened concerns over academic integrity in online education, our work provides a data-driven framework to identify suspicious performance patterns that may warrant further scrutiny.

Leveraging a dataset spanning multiple disciplines and academic terms within a single institution, we developed a robust pipeline incorporating data preprocessing, feature engineering, and rigorous model training. Our analysis demonstrated that traditional ML approaches such as Random Forests and Logistic Regression offered solid baseline performance, achieving accuracies around 75% but with limited ability to capture sequential or temporal nuances inherent in cheating patterns.

In contrast, advanced DL architectures, especially LSTM and BiLSTM networks, delivered markedly superior results. The BiLSTM model, capable of processing input sequences in both temporal directions, achieved an accuracy of 97.61% with an AUC of 1.00, highlighting its exceptional capacity to discern subtle deviations in student behavior. Complementing these automated methods with human-in-the-loop validation, where academic experts reviewed flagged case further bolstered the practical fairness and credibility of the system.

It is important, however, to interpret these findings within the study's contextual boundaries. Although our models showed reduced false positive rates relative to simpler baselines on this dataset, these outcomes are intrinsically tied to the characteristics of a single institution sample and to class balancing via oversampling. Thus, the observed performance, including minimization of false positives, should be viewed as preliminary, warranting cautious generalization until validated across broader, multi-institutional datasets.

Additionally, by relying primarily on numerical marks data, this work inherently limits the spectrum of detectable cheating behaviors, potentially overlooking collaborative schemes, unauthorized resource use, or sophisticated behavioral cues that do not manifest as abrupt mark fluctuations.

Looking ahead, future research will address these gaps by incorporating richer, multimodal data streams. Integrating image and video analysis from online proctoring can enable the detection of nuanced behaviors such as gaze aversion, suspicious hand movements, or multiple individuals present during an exam, while preliminary explorations in our pilot setups indicate that such modalities can substantially enhance detection sensitivity. Further, embedding biometric verification (e.g., facial recognition, fingerprint scans) and leveraging unsupervised anomaly detection algorithms may uncover novel or evolving cheating tactics not captured by supervised approaches.

In sum, this work establishes a foundational, transparent pipeline for automated academic integrity monitoring that blends powerful DL models with expert oversight. By iteratively enriching data sources, expanding across diverse educational contexts, and embedding rigorous fairness audits, we aim to evolve this framework into a comprehensive solution that upholds both the efficacy and ethical principles vital for safeguarding academic standards in increasingly digital learning environments.

9 Declarations

Data Availability Statement

The dataset will be available on reasonable request from the corresponding author.

Author Contribution

M.M. and I.C. jointly conceived the research idea and developed the methodology for the study. M.M. contributed to conceptualization, formal analysis, methodology, data curation, and writing the original draft. I. C. took the responsibility of review, editing, supervision, and project administration.

References

[1] L. C. O. Tiong and H. J. Lee, "E-cheating prevention measures: detection of cheating at online examinations using deep learning approach–a case study," *arXiv preprint arXiv:2101.09841*, 2021.

[2] S. Habib and T. Parthornratt, "Anticipated and actual challenges pertaining to online delivery of university courses during covid-19 pandemic: The engineering faculty's experience at assumption university," in *2020 5th International STEM Education Conference (iSTEM-Ed)*. IEEE, 2020, pp. 5–8.

[3] P. M. Newton and K. Essex, "How common is cheating in online exams and did it increase during the covid-19 pandemic? a systematic review," *Journal of Academic Ethics*, vol. 22, no. 2, pp. 323–343, 2024.

[4] L. Zhao, Y. Zheng, J. Zhao, G. Li, B. J. Compton, R. Zhang, F. Fang, G. D. Heyman, and K. Lee, "Cheating among elementary school children: A machine learning approach," *Child Development*, vol. 94, no. 4, pp. 922–940, 2023.

[5] Y. Atoum, L. Chen, A. X. Liu, S. D. Hsu, and X. Liu, "Automated online exam proctoring," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609–1624, 2017.

[6] X. H. Nguyen, V. M. Le-Pham, T. T. Than, and M. S. Nguyen, "Proctoring online exam using iot technology," in *2022 9th NAFOSTED Conference on Information and Computer Science (NICS)*. IEEE, 2022, pp. 7–12.

[7] W. J. Hussar and T. M. Bailey, "Projections of education statistics to 2027. nces 2019-001." *National Center for Education Statistics*, 2019.

[8] X. Ren, "Investigating the experiences of online instructors while engaging and empowering non-traditional learners in ecampus," *Education and Information Technologies*, vol. 28, no. 1, pp. 237–253, 2023.

[9] C. Chen, J. Long, J. Liu, Z. Wang, L. Wang, and J. Zhang, "Online academic dishonesty of college students: A review," in *2020 International Conference on Advanced Education, Management and Social Science (AEMSS2020)*. Atlantis Press, 2020, pp. 156–161.

[10] N. Baijnath and D. Singh, "Examination cheating: Risks to the quality and integrity of higher education," *South African Journal of Science*, vol. 115, no. 11-12, pp. 1–6, 2019.

[11] T. Hodgkinson, H. Curtis, D. MacAlister, and G. Farrell, "Student academic dishonesty: The potential for situational prevention," *Journal of Criminal Justice Education*, vol. 27, no. 1, pp. 1–18, 2016.

[12] A. C. Özgen, M. U. Öztürk, and U. Bayraktar, "Cheating detection pipeline for online interviews and exams," *arXiv preprint arXiv:2106.14483*, 2021.

[13] S. D. Levitt and M.-J. Lin, "Catching cheating students," National Bureau of Economic Research, Tech. Rep., 2015.

[14] C. Cleophas, C. Hönnige, F. Meisel, and P. Meyer, "Who's cheating? mining patterns of collusion from text and events in online exams," *INFORMS Transactions on Education*, vol. 23, no. 2, pp. 84–94, 2023.

[15] F. Kamalov, H. Sulieman, and D. Santandreu Calonge, "Machine learning based approach to exam cheating detection," *Plos one*, vol. 16, no. 8, p. e0254340, 2021.

[16] J. Ranger, N. Schmidt, and A. Wolgast, "The detection of cheating on e-exams in higher education—the performance of several old and some new indicators," *Frontiers in Psychology*, vol. 11, p. 568825, 2020.

[17] R. Peytcheva-Forsyth, L. Aleksieva, and B. Yovkova, "The impact of technology on cheating and plagiarism in the assessment–the teachers' and students' perspectives," in *AIP Conference Proceedings*, vol. 2048. AIP Publishing, 2018, p. 020018.

[18] F. Noorbehbahani, A. Mohammadi, and M. Aminazadeh, "A systematic review of research on cheating in online exams from 2010 to 2021," *Education and Information Technologies*, vol. 27, no. 6, pp. 8413–8460, 2022.

[19] I. H. Sarker, "Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects," *Annals of Data Science*, vol. 10, no. 6, pp. 1473–1498, 2023.

[20] S. Kaddoura, S. Vincent, and D. J. Hemanth, "Computational intelligence and soft computing paradigm for cheating detection in online examinations," *Applied Computational Intelligence and Soft Computing*, vol. 2023, no. 1, p. 3739975, 2023.

[21] A. Balderas and J. A. Caballero-Hernández, "Analysis of learning records to detect student cheating on online exams: Case study during covid-19 pandemic," in *Eighth international conference on technological ecosystems for enhancing multiculturality*, 2020, pp. 752–757.

[22] D. Faucher and S. Caves, "Academic dishonesty: Innovative cheating techniques and the detection and prevention of them," *Teaching and Learning in Nursing*, vol. 4, no. 2, pp. 37–41, 2009.

[23] D. Von Gruenigen, F. B. d. A. e Souza, B. Pradarelli, A. Magid, and M. Cieliebak, "Best practices in e-assessments with a special focus on cheating prevention," in *2018 IEEE global engineering education conference (EDUCON)*. IEEE, 2018, pp. 893–899.

[24] B. Keresztury and L. Cser, "New cheating methods in the electronic teaching era," *Procedia-Social and Behavioral Sciences*, vol. 93, pp. 1516–1520, 2013.

[25] C. Y. Chuang, S. D. Craig, and J. Femiani, "Detecting probable cheating during online assessments based on time delay and head pose," *Higher Education Research & Development*, vol. 36, no. 6, pp. 1123–1137, 2017.

[26] M. AlSallal, R. Iqbal, S. Amin, A. James, and V. Palade, "An integrated machine learning approach for extrinsic plagiarism detection," in *2016 9th International Conference on Developments in eSystems Engineering (DeSE)*. IEEE, 2016, pp. 203–208.

[27] H. Cherroun, A. Alshehri *et al.*, "Disguised plagiarism detection in arabic text documents," in *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*. IEEE, 2018, pp. 1–6.

[28] H. Qiubo, T. Jingdong, and F. Guozheng, "Research on code plagiarism detection model based on random forest and gradient boosting decision tree," in *Proceedings of the 2019 International Conference on Data Mining and Machine Learning*, 2019, pp. 97–102.

[29] L. C. O. Tiong, H. J. Lee, and K. L. Lim, "Online assessment misconduct detection using internet protocol and behavioural classification," *arXiv preprint arXiv:2201.13226*, 2022.

[30] A. Nigam, R. Pasricha, T. Singh, and P. Churi, "A systematic review on ai-based proctoring systems: Past, present and future," *Education and Information Technologies*, vol. 26, no. 5, pp. 6421–6445, 2021.

[31] P. Shevale, V. Shitole, S. More, Y. Gaykar, and A. Gaigol, "Xampro (voice-based exam proctoring system)," in *2023 11th International Conference on Emerging Trends in Engineering & Technology-Signal and Information Processing (ICETET-SIP)*. IEEE, 2023, pp. 1–4.

[32] J. H. Y. Ho, D. Z. Tan, J. Y. Yap, K. P. Tse, M. F. B. Abbas, A. W. Y. Loo, and W. Goh, "Iot-enhanced remote proctoring: A new paradigm for remote assessment integrity," in *2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, 2023, pp. 197–198.

[33] M. De La Roca, M. Morales, and A. García-Cabot, "The impact of online proctoring on students' perception and level of satisfaction in a mooc," in *2022 IEEE Learning with MOOCS (LWMOOCS)*. IEEE, 2022, pp. 23–27.

[34] J. Plochaet and T. Goedemé, "Towards automatic proctoring of online exams using video anomaly detection," *Joint International Scientific Conferences on AI and Machine Learning*, 2022.

[35] H. A. Atabay and H. Hassanpour, "Abnormal behavior detection in electronic-exam videos using beatgan," in *2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*. IEEE, 2022, pp. 1–5.

[36] L. R. Bommireddy, R. T. Marasu, R. P. Karanam, and K. S. Sri, "Smart proctoring system using ai," in *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*. IEEE, 2023, pp. 591–593.

[37] G. J. Cizek and J. A. Wollack, *Handbook of Quantitative Methods for Detecting Cheating on Tests*. New York: Routledge, 2017.

[38] D. L. Olson, "Data set balancing," in *Chinese academy of sciences symposium on data mining and knowledge management*. Springer, 2004, pp. 71–80.

[39] M. Pérez-Ortiz, P. A. Gutiérrez, P. Tino, and C. Hervás-Martínez, "Oversampling the minority class in the feature space," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 9, pp. 1947–1961, 2015.

[40] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[41] S. Ertekin, J. Huang, L. Bottou, and L. Giles, "Learning on the border: active learning in imbalanced data classification," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 127–136.

[42] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2008.

[43] N. García-Pedrajas, "Partial random under/oversampling for multilabel problems," *Knowledge-Based Systems*, p. 112355, 2024.

[44] E. F. Swana, W. Doorsamy, and P. Bokoro, "Tomek link and smote approaches for machine fault classification with an imbalanced dataset," *Sensors*, vol. 22, no. 9, p. 3246, 2022.

[45] S. K. Kwak and J. H. Kim, "Statistical data preparation: management of missing values and outliers," *Korean journal of anesthesiology*, vol. 70, no. 4, pp. 407–411, 2017.

[46] E. Kock, Y. Sarwari, N. Russo, and M. Johnsson, "Identifying cheating behaviour with machine learning," in *2021 Swedish Artificial Intelligence Society Workshop (SAIS)*. IEEE, 2021, pp. 1–4.

[47] S. Sperandei, "Understanding logistic regression analysis," *Biochemia medica*, vol. 24, no. 1, pp. 12–18, 2014.

[48] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *Journal of clinical epidemiology*, vol. 49, no. 11, pp. 1225–1231, 1996.

[49] J. A. Ruiperez-Valiente, P. J. Munoz-Merino, G. Alexandron, and D. E. Pritchard, "Using machine learning to detect 'multiple-account'cheating and analyze the influence of student and problem features," *IEEE transactions on learning technologies*, vol. 12, no. 1, pp. 112–122, 2017.

[50] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling." in *Interspeech*, vol. 2012, 2012, pp. 194–197.

[51] G. Edholm and X. Zuo, "A comparison between aconventional lstm network and agrid lstm network applied onspeech recognition," 2018.

[52] W. Liu, W. Jing, and Y. Li, "Incorporating feature representation into bilstm for deceptive review detection," *Computing*, vol. 102, no. 3, pp. 701–715, 2020.

Manit Malhotra is a Research Scholar in the Department of Computer Science Applications, Panjab University, Chandigarh, and an Assistant Professor in the Department of Computer Applications, Government College of Commerce and Business Administration, Chandigarh, since 2019. A UGC-NET qualifier and the highest Gold Medallist of MCA (Panjab University, 2019), he has authored a book, published research in reputed journals including IEEE, and presented papers at numerous national and international conferences. With over 150 awards in coding and public speaking, he has also received multiple accolades for best research papers and poster presentations.

Professor Indu Chhabra Department of Computer Science, Punjab University, is the first professor in the northern region to earn a Post-Doctoral Fellowship (Computer Science) from a top ICSSR institute, Ministry of Education, Govt. of India, specializing in Neural Networks and Genetic Algorithms. She has received multiple honors, including the Young Scientist Award (2007), Award of Honor (2018), and Award of Excellence (2023). With nine Best Research Paper Awards, she serves as a project sanctioning expert for the Dept. of Science and Technology, Govt. of India, and reviews for IEEE, Elsevier, and Springer Nature. Author of four books and seven international book chapters, she has 91 publications and is on various government and semi-government selection committees.

Spatio-Temporal Ontological Query Processing in IoE Environments

Maza Abdelwahab*
LMSE Laboratory, Computer Science Department,
Faculty of Mathematics and Information Technology,
Mohamed El Bachir EL Ibrahimi University of Bordj Bou Arriridj, Algeria

Lyazid Sabri†
Intelligent Systems and Cognitive Computing Laboratory,
Faculty of Mathematics and Information Technology,
Mohamed El Bachir EL Ibrahimi University of Bordj Bou Arriridj, Algeria

Abstract

purpose: Leveraging ontologies to manage, analyze and understand the semantic context surrounding data generated by interconnected devices, sensors, and people in the Internet of Everything ecosystem. Provide early warnings for potential risks, such as health deterioration or unsafe behaviours.

Methodes: Ontology-based querying using chronological events enhances activity recognition and predicts future issues. Using temporal ontology and semantic reasoning ensures that queries are accurate and relevant.

Results: Combining spatial and temporal data with contextual awareness allows the system to assess the environment dynamically, perform adaptive processing, predict, and adjust its context-sensitive analyses.

conclusion: Contrary to the temporal Description Logic frameworks for dynamic context/event recognition and spatiotemporal concept representation, our Spatio-temporal querying approach further refines the system's responsiveness, enhances efficiency, and personalises relevant human-machine interaction.

KeyWords: Temporal ontology, activity recognition, context-aware, ontology-based querying, description logic, intelligent system, IoE ecosystem

1 Introduction

The promising research trends in Internet of Everything applications areas have led to the emergence of the known Internet of Robotic Things. In these environments, robots are designed to ensure complex cognitive tasks such as assisting and supervising dependent persons. These tasks require the manipulation of knowledge about the properties of objects and performing complex actions. An intelligent system must have advanced cognitive abilities to interpret context, recognize user activities and intentions, and make adequate decisions. Therefore, it is necessary to delve in human behaviour and understand why actions occur in specific sequences (i.e., time points and intervals) and spaces. As stated by the philosopher, Noël Carroll [1], the causality of later events and/or states of affairs depends on the earlier events. Consequently, finding correlations between events over time is an important aspect that leverages ambiguities of interpretation, which allows building up a kind of causal explanation automatically. Recent studies have addressed ontologies as a de facto solution for implementing intelligent systems for activity recognition and planning functions, tasks, and service composition ([2], [3], [4], [5]). Indeed, ontologies provide a vocabulary of concepts and properties, fostering a shared understanding of semantics among humans and machines. Although various methods for representation and reasoning over temporal data ([6], [7], [8], [9]) developed, they only deal with specific time intervals or time points. Even so, time points and semantic relationships between two- or a-time interval and a time point are not what they are designed for. Furthermore, we must handle that connectivity, such as event causality and goal. Despite this, no proposals for n-ary relations are included in OWL . Due to significant issues that remain unhandled by Ontology Web Language, it remains unsuitable for dynamic context/event recognition and spatiotemporal concept representation, expressing a chronological ordering between events and contexts. These points highlight the limitations of current methods ([10], [11], [12], [13]) and underscore the limitations of temporal description logic frameworks. One method to address this issue would be to use n-ary predicates to represent the evolution of knowledge and the chronological relationships between events and their contexts in both present and past. A statement like: "The robot observed that a person turned on the stovetop and left the kitchen towards the bathroom where he spent more than 25 minutes" is a complex task requiring consideration as a single indivisible entity. So, it can be challenging to fully describe this type of information using the usual binary Semantic Web languages such as Ressource Description Framework

*Email:abdelouahab.maza@univ-bba.dz
†Email:lyazid.sabri@univ-bba.dz

and OWL. However, explanatory role properties should then be necessarily introduced to represent a context fully. This paper has three contributions. First, the semantic annotation layer aims to describe an approach that permits the semantic description of heterogeneous entities that can change over time and interact with each other. Moreover, this layer deals with the semantic modelling of raw sensor data extracted from different sources, facilitating better analysis and reasoning. The second contribution is the usefulness of the narrative model of the narrative knowledge representation (NKRL) used for the first time in ambient intelligence ([14], [15]). It consists of adding to the usual ontologies of concept HClass (Hierarchical Class) as generalisation/specialisation structure, the ontology of events called HTemp (hierarchical Temporal) ontology. The third contribution is designing a Query-Processing Mechanism (QPM) about activity recognition and dynamic events/contexts,figure1. The QPM uses hierarchical structures of semantic predicates and functional roles in HTemp. Therefore, NKRL overcomes the disadvantages of Semantic Web Language by providing the HTemp ontology.

NKRL provides a means to reconstruct the context from the potential semantic relationships about events occurrences in both past and present time, as well as their spatial-temporal dependencies, demonstrating its adaptability to various scenarios in Internet of Everything ecosystems. The QPM relies on two kinds of rules: transformations and hypothesis to determine contexts and recognise human activities and intentions. The rules are concisely described based on the application domain and specific sensor outputs. When it is impossible to find explicit knowledge within the knowledge base using hypothesis rules, the QPM combines the two classes of rules to discover all the possible implicit information associated with the original context. Transformation rules try to adapt the search pattern (*query1 = intial query*) by automatically transforming *query1* into one or more sub-queries *q11, q12 , ..., q1n* that are not strictly equivalent but only semantically close to the initial query. The paper is structured as follows: Section 2 presents a general-related work on ontology-based knowledge representation and query processing for activity recognition. Section 3 introduces a novel knowledge representation and query processing for IoE ecosystems. Section 4 describes the method for recognising activities using specific scenarios. Section 5 presents an evaluation and scalability of the proposed approach. Finally, section 6 outlines discussion, a conclusion.

2 Related work

Some significant conceptual and practical issues still plague the use of W3C languages regarding the creation and processing of rules. Despite the important contribution these languages have made, for example, in simplifying the management and interpretation of contexts through the use of semantic representations and querying/reasoning tools. Description Logic (DL) has become a formalism in symbolic knowledge representation because it offers complete reasoning and is supported by tools (e.g., Pellet)). OWL 2 have extended the original OWL 1 with a few practical features. The three OWL 2 profiles can offer some advantages in particular application scenarios but are more restrictive than the full OWL 2 DL. OWL 2 QL enables conjunctive queries to be answered similarly to the standard relation database principle. In this last case, reasoning will always be sound, but it may not be complete (that is, it is not guaranteed that all correct answers to queries will be computed). Researchers have explored expanding the DL is syntax to include the OWL language.

2.1 Knowledge representation

Key-value-based techniques have been proposed by [16] using a simple data structure to describe a sensor's outputs and, therefore, trying to represent an activity. Moreover, [17] proposed hierarchical structures relying on deep neural networks. Unfortunately, all those approaches are very limited in handling the interoperability in activity recognition systems. Various research for activity recognition approaches combining ontolgies and rule-based models or machine learning, such as [18]. Authors have relied on naive Bayesian models to represent objects to infer the possible actions on these objects and, thus, deduce the associated activity. This approach is based on the semantic relations between everyday actions that can be executed through these objects. However, the authors did not use an ontology but a taxonomy of concepts and did not implement an ontology of roles. A several symbolic representations of the user environment and ontological reasoning have been proposed in the literature to deduce activities according to a set of preselected actions using the OWL ontology. They exploited human-object interaction and, therefore, events using the flow of sensors for activity recognition. These systems are excellent at contextualising activities by establishing connections between objects, actors, and environments, a skill that is essential for accurately interpreting human behaviour. However, the ontology paradigm historically emphasises structure and lacks behavioural components such as role. Recent research underscores the pivotal role of ontologies in enhancing robotic autonomy. An in-depth review delves in to their contribution to knowledge representation, task planning, and adaptability in dynamic environments, empowering robots to reason about their environment and act reliably [19]. In the Internet of Things (IoT), a context-aware edge computing framework, CONTESS, harnesses context to optimize resources by reducing latency and adapting processing at the network's edge [20]. The semantic representation of robotic manipulations has also made significant strides through knowledge graphs. A multi-layer model describes objects, actions, and effects, facilitating automatic task planning [21]. In parallel, motion planning in dynamic environments benefits from context-aware human trajectory prediction, enabling robots to anticipate behaviors and avoid collisions [22]. This idea is extended to autonomous driving, where a multimodal framework uses neural networks to

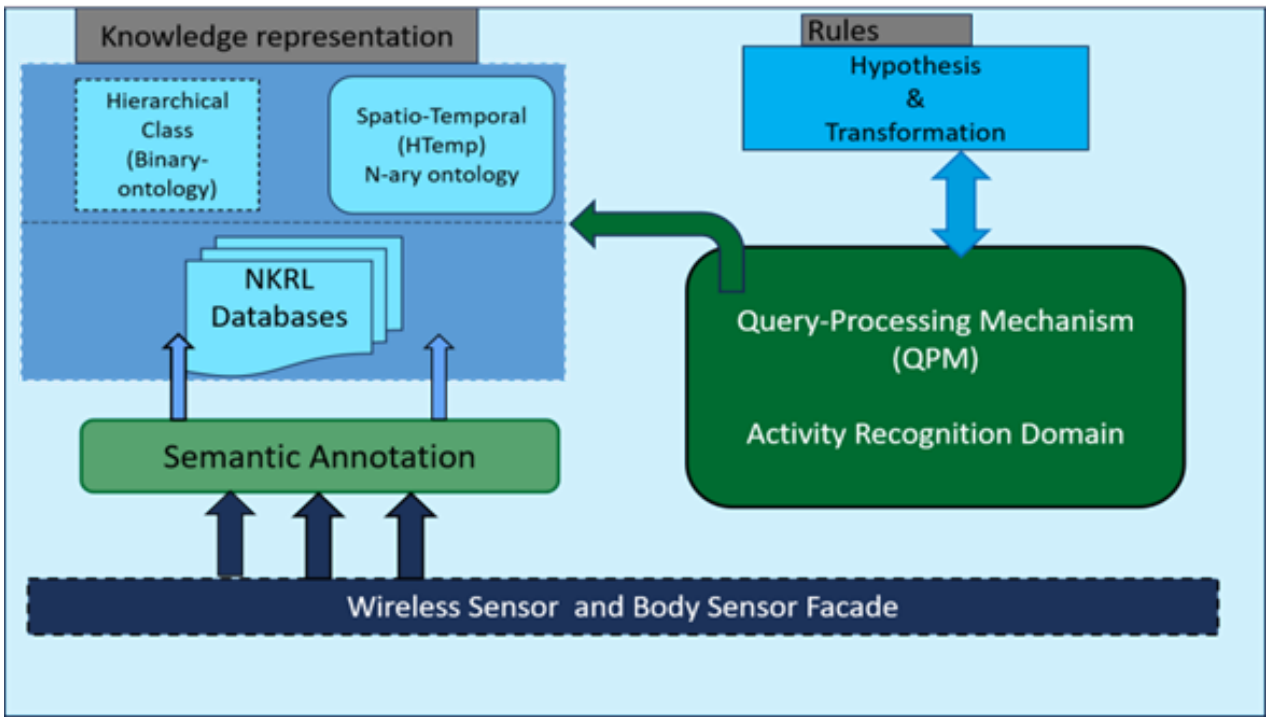


Figure 1: General activities and intentions recognition platform.

predict trajectories in heterogeneous environments, considering social intentions and interactions [23]. The integration of ontological frameworks in robotic task management is a significant advancement that promotes greater autonomy. For instance, one system proposes robotic task processing based on semantic modeling, which combines perception, reasoning, and execution [24]. This means that the robot can understand its environment, make decisions based on that understanding, and execute tasks accordingly. The ORKA ontology takes this a step further by formalizing the acquisition of knowledge from sensors and perceptions, enabling coherent information exploitation [25]. This work is further strengthened by approaches that combine ontologies and rules for collaborative task planning, such as disassembly in remanufacturing, where human and robotic roles are optimally coordinated [26]. In mixed reality, OWL ontologies have facilitated semantic mediation between humans and robots, enabling seamless and contextualized interaction during collaborative assembly [27]. This logic extends to ethical considerations, with proposals for the standardization of ontologies aimed at framing the decisions and behaviors of autonomous systems from a moral point of view [28]. This significant innovation also consists of integrating language models such as ChatGPT into robotic architectures (ROS), making possible a more intuitive and expressive interaction with users [29]. The latter is based on ontology formalism but does not integrate negation or define spatiotemporal relationships. Contextual gestural interactions also make an essential contribution. For instance, a dual-flow model allows the efficient recognition of control gestures. This model uses two streams of information, one for the hand's trajectory and the other for the hand's shape, to strengthen human-robot cooperation in complex environments [30]. Finally, an advanced approach to contextual indoor navigation for robots integrates semantic, spatial and temporal dimensions, allowing intelligent exploration of unknown environments with increased adaptability thanks to context modeling Incorporating ontological frameworks into robotic task management promotes greater autonomy. For example, one system proposes robotic task processing based on semantic modeling, combining perception, reasoning, and execution [31]. These contributions, derived from diverse work, demonstrate a convergence toward intelligent, adaptive, and human-centered robotic systems, supported by robust semantic structures and advanced contextual processing, and remain a relevant and promising avenue for better context management.

2.2 Query-processing mechanism

Few works have been done on developing query languages and inference rules based on temporal description logic, and Most of these works are based on Allen's temporal logic. Among these works, a temporal language TL-OWL an OWL-2 DL ontology of temporal concepts based on the idea of time interval and combing 4-D fluents [32]. Nevertheless, 4D-fluents maintain OWL expressiveness and reasoning support but still suffer from data redundancy [12]. Furthermore, unfortunately, TL-OWL ontology does not support temporal relations or consistency checking and is not compatible with OWL inferencing and querying tools. The authors of [10] have

developed a semantic geospatial database system, introducing two sub-languages built on top of RDF and SPARQL query language. They have introduced t-SPARQL, an extension that can be directly mapped to standard SPARQL to express temporal queries. While t-SPARQL has not yet managed the temporal features, its potential for future development is promising. The principle of reification in ([6], [7], [13], [33], [37]), which depicts n-ary relations, has a significant problem of data redundancy. The authors in [34] propose knowledge reification as a solution for representing complex relationships and multilevel abstractions using the property graph model. The SWRL and SQWRL [35] rules languages are employed in both approaches [7], [33]. Using inference rules is a fundamental part of knowledge management and a crucial component of the reasoning process. These rules are either written in the SWRL language or incorporate the Horn clause and an OWL- DL. SWRL Temporal Ontology, a significant extension of the SWRL language, allows the annotation, reasoning and querying of temporal knowledge bases. This ontology's proposition, instant, and time interval concepts are crucial for presenting temporal knowledge. It is important to note that the representation knowledge may be complex and better suited for describing temporal entities than the temporal context. Moreover, SWRL raises several limitations, such as the lack of negation. The OWL-Time , a W3C recommendation since October 2017, is a powerful tool that provides a vocabulary for expressing facts about topological (ordering) relations among instants and intervals. OWL-Time does not support dynamic events for representing object properties that change over time. To the best of our knowledge, no reasoning tools allow us to infer new temporal data. Despite these constraints, OWL- Time is still a valuable resource for describing the temporal content of web pages and the temporal properties of web services. The main reasons are: 1) OWL and RDF language are based on binary relations that supply connect two instances, and 2) It cannot be combined with the existing OWL tools [36].

3 Methodology

3.1 Modeling knowledge

HClass is an ontology of concepts. It encompasses more than 2700 concepts. It is identical to the binary OWL ontology. A generalization/specialization structure can be created by using HClass to represent general concepts. The process of naming a concept involves using lowercase symbolic labels and an underscore, like human being, artifact, doctor, sensor and robot . Like OWL, HClass also contains instances (individuals) which are represented in the upper case, including an underscore symbol; For example, BLOOD-SENSOR-PRESSURE-2 SENSOR- ECG-1 are examples of the body sensor concept, and WHEEL-CHAIR-3 is an example of the artefact concept. The nodes in HTemp are hierarchically connected as n-ary structures. This ontology is defined as the formal depiction of elementary events. Our approach distinguishes between an elementary event and a complex event, which describes an entity's behaviour (motions, actions, temporal events, etc.). For example, turning on the coffee machine early in the morning and opening the door are elementary events. However, if the robot moves towards the space where a human is located and interacts with him, it is a complex event. Figure2 depicts the general structure of HTemp ontology divided into seven branches called templates or predicates (MOVE, PRODUCE, RECEIVE, EXPERIENCE, BEHAVE, OWN, EXIST). The BEHAVE predicate, a crucial concept in our understanding of actions and behaviours, represents the actions or behaviours of one or more individuals. On the other hand, EXIST indicates an entity's presence in a given space. EXPERIENCE is typically employed to describe an event that affects an individual, like illness, success, accident, etc. MOVE, a versatile predicate, describes many actions like moving, sending, etc. The OWN predicate can represent the notion of ownership between entities or the state of an entity. The PRODUCE predicate describes the execution of a task, activity, or other action. RECEIVE describes events related to the reception of information.

Table 1: Elementary events (ioe.e85) vs complex events (ioe.m156).

Description	narrative event
The robot gives its assistance by moving itself towards the bathroom where a human is localized and tries to interact with him.	ioe.m156) MOVE SUBJ ROBOT_KOMPAI: (KITCHEN_1) OBJ ROBOT_KOMPAI: (BATHROOM_1) MODAL speech_interaction CONTEXT potential_risq date-1: 2024/11/25/15:25 date-2:
On 2024/11/25, at 14:40, the system observes that a stovetop in the kitchen is turning on in the kitchen denoted respectively by the symbols STOVETOP_1 and KITCHEN_1.	ioe.o145) OWN SUBJ STOVETOP_1:(KITCHEN_1) OBJ property_ TOPIC TURN_ON {obs} date-1: 2024/11/25/14:40 date-2:
On 2024/11/25, at 14:58, the system observes that the temperature on the kitchen stovetop has risen	ioe.e85) EXPERIENCE SUBJ (SPECIF temperature KITCHEN_1) OBJ growth_ {obs} date-1: 2024/11/25/14:58 date-2:

Each Template can be customized to derive the new templates that could be needed for a particular application. HTemp ontology contains 165 templates. Each branch of template contains seven generic roles (subject(SUBJ), object(OBJ), SOURCE, MODAL, TOPIC, CONTEXT, Beneficiary (BENF)). The space where an event/situation occurs and temporal knowledge are respectively described by location and modulators (described in more detail in the next section). Modulators represent the (start, end, duration) of a given event/context. A role or a variable defined in square brackets ([ ]) are optional elements. In figure 2, the SUBJ, MODAL and TOPIC roles and (var1, var3, var5 and var6) are mandatory, while SOURCE, CONTEXT, roles, and variables (var2, var4, var7) are optional. The variables var1, ..., and var7 represent constraints allowing us to check that the values assigned to each variable when a Cognitive Behaviour Template

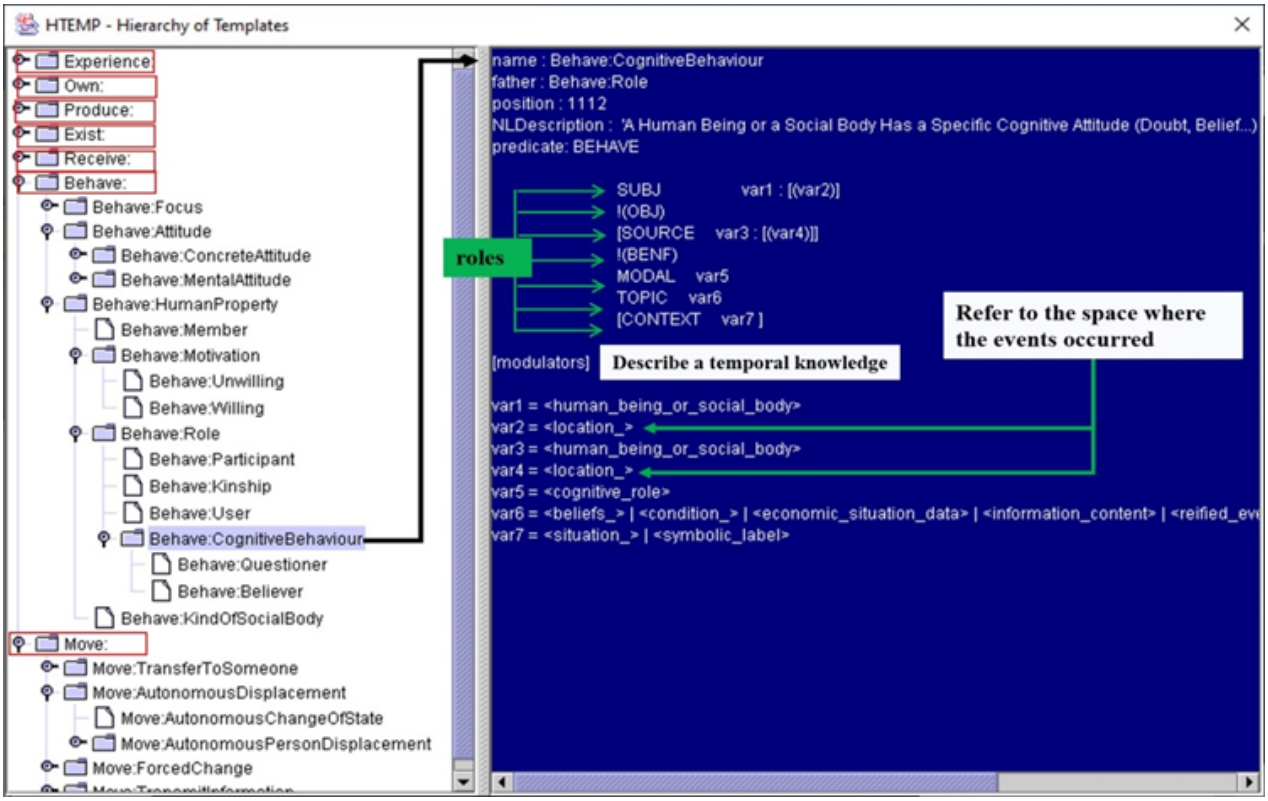


Figure 2: General HTemp ontology structure and Cognitive Behaviour structure.

is instantiated correspond to (concept, sub-concepts) defined in the HClass ontology. Table 1 depicts three examples of elementary events and complex vents. Where each template has a unique symbolic label (SymL) identifying a given template, some examples of SymL : (ioe.m156), (ioe.o145).

3.2 Spatio-temporal representation

According to [37] narrative events are those that take place in reality. As for [38], a narrative event provides the classical theory of narratology. A logical and chronological sequence of events makes up the fabula entity. The story entity is a fragment of fabula arranged into a new sequence. Finally, the narrative describes how events are narrated in a given language, media, signal, etc. In our approach, the Allen interval's logic can be recreated relying on two properties (date 1) and (module):

1. 1. The property (date 1) represents the event that begins at timestamp t1;
2. Date 2 is the property that signifies the maximum time limit for the event at timestamp t2;
3. Temporal attributes can be associated with temporal modulators like begin, end, and observe (obs) to mark the start or end of an event;
4. Point time is a time stamp that indicates that the date associated with date-1 is solely a specific point in the temporal interval associated with the event. The second property, date-2, is empty;

Table 2 shows two examples, the narrative event denoted by (ioe.b148) expresses that the symbol DAVID-1, which is used as filler of the SUBJ(ect) role, represents a human who is localized at the bathroom denoted with the symbol BATH-ROOM-1, the user filler of the Modal role describes that INDIVIDUAL-PERSON-1 is performing the activity (using the bathroom's shower tap) described in the TOPIC role as SHOWER-TAP-1. The property (date 1) depicts a specific time-point within the temporal interval corresponding to an event. As for (ioe.o25), throughout the scenario, DAVID-1 is the house owner denoted with HOUSE-1.

3.3 Chronological Knowledge representation

Binding narrative, a structure used to link together several events/contexts, taking into account semantic linking, are formalized by the binding operators. These operators, such as GOAL, COORD(ination), and CAUSE, play a crucial role in formalizing the logical semantic link between the narrative events using their symbolic labels (SymL). Furthermore, they allow describing complex IoE scenarios. The binding narrative can be expressed as follows:

$$(bind.operator\ [\ SymL_1\ o\ SymL_2\ o\ SemL_3 \ldots\ SemL_i]) \quad (1)$$

Table 2: The location where the scenario takes place is depicted in this narrative event. The house referred to as HOUSE 1 here belongs to INDIVIDUAL PERSON 1

narrative event
ioe.b148) BEHAVE SUBJ DAVID_1: (BATH_ROOM_1) MODAL user_ TOPIC SHOWER_TAP_1 date-1: 2024/11/25/15:03 date-2:
Behave:CognitiveBehaviour
ioe.o25) OWN SUBJ INDIVIDUAL_PERSON_1 OBJ HOUSE_1 date-1: 2024/05/25/10:00 date-2: Own:ConcreteResource

Formul 1 denoted a binary structure under the list of arguments SemL. The SymL corresponds to a symbolic label or recursively to sets of labelled lists in (Equation1). For instance, the narrative event ioe.b148 allows the robot to determine David's presence in the bathroom. Simultaneously, the narrative ioe.e85 indicates that the oven is in use, Table 1. In response, the robot sends a proposal to turn off the stovetop, a crucial action to prevent a potentially hazardous issue. This decision depicts a complex event that should be separated into three formal elementary events (derived from three different templates of the HTemp ontology):

- The temperature on the kitchen stovetop has risen (ioe.e85, Table 1);
- The robot takes note that David is not in the kitchen since he is in the bathroom (ioe.b148, Table 2);
- Provide an early warning, the robot moves towards BATHROOM_1 where the person is localized (ioe.m156 depicted in Table 1);

Using the COORD operator, the narrative events (ioe.m148) and (ioe.e85) can be linked to represent the entire narrative described by (ioe.c1), Table 3. So, the full description of these events is represented by the unique narrative event (ioe.s2).

4 Experiment and results

This section provides a detailed explanation of how the inference process is implemented. We, therefore, exclude aspects such as modelling and rule editing tools that are not necessary for the system to run, as they are mostly used during the design phase. We explain thorough knowledge acquisition methods, the process of integrating perceptual information into the knowledge base, and general query processing. Context recognition requires the fundamental knowledge provided by HClass and HTemp ontologies. The HClass ontology comprises 2700 concepts, while the HTemp ontology features 165 templates.

4.1 A use case scenario

The following will describe a scenario demonstrating the proposed approach in a practical, real-world context. Identifying situations and providing customized assistive and monitoring services in elderly healthcare can be challenging for any system if it cannot capture and comprehend chronologically related events. In our scenario, the robot not only gathers real-time information about the senior citizen's actions but relies on narrative querying-processing, demonstrating the effectiveness of a high level of understanding of the activities. The system is responsible for identifying the activity the person is engaged in and interpreting the associated risks. Let us now assume that David, a senior citizen living alone, wishes to prepare a meal which involves using appliances such as a stovetop and various kitchen utensils like pots and baking dishes. After 20 minutes, David heads to the bathroom and opens the shower tap. A sensor installed on the shower tap confirms when it's open, which allows the robot to determine David's presence in the bathroom. At the same time, the oven's temperature sensor detects an increase in temperature, which indicates that the oven is in use and there is no one around. The robot concludes that David cannot be in two different locations simultaneously since David is taking a shower in the bathroom and the stovetop is turning on. The robot, acting as a vigilant companion, moves towards where David is localized and tries interacting with David by sending an audio notification to suggest turning the stovetop off. David did not respond immediately. Two minutes later, the robot tries to ensure everything is okay and tries to confirm David's health condition. Establishing dialogue-based interaction with David will help collect information about his health. If David does not interact with the robot, he is considered unconscious, and consequently, the current context corresponds to an emergency. Let us clarify why these analyses are crucial.

1. Chronological analysis involves understanding the sequence of events, such as moving from the kitchen to the bathroom and the time spent there;
2. If the person stops moving, the system will recognize a potential issue and react accordingly;
3. Consider suggesting or taking action, like turning off the stovetop, based on a time interval (e.g., after a long cooking session or be a while in the bathroom);
4. The chronological analysis significantly enhances safety by proactively preventing forgotten cooking sessions, thereby ensuring a secure environment free from potential fires or accidents;

4.2 General Querying-processing mechanism

The following equation governs the handling of all inference rules:

$$S \ iff \ Y1 \ and \ Y2 \ :: \ Yn \tag{2}$$

Where S is the event/context to infer and Y1,..., Yn represent the reasoning steps. X, Y1,..., and Yn are modeled as instances of the template (narrative event). Y1 is called condition in

Table 3: Binding narrative events.

It is clear that there is a logical connection between ioe.e85 and ioe.b148.	ioe.c1) (COORD ioe.b148 ioe.e85)
ioe.c1 triggers the narrative event that is described in ioe.m156	ioe.s2) (CAUSE ioe.c1 ioe.a156

hypothesis rules and called antecedent in transformation rules. A transformation rule contains an antecedent (i.e., a left-hand side) representing the search query to transform and one or more consequents (i.e., right-hand sides) representing search patterns for which a QPM will substitute the query.The reasoning step Yi is started once the reasoning step Yi-1 has succeeded. The Yn (Equation 2) denotes the leaf in the tree structure, which symbolizes the success of the reasoning process. A QPM (figure1) component converts during a reasoning steps a search pattern derived from the variables and their values into search patterns Si that attempt to match and unify these queries with the knowledge stored in the knowledge base. The ontology HClass which represents a higher-level abstraction within the system allow adapting each concept/individual that occurs in the query to all subsumed concepts/indivduals.

Query Formulation: In this context, the antecedent refers to the condition or situation that prompting the system to create a query to understand the situation better. For example, the senior person has not moved from the bathroom for an unusually long time. This would involve applying semantic and chronological analysis, as well as correlating other factors, such as health status, time, and location.

4.3 Chronological and Semantic Analysis of Events

David's failure to hear the audio notification message results in his unawareness of the robot's interaction. This breakdown in communication disrupts the robot's on David's interaction, leading it to assume that David is in danger and the situation is an emergency. This scenario underscores the need for deep reasoning about spatiotemporal events, semantic analysis, and past and ongoing events. It also reiterates the importance of human-robot communication in the robot's decision-making process, as it is a key take away from the scenario.

4.4 Extraction of implicit observations

The hypotheses rules and transformation rules explain the causal reasoning by extracting and transforming relevant information from the knowledge base. The first query is adjusted to obtain relevant information or infer new causal relationships from existing data, enabling the creation of a narrative that explains the triggering alarm.

X1) Initial request (search pattern)
PRODUCE
SUBJ(ect): robot_
OBJ(ect): triggering_
TOPIC: alarm/control_tool

Table 4: Since David did not respond, the robot triggered an alarm

Description	Narrative event
Narrative event representing the initiator (agent) who triggers the alarm	ioe.p158)PRODUCE SUBJ ROBOT_KOMPAI OBJ triggering_ TOPIC emergency_alarm CONTEXT EMERGENCY_SITUATION_1 date-1: 2024/11/25/15:28 date-2: is instance of Produce:PerformTask/Activity

The NKRL search patterns operate like database queries in conventional systems, such as those used in information retrieval (IR). Similar to database query (e.g., in SQL), a pattern in NKRL enables systems to query and obtain answers directly from the knowledge base. Nevertheless, in our approach, a pattern is a formalized representation of a query that may involve logical relationships, constraints, and conditions expressed in a knowledge representation language as instances of HTemp ontology. The pattern is used to search for information, facts, or relationships within a semantic or knowledge-based system. Therefore, when the reasoning process is performed, the explicit variables in the template are replaced with concepts (abstract categories like "person," "robot," or "location") or individuals (specific entities like "DAVID_" "BATH_ROOM_1," or "STOVETOP_1"). The constraints imposed on these variables ensure that the substitute is consistent with the knowledge base. For instance, if a template has a variable "vari" that represents a "location," only concepts or individuals classified as a location in the knowledge base would be valid replacements for this variable. In a narrative-based knowledge base, all events might be represented as structured statements or facts, often involving temporal or causal relationships (e.g., David be present in the bathroom since 15h03, The temperature increase, David heads to the bathroom and opens the shower tap). For example, if the search pattern asks for events involving David, the system will also check if the symbol "DAVID_" is a valid instance of the person_ concept or any of its subclasses (like human_being, owner_, etc.).

The query (X1) plays a crucial role in defining the event. The search pattern defined by the conceptual predicates (PRODUCE with the roles of SUBJ and TOPIC) will result in a set of narrative events that match with the specified concepts (i.e., the output of the query will consist of all instances where the robot_ is associated with the production of an alarm/control_tool).

The system symbolised by (ioe.p158, Table 4) depicts that ROBOT_KOMPAI is the agent responsible for triggering the alarm. The consistency-checking mechanisms validate the symbol robot_ in Table 4 by relying on the HClass ontology and

the constraint associated with variable var1 in the hypothesis rule. This component checks that the robot_ symbol is an instance of the alarm/control_tool concept and establishes a hierarchy of concepts from the generalisation/specialisation relationship between the emergency_ alarm concept and the alarm/control_tool concept. The inference process continues its reasoning by attempting to verify the step indicated by Y1, which corresponds to condition 1 of the hypothesis rule. The new pattern is produced (see pattern (Y1)) by utilising the value var2 = human_being and the var1 = robot_ symbol. Condition 1 is used to check that the filler represented by ROBOT_KOMPAI is an agent (i.e., subconcept of control_tool) and monitoring system.

Y1: condition 1) PREDICAT: OWN
SUBJ(ect): ROBOT_1
OBJ(ect): control_
BENF: human_being

4.5 Using a transformation rule

Based on an ontology-based system, the query (Y1), the robot tries to find direct matches or relevant data. Since the direct search might not yield a valid or concrete result, the robot employs a transformation rule. This rule infers implicit knowledge not directly available in the knowledge base but can be derived logically. Applying the transformation rule, the system finds a form of knowledge not directly queried as implicit knowledge. The narrative event (ioe:p159), Table 5 denotes a specific event where someone is in the bathroom. The property detection is a role that holds the "object" of the event as a filler of the OBJ(ject) role. DAVID represents an individual human being. So, DAVID is the filler of the Topic. Thus, the querying processing infers that DAVID (a human being) is in the bathroom. The latter knowledge is not directly found through the query but is derived through implicit knowledge inferred from the system's transformation rule. The first condition of the hypothesis rule has been satisfied, and the reasoning process can now proceed with the processing of condition 2. In this step, the inference engine tries to find within the knowledge base any information indicating that the robot has attempted to establish a dialogue with DAVID (i.e., David as a person), thereby creating the search pattern (Y2).

Table 5: Results for transformation rule 1

Description	Narrative event
Narrative event specifying that DAVID is located in the bathroom.	ioe.p159) PRODUCE SUBJ ROBOT_KOMPAI OBJ detection_: BATH_ROOM_1 TOPIC DAVID_ date-1: 2024/11/25/15:03 date-2: is instance of Produce:Assessment/Trial

The first condition of the hypothesis rule has been satisfied, and the reasoning process can now proceed with the processing of condition 2. In this step, the inference engine tries to find within the knowledge base any information indicating that the robot has attempted to establish a dialogue with DAVID_ (i.e., David as a person), thereby creating the search pattern (Y2).

Y2) PREDICATE BEHAVE
SUBJ(ect) :ROBOT_KOMPAI :
MODAL(ity) : user_
TOPIC : robot_
CONTEXT : (SPECIF control_ DAVID_)

Actions/Relations: Semantic Representation

- **Communication (Robot, David**: The robot plays a crucial role in the communication process, being the entity responsible for interacting with David;
- **Modality (Robot, Touch Screen, David)**: The robot's touch screen serves as a powerful tool, enabling DAVID to communicate effectively;
- **Notification (David, Touch Screen, Help)**: A message was sent to David to inform him that he can request assistance using the robot's touch screen.

Transformation Rule 2

The following formal representation provides a clear and concise explanation of how the message is transmitted, who receives it, and the communication mode.

- MOVE(ROBOT_KOMPAI, DAVID_, "You can use the touch screen to request help"): DAVID_ is being notified by the robot that he can use the robot's touch screen to request help;
- BENF(ROBOT_KOMPAI) : It is evident that ROBOT_KOMPAI is the intended recipient of the message if DAVID_ responds;
- MODALITY(touch_screen) : states that the robot's touch screen is the communication mode;

Table 6: Results for transformation rule 2

Description	Narrative event
David can use the touch screen to interact with the robot	ioe.m145) MOVE SUBJ DAVID_:BATH_ROOM OBJ confirmation_statement BENF ROBOT_KOMPAI TOPIC (SPECIF assistance_ ROBOT_KOMPAI) date-1: 2024/11/25/15:25 date-2: Produce:Assessment/Trial

var4 = emergency button in (Y3). The search pattern (Y3) is designed to explore past events (according to their temporal interval) and find narrative events that indicate DAVID should press the emergency button. The search pattern (Y4) aims to retrieve explicit knowledge indicating that the emergency button is an actuator embedded in the robot.

Y3) PREDICATE PRODUCE
SUBJ(ect) : DAVID_

OBJ(ect): button_pushing
TOPIC: emergency_alarm

Y4) PREDICATE OWN
SUBJ(ect): SOS_BUTTON_1
OBJ(ect): property_
TOPIC:(SPECIF part_of (SPECIF alarm/control_tool ROBOT_KOMPAI)

The search pattern (Y3) derives the answer depicted in Table 7. The oblig(action) modulator expresses obligations, permissions, and prohibitions in formal logic to validate an emergency situation. The narrative event (i.e., ioe.o24) indicates that the SOS BUTTON 1 button is part of the robot's touch screen, and this relationship is established using a "part of" property. After treating condition 5 of the hypothesis rule, the inference engine will verify that David did not press the emergency button to explain why the alarm was triggered. The (Y5) search pattern below is used to infer this knowledge.

Y5) PREDICATE PRODUCE
SUBJ(ect) : DAVID_
OBJ(ect): button_pushing
TOPIC: SOS_BUTTON_1
{negv}

A crucial aspect of our work is the reasoning process, which is significantly driven by a formal narrative representation. Modulator negv is a formal narrative mark of negative events in our querying-processing system. It represents negation denoting an event's negation (in this case, not pushing the emergency button). The rule processing hypothesis, derived from the (X1) initial request, plays a pivotal role in recognizing the emergency context situation. The successive reasoning process, crucially involving the consideration of missed actions and the overall chronological of events, is instrumental in understanding the sequence of events that led to the triggering an emergency situation. According to the knowledge base's ioe.m160, Table 6 event, the robot offered David assistance, but he didn't respond, as evidenced by the ioe.p161 event, Tables 7.

Table 7: Formal narrative mark of negative events in our querying-processing system

Description	Narrative event
Narrative event specifying that the emergency state has been triggered because David did not push the emergency button after the fall has been observed.	ioe.p161)PRODUCE SUBJ DAVID_ OBJ button_pushing TOPIC DAVID_ CONTEXT LIVE_SAVING_BUTTON_1 {negv} date-1: 2024/11/25/15:05 date-2: is instance of Produce:PerformTask:Activity

5 Evaluation and scalability

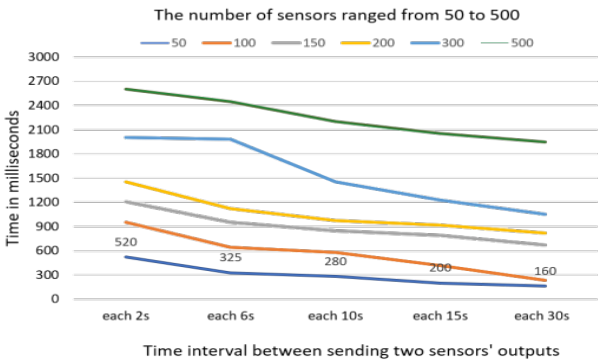
The purpose of the use case is to assess the proposed framework's performance in real-time, with a focus on response time and emergency context processing as follows:

- **Detecting Inactivity**
  1. **Goal**: Determine if the system can recognize when someone has left an activity, interrupted it, or been inactive for a specified period, and categorize it as a potential emergency;
  2. **Expected Action 1**: In order to respond, the system should activate an emergency protocol;
  3. **Expected Action 2**: To ensure user safety and prevent accidents, the system should recommend preventative safety measures (such as turning off the stovetop;

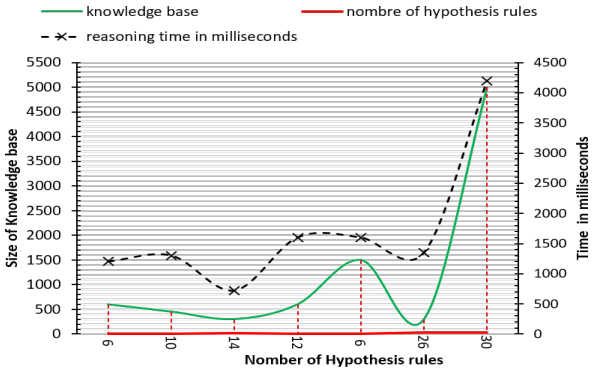
- **The Framework's evaluation criteria**
  1. **Real-Time Responsiveness**: What is the system's response time to recognizing inactivity or dangerous contexts and taking action?
  2. **User Trust and Intervention**: User Trust and Intervention : The system's ability to suggest or take preventive actions without constant user intervention is dependent;

Through our narrative querying-processing approach, the system can be both responsive and able to prevent accidents in real time, while also taking into account the safety of the user. The narrative model balances a trade-off between reasoning time and the amount of context knowledge inferred. The model can infer a broader and deeper understanding of implicit knowledge while taking more time to combine hypothesis rules with transformation rules. Recognizing complex situations or removing doubts is crucial in emergency management. The effectiveness of this approach lies in its ability to recognize complex and specific contexts, particularly in scenarios that do not require immediate response times but require deep contextual understanding. Emergency management and doubt removal require a response time of 3.8 seconds to recognize context figure 4, part (b). The querying-processing approach operates efficiently for real-time applications because it falls within an acceptable range. The response covers the time it takes an Abstraction Layer to process sensor outputs, encode them, and add facts to a knowledge base.

Our evaluation of scalability regarding sensor outputs was comprehensive. We developed a set of synthetic scenarios that incorporate HClass concepts and up to 30 hypothesis rules. We also included 24 transformation rules. The number of sensors ranged from 50 to 500. We conducted rigorous testing of the platform multiple times for each scenario and measured the average execution time across various parts of the architecture. The effectiveness of the proposed representations is measured by relying on both time intervals and points to recognize an activity effectively. Therefore, elementary events containing points and intervals were exploited to measure the response times of the reasoning process. First, we sought to infer contexts based solely on hypothesis rules. Subsequently, the semantic relationships extracted from the hypothesis rules were combined with transformation rules. The experiments were conducted on



(a) Real time Semantic Annotation



(b) Average reasoning time with hypothesis rules.

Figure 3: Evaluate the scalability of a system concerning sensor outputs while considering only hypothesis rules.

a PC with the Intel Core i5 processor, Dell Latitude 5550 15p, 16GB of RAM, and a 500GB SSD. It is crucial to point out that all the executions were done in a single-threaded way. Each test run was designed to contain 6-30 hypothesis rules and 12-24 hypothesis and transformation rules.

Our approach is versatile and can be applied across domains where sensor data plays a pivotal role in decision-making. For instance, a system with around 250 sensors dispersed in a home automation environment can maintain adequate context awareness without causing a bottleneck, thanks to the short time interval between sending two sensors' outputs, figure3. The figure also illustrates how data annotations in semantic representation enable actions to be executed in near real-time, ensuring the system's responsiveness. The annotation process is significantly influenced by each sensor's signal transmission time differences; the time annotations decrease when there is a longer gap between two sendings. For example, a door/window sensor may not need to transmit a signal every second, as it only matters when the state changes (open/closed). Similarly, for human health monitoring, wearable sensors can transmit updates every 180-300 seconds, providing sufficient data for the application without overburdening the system.

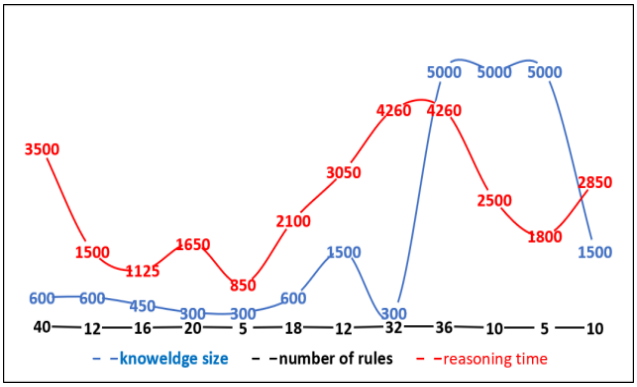
6 Discussion and Conclusion

6.1 Discussion

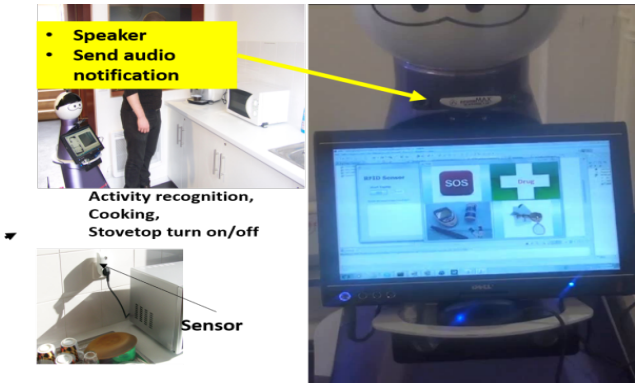
OWL, the de facto ontological language for defining concepts and reasoning about static relationships, is currently limited in its potential for building advanced inference engines due to its lack of variables. This is a significant limitation that needs to be addressed. OWL's strengths lie in classification and subclass reasoning, but its static nature hinders its ability to fully capture the complexity of real world phenomena, particularly those that change over time or depend on specific temporal conditions. Our approach is focused on managing, analysing, and understanding interconnected devices, sensors, and people in the Internet of Everything (IoE) ecosystem. We believe that narrative models are essential for understanding and processing the semantic, spatio-temporal context, enabling us to provide early warnings for potential risks, such as health deterioration or unsafe behaviors. Introducing temporal reasoning and event modeling in OWL often leads to redundancy in the ontology, making it more challenging to maintain, error-prone, and inefficient, as both the static and dynamic aspects must be kept. Modifying existing static ontologies to incorporate dynamic reasoning necessitates significant changes to the ontology structure. We underline that one of the key drawbacks of the approaches discussed earlier is the challenge of defining predicates of any arity to represent the temporal dimension of properties. This challenge underscores the need for further research in the field, as temporal reasoning often requires the representation of time-dependent relationships between entities, a task that is complex and difficult to achieve within the structure of a typical ontology or logic system. Combining OWL with SWRL never tackles the issues below. Indeed, SWRL raises several limitations. First, SWRL does not natively support negation and does not have built-in support for negation as a primitive feature. Therefore, we can not say David did not push the emergency button or express any negation statement. As an alternative approach, SWRL uses the DL safe rule, where negation is allowed on the variables in the head of the rule, which restricts negation when an application must exclude any fact. Moreover, in OWL 2 DL, The owl:NegativePropertyAssertion is used to represent a negative property assertion (i.e., state that there is no relation between two individuals), such as Robot Kompai does not assist David. Formerly, the owl:NegativePropertyAssertion relies on three RDF components: Subject, Predicate and Object. The subject concerned the individual involved in the relation; the predicate represents the negative relation, and the object represents the relation's target.

6.2 Limitations

While the proposed spatio-temporal ontology-based querying approach enhances context-aware reasoning and adaptive activity recognition, several limitations remain. First, the



(a) Average reasoning time combining hypothesis and transformation rules



(b) Robot and push button are used to detect and confirm early warning contexts

Figure 4: Assess the proposed framework's performance in real-time. Response time to recognizing inactivity or dangerous contexts and taking action.

current system has been evaluated in controlled or small-scale IoE environments; scalability to large, heterogeneous, and noisy deployments has not yet been demonstrated. Second, the ontology's conceptual scope is limited to predefined spatio-temporal and contextual dimensions, which may restrict adaptability to new domains or unforeseen events. Third, the reasoning process relies on deterministic temporal logic, which may be less effective in handling uncertainty, incomplete data, or contradictory sensor readings.

6.3 Future work

Scalability and real-time deployment – Evaluate the system's performance when deployed at large scale in heterogeneous IoE environments with high-frequency data streams, ensuring low-latency reasoning and response.

Integration of richer contextual dimensions – Extend the ontology to incorporate psychological, social, and environmental factors, allowing more nuanced activity recognition and risk prediction.

6.4 Conclusion

We present an ontological querying-processing approach that leverages narrative querying and event correlation analysis to monitor and ensure senior person safety in a smart environment. Ontology querying-processing allows gathering information from sensors (e.g., cameras, directly through voice interaction or robot's embedded tools) and a dynamic understanding of the environment. Relying on context-based questions ("Was there an unusual change in the human behaviour?"), the system can evaluate real-time actions or interruptions. Moreover, using a temporal ontology (HTemp) to perform a chronological and semantic analysis of events might track a sequence like cooking, taking a shower, moving to the bathroom, and an unexpected behaviour change. The system can then determine if an activity interruption (e.g., doesn't finish cooking) is abnormal and needs preventive actions. Lastly, contextual awareness has different safety implications according to location and activity in the person's environment (e.g., bathroom, kitchen). The system must know about the expected activities in each area (such as cooking in the kitchen); thanks to the hypothesis rules and transformation rules, the system should be able to determine if the activity is interrupted or if the person is not performing as expected.

References

[1] N. Carroll, *Beyond Aesthetics: Philosophical Essays*. Cambridge University Press, New York, 2001.

[2] D. Riboni and F. Murru, "Unsupervised Recognition of Multi-Resident Activities in Smart-Homes," *IEEE Access*, vol. 8, pp. 201985–201994, 2020. http://doi.org/10.1109/ACCESS.2020.3036226

[3] L. Sabri, S. Bouznad, S. Rama Fiorini, A. Chibani, E. Prestes and Y. Amirat, "An integrated semantic framework for designing context-aware internet of robotic things systems," *Integrated Computer-Aided Engineering*, vol. 25, no. 2, pp. 137–156, 2018. https://doi.org/10.3233/ICA-170559

[4] C. Khnaisser, V. Looten, L. Lavoie, A. Burgun and J. F. Ethier, "Building ontology-based temporal databases for data reuse: An applied example on hospital organizational structures," *Health Informatics Journal*, vol. 30, no. 2, 2024. http://doi.org/10.1177/14604582241259336

[5] M. Javeed, N. A. Mudawi, A. Alazeb, S. S. Alotaibi, N. A. Almujally and A. Jalal, "Deep Ontology-Based Human Locomotor Activity Recognition System via Multisensory Devices," *IEEE Access*, vol. 11,

pp. 105466–105478, 2023. http://doi.org/10.1109/ACCESS.2023.3317893

[6] S. F. Ghoreishi, W. D. Thomison and D. L. Allaire, "Sequential information-theoretic and reification-based approach for querying multi-information sources," *Aerosp. Inf. Syst.*, vol. 16, no. 12, pp. 575–587, 2019. https://doi.org/10.2514/1.I010753

[7] M. J. O'Connor and A. K. Das, "A Method for Representing and Querying Temporal Information in OWL," in *Biomedical Engineering Systems and Technologies*, vol. 127, Springer, Berlin, Heidelberg, 2010. https://doi.org/10.1007/978-3-642-18472-7_8

[8] X. Zhu, L. Bin, Y. Li, D. Zhaoyun and Z. Cheng, "TGR: Neural-symbolic ontological reasoner for domain-specific knowledge graphs," *Applied Intelligence*, vol. 53, no. 20, pp. 923946–23965, 2023. https://doi.org/10.1007/s10489-023-04834-8

[9] M. Larhrib, M. Escribano and J. J. Escribano, "An Ontological Behavioral Modeling Approach With SHACL, SPARQL, and RDF Applied to Smart Grids," *IEEE Access*, vol. 12, pp. 82041–82056, 2024. https://doi.org/10.1109/ACCESS.2024.3412656

[10] T. Ioannidis, G. Garbis, K. Kyzirakos, B. Konstantina and M. Koubarakis, "Evaluating Geospatial RDF Stores Using the Benchmark Geographica," *J. Data Semant.*, vol. 10, pp. 189–228, 2021. https://doi.org/10.1007/s13740-021-00118-x

[11] Z. Brahmia, F. Grandi and R. Bouaziz, "SQWRL: A TSQL2-Like Query Language for Temporal Ontologies Generated from JSON Big Data," *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 288–300, 2023. https://doi.org/10.26599/BDMA.2022.9020044

[12] S. Batsakis, I. Tachmazidis and G. Antoniou, "Representing Time and Space for the Semantic Web," *International Journal on Artificial Intelligence Tools*, vol. 26, no. 3, p. 1750015, 2017. https://doi.org/10.1142/S0218213017600156

[13] O. Fabrizio, G. Damien and D. O'Sullivan, "RDF Reification Benchmark using the Biomedical Knowledge Repository," *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*. https://doi.org/10.1109/ICSC50631.2021.00049

[14] G. P. Zarri, "NKRL, a knowledge representation tool for encoding the 'meaning' of complex narrative texts," *Natural Language Engineering*, vol. 3, no. 2, pp. 231–253, 1997. https://doi.org/10.1017/S135132499700179

[15] L. Sabri, "Internet of Robot Things in a Dynamic Environment: Narrative-Based Knowledge Representation and Reasoning," in *MobiQuitous 2021*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 419, Springer, Cham. https://doi.org/10.1007/978-3-030-94822-1_33

[16] T. M. Thanh, N. H. Thuy and N.-T. Huynh, "Key-value based data hiding method for NoSQL database," in *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, Ho Chi Minh City, Vietnam, pp. 193–197, 2018. https://doi.org/10.1109/KSE.2018.8573334

[17] F. Gargiulo, S. Silvestri and M. Ciampi, "Exploit Hierarchical Label Knowledge for Deep Learning," in *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, Cordoba, Spain, pp. 539–542, 2019. https://doi.org/10.1109/CBMS.2019.00110

[18] N. Yamada, K. Sakamoto, G. Kunito, Y. Isoda, K. Yamazaki and S. Tanaka, "Applying ontology and probabilistic model to human activity recognition from surrounding things," *IPSJ Digital Courier*, vol. 3, pp. 506–517, 2007. https://doi.org/10.2197/ipsjdc.3.506

[19] E. Aguado, V. Gomez, M. Hernando, C. Rossi and R. Sanz, "A Survey of Ontology-Enabled Processes for Dependable Robot Autonomy," *Frontiers in Robotics and AI*, vol. 10:11, 2024. https://doi.org/10.3389/frobt.2024.1377897

[20] A. Ben Sada, A. Naouri, A. Khelloufi, S. Dhelim and H. Ning, "A Context-Aware Edge Computing Framework for Smart Internet of Things," *Future Internet*, vol. 15, no. 5, p. 154, 2023. https://doi.org/10.3390/fi15050154

[21] R. Miao, Q. Jia, F. Sun, G. Chen, H. Huang and S. Miao, "Semantic Representation of Robot Manipulation with Knowledge Graph," *Entropy*, vol. 25, no. 4, p. 657, 2023. https://doi.org/10.3390/e25040657

[22] M. N. Finean, L. Petrović, W. Merkt, I. Marković and I. Havoutis, "Motion planning in dynamic environments using context-aware human trajectory prediction," *Computers, Environment and Urban Systems*, vol. 90, p. 101678, 2021. https://doi.org/10.1016/j.robot.2023.104450

[23] Z. Li, Z. Chen, Y. Li and C. Xu, "Context-aware trajectory prediction for autonomous driving in heterogeneous environments," *Mathematical Methods in Civil Engineering*, vol. 29, no. 3, p. e12989, 2023. https://doi.org/10.1111/mice.12989

[24] Y. Ge, S. Zhang, Y. Cai, T. Lu, H. Wang, X. Hui and S. Wang, "Ontology Based Autonomous Robot Task Processing Framework," *Frontiers in Neurorobotics*, vol. 18, 2024. https://doi.org/10.3389/fnbot.2024.1401075

[25] M. Adamik, R. Pernisch, I. Tiddi and S. Schlobach, "ORKA: An Ontology for Robotic Knowledge Acquisition," in *Knowledge Engineering and Knowledge Management. EKAW 2024*, Lecture Notes in Computer Science, vol. 15370, Springer, Cham. https://doi.org/10.1007/978-3-031-77792-9_19

[26] Y. Hu, C. Liu, M. Zhang, Y. Lu, Y. Jia and Y. Xu, "An ontology and rule-based method for human–robot collaborative disassembly planning in smart remanufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 89, p. 102766, Oct. 2024. https://doi.org/10.1016/j.rcim.2024.102766

[27] J. David, E. Coatanéa and A. Lobov, "Deploying OWL Ontologies for Semantic Mediation of Mixed-Reality Interactions for Human–Robot Collaborative Assembly," *Journal of Manufacturing Systems*, vol. 70, pp. 359–381, 2023. https://doi.org/10.1016/j.jmsy.2023.07.013

[28] M. A. Houghtaling, S. R. Fiorini, N. Fabiano, J. S. P. Gonçalves, O. Ulgen and T. Haidegger, "Standardizing an Ontology for Ethically Aligned Robotic and Autonomous Systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 3, pp. 1791–1804, Mar. 2024. https://doi.org/10.1109/TSMC.2023.3330981

[29] A. Koubaa, W. Boulila and A. Ammar, "Next–Generation Human–Robot Interaction with ChatGPT and Robot Operating System," *Software: Practice and Experience*, vol. 55, no. 9, pp. 1234–1245, 2024. https://doi.org/10.1002/spe.3377

[30] X. Wang, D. Veeramani, F. Dai and Z. Zhu, "Context-aware hand gesture interaction for human–robot collaboration in construction," *Comput. Aided Civ. Infrastructure Eng.*, vol. 39, no. 22, pp. 3355–3504, Apr. 2024. https://doi.org/10.1111/mice.13202

[31] R. Gayathri, V. Uma, R. Rajakumar, R. Vishnu Priya and R. Vedhapriyavadhana, "Spatio-Temporal and Semantic Enhanced Context-Aware Navigation for Indoor Robots," *IEEE Access*, vol. 13, pp. 36909–36929, 2025. https://doi.org/10.1109/ACCESS.2025.3545074

[32] S. Batsakis and E. G. M. Petrakis, "Representing temporal knowledge in the semantic web: The extended 4d fluents approach," *Combinations of Intelligent Methods and Applications*, vol. 8, pp. 55–69, 2011. https://doi.org/10.1007/978-3-642-19618-8_4

[33] T. Cui, W. Wei-Qi, R. S. Harold, S. Guergana and G. C. Christopher, "Cntro: A semantic web ontology for temporal relation inferencing in clinical narratives," *AMIA Annu Symp Proc*, vol. 13, pp. 787–791, 2010. https://pmc.ncbi.nlm.nih.gov/articles/PMC3041418/

[34] S. Benelhaj-Sghaier, A. Gillet and É. Leclercq, "Knowledge Graph Multilevel Abstraction: A Property Graph Reification Based Approach," in *Research Challenges in Information Science. RCIS 2024*, Lecture Notes in Business Information Processing, vol. 514, Springer, Cham. https://doi.org/10.1007/978-3-031-59468-7_2

[35] M. O'Connor and A. Das, "SQWRL: a query language for OWL," in *Proc. of the 6th International Conference on OWL: Experiences and Directions - Volume 529*, OWLED'09, pp. 208–215. CEUR-WS.org, Aachen, 2009. https://doi.org/10.5555/2890046.2890072

[36] A. Sheth and K. Thirunarayan, "Semantics Empowered Web 3.0: Managing Enterprise, Social, Sensor, and Cloud-based Data and Services for Advanced Applications," Morgan & Claypool Publishers, ACM Digital Library, 2012. https://doi.org/10.2200/s00433ed1v01y201207dtm031

[37] M. Jahn, "Narratology 2.3: A Guide to the Theory of Narrative," English Department, University of Cologne, 2021. http://www.uni-koeln.de/~ame02/pppn.pdf

[38] B. Meike, "Narratology: Introduction to the Theory of Narrative," University of Toronto Press, Toronto, 2009.

Maza Abdelouahab is a Ph.D. student at the University of Bordj Bou Arreridj, Algeria, specializing in communication and information technologies. He obtained his Master's degree in communication and information technologies from the University of Bordj Bou Arreridj. His current research interests include artificial intelligence and human activity recognition.

Sabri Lyazid is an Assistant Professor at the Bachir Ibrahimi University. He holds a Ph.D. in Computer Science, Images, Signals and Intelligent Systems from University Paris-Est Cr´eteil (UPEC). He is senior consultant on artificial intelligence, information systems architectures, and cybersecurity. While at UPEC, he worked on several collaborative research projects (e.g., SembySem, Web of Objects, Predykot, A2nets) dealing with IoT and ambient assisted living, robotics, artificial intelligence, and access management. His research interests are in ontology, knowledge representation and reasoning, robotics, artificial intelligence, deep learning, cognitive psychology, the Internet of Things paradigm, and blockchain technology.

Multi Stream Attention Networks with Adaptive Syntactic-Emotional Fusion for Sentiment Analysis

Nidhi Mishra*, Aakanshi Gupta*, Shuchi Mala*, Srishti Das*,
Naman Tyagi*, Sanjam Bhardwaj*, Narayan C Debnath†, Amit Mishra‡

*Department of Computer Science and Engineering, Amity School of Engineering Technology, AUUP, Noida, India.

†School of Computing and Information Technology Department of Software Engineering, Binh Duong Province, Vietnam.

‡Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida.

Abstract

Sentiment analysis is a challenging task in natural language processing due to the complexities between the syntactic and emotional structure of human content. This paper presents a dual-stream attention architecture that models both the syntactic representation and emotional intensity through separate attention mechanisms fused together via an adaptive mechanism. We evaluate our architecture on IMDb Movie Review Dataset, demonstrating its effectiveness compared to various lexicon-based solutions such as TextBlob, SentiWordNet and Vader. Additionally, we also examine Machine Learning based approaches with TF-IDF and Bag-of-Words for feature extraction, as well as sequential models like LSTM and BiLSTM combined with embeddings (GloVe). Furthermore, we examine methods to improve classification performance, including stacking and voting techniques. The performance of each solution is examined across various metrics, including accuracy, precision, recall, F1 Score and AUC-ROC. Our results highlight the importance of explicitly modeling both syntactic and emotional representations for sentiment analysis.

Key Words: sentiment, emotion mining, natural language processing, attention.

1 Introduction

Sentiment Analysis refers to the task of determining the emotional tone and opinions expressed in a text and is fundamental to various applications including social media monitoring, customer feedback analysis, and market research. In recent years, sentiment analysis has emerged as one of the important subdomains of NLP (Natural Language Processing) driven by the proliferation of online reviews, social media, and other user-generated content. Movie reviews are a valuable source of public opinion as they provide insights into viewer preferences, emotions, and overall satisfaction with films. Despite significant advances in natural language processing, accurately capturing sentiments remains challenging due to the ways in which humans express emotions through both words and grammatical structures.

Current approaches to sentiment analysis majorly rely on machine learning techniques and deep learning models, which, while successful, face challenges integrating various aspects such as syntactical structure and emotional nuances. Early models like lexicon-based sentiment analysis [13, 30] and machine learning methods focused on keyword matching or feature extraction. While these models provide valuable insights, they often failed to capture the deeper, context-dependent meaning inside a text. More recent approaches, such as Recurrent Neural Networks (RNNs) [17], and long short-term memory (LSTM, BiLSTM) [26, 25] have made strides in handling sequential data. Transformer [32] based models like Bidirectional Encoded Representations from Transformers (BERT) [7] have further pushed boundaries, demonstrating state-of-the-art performance across many NLP tasks. However, these models still struggle to integrate the syntactic and emotional dimensions that are crucial to sentiment in complex scenarios.

To address these challenges, we propose a hybrid architecture, the Multi Stream Attention with Adaptive Syntactic-Emotional Fusion, which combined two different attention streams: one focused on syntactic information and other on emotional representation. By utilizing both syntactic and emotional information, our model aims to enhance the representation of sentiment and overall increase the overall accuracy of sentiment analysis. The dual stream attention mechanism uses dependency guided attention to align syntactic structures, while an emotion enhanced representation integrates external lexicon and emotion embedding to capture the emotional tone of the text.

In this research, we developed a Hybrid Sentiment Analysis model that integrates syntactic structure with emotional semantics through a novel dual-attention mechanism. Our

approach combines dependency parsing with emotional lexicon information to create structure-aware and emotion-enhanced attention pathways. The model employs spaCy for dependency parsing and aligns these parse trees with BERT's tokenization scheme to generate syntactic attention masks. Simultaneously, we incorporate sentiment lexicon scores to create emotion intensity embeddings that guide a parallel attention mechanism. These dual pathways are adaptively combined using a trainable alpha parameter that automatically determines the optimal balance between syntactic and emotional information for each input. Implementation in PyTorch with BERT-base as the encoder (8 attention heads, 768 hidden dimensions) demonstrated significant improvements over baseline models, with particularly strong performance on texts where sentiment is expressed through complex syntactic structures or emotionally charged language.

The primary contributions of this paper include development of an adaptive fusion mechanism that combines the two attention streams based on their importance, an approach to token alignment through dependency parsing and the integration of emotion enhanced representations. This paper is organized as follows: Section 2 provides an overview of related work in sentiment analysis, focusing on traditional methods, deep learning-based techniques. Section 3 introduces the architecture in detail, including the dual stream attention mechanism and adaptive fusion method. The experiment setup and evaluation are discussed in Section 4 followed, followed by a comprehensive analysis of results in Section 5. Finally, Section 6 concludes the paper and outlines future work.

2 Related Work

A lexicon-based method is one of the earlier procedures where sentiment analysis is performed by using pre-assigned sentiment lexicons that articulate polarity scores for words in the text. For example, Kiritchenko et al [13] introduced lexicons specifically designed for the analysis of social media texts such as tweets. This approach offers a simple yet effective mechanism, requiring no training data, and it relies on the explicit sentiment values contained in the lexicons. The main advantage of this approach is its ease of implementation and its applicability in situations where annotated training data is scarce. However, a notable limitation is that these lexicons often fail to capture contextual nuances and irony, resulting in misinterpretation when the text is taken out of context. This drawback is especially apparent in texts that incorporate sarcasm or subtle emotional cues that are not represented in the static lexicon. Moreover, while lexicon-based models work well for straightforward cases, they struggle with the polysemy and ambiguity inherent in natural language.

Traditional machine learning models such as Naïve-Bayes [11], Support Vector Machines [9], and Random Forest [3] have been widely employed for sentiment classification tasks. These models typically work by extracting a range of features from textual data such as n-grams, part-of-speech tags, and syntactic structures to predict sentiment labels. One of the key strengths of these approaches is their ability to achieve high accuracy, particularly in controlled environments where feature engineering is optimized. However, a major disadvantage is that their performance is highly sensitive to the choice of feature selection and pre-processing techniques. The reliance on manual feature engineering makes these methods less adaptable to the subtle nuances of language and context found in diverse datasets. Despite this, machine learning techniques remain popular due to their efficiency and the relative simplicity of their implementation in comparison to more complex deep learning methods.

The introduction of word embeddings has revolutionized the way textual data is represented for sentiment analysis. Techniques such as word2vec [19] and GloVe [23] transform words into high-dimensional vectors that encapsulate both semantic and syntactic information. These embeddings enable models to capture relationships between words and understand the context in which they appear, thereby providing a richer and more nuanced representation of the text. Despite their success, word embeddings are generally trained on large, general-purpose corpora and may not perform optimally when applied directly to domain-specific sentiment analysis tasks without fine-tuning. This limitation highlights the importance of adapting these embeddings to the particularities of the task at hand, ensuring that they capture the unique characteristics and subtleties present in specialized datasets.

Deep learning has introduced a new paradigm in sentiment analysis through models that can automatically learn hierarchical representations of text. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have proven effective in handling sequential data, as they are capable of modeling dependencies over long sequences. For example, Rao et al [26] demonstrated that LSTM networks outperform traditional classifiers by capturing long-range dependencies in text data. More recently, transformer-based models, such as BERT, have further advanced the field by employing self-attention mechanisms that more accurately capture contextual relationships. Although these models represent a significant leap forward in performance, they also require substantial computational resources and large volumes of training data. As such, while deep learning approaches have set new benchmarks in sentiment analysis, their complexity and resource demands can be a barrier to their widespread adoption in all contexts.

Hybrid models aim to capitalize on the strengths of both traditional machine learning techniques and deep learning approaches. For instance, dos Santos and Gatti [8] introduced a hybrid model that integrates Convolutional Neural Networks (CNNs) with character-to-word embeddings for sentiment classification. This approach leverages the robust feature extraction capabilities of traditional methods and the powerful representation learning of deep learning models. By combining these methodologies, hybrid approaches are able to create a more robust framework for sentiment analysis that mitigates the weaknesses of each individual method. Such models are

Table 1: Comparison of Related Studies in Sentiment Analysis

Study	Approach	Key Technique	Dataset	Performance
Pang et al. (2002)	Classical ML	SVM, Naive Bayes (bag-of-words)	Movie reviews	82.9% (Acc.)
Taboada et al. (2011)	Lexicon-based	SO-CAL lexicon (with intensifiers)	Various domains	78.7% (Acc.)
Socher et al. (2013)	DL (Recursive)	Recursive Neural Tensor Network (RNTN)	Stanford Sentiment Treebank	85.4% (Acc.)
Mohammad et al. (2013)	Lexicon-based	NRC Emotion Lexicon (features for SVM)	SemEval-2013 Task 2 (Twitter)	69.0% (F1)
Tang et al. (2016)	DL (LSTM)	Target-dependent LSTM	Twitter (target-dependent)	71.5% (Acc.)
Howard & Ruder (2018)	DL (Transfer)	ULMFiT (fine-tuned AWD-LSTM)	IMDb movie reviews	95.4% (Acc.)
Peters et al. (2018)	DL (Contextual)	ELMo embeddings	SST-5 (Stanford Sentiment Treebank)	54.7% (Acc.)
Devlin et al. (2019)	DL (Transfer)	BERT pre-training	GLUE benchmark (e.g., SST-2)	94.9% (Acc.)
Yang et al. (2019)	DL (Transfer)	XLNet (Transformer)	IMDb movie reviews	96.2% (Acc.)
Liu et al. (2019)	DL (Transfer)	Multi-task Deep NN (MT-DNN)	GLUE benchmark (e.g., SST-2)	95.6% (Acc.)

particularly effective in capturing both the overt sentiment cues and the underlying contextual subtleties present in complex textual data.

3 Methodology

3.1 Input Representation

Our architecture enhances traditional transformer-based models by introducing a dual-stream attention mechanism that processes both syntactic and emotional features simultaneously. The model takes as input a sequence of tokens. The input sequence consists of tokens arranged in order, where each token is an element of the sequence, and the total number of tokens corresponds to the sequence length.

These tokens are then processed by a pre-trained BERT encoder to obtain contextual representations. The BERT encoder transforms the input token sequence into a matrix of contextual embeddings that capture the semantic and syntactic context of each token.

$$\mathbf{H} = \text{BERT}(\mathbf{X}) \tag{1}$$

where $\mathbf{X}$ is the tokenized input sequence (in embedding form) and $\mathbf{H}$ is the resulting matrix of contextual embeddings capturing the semantic and syntactic context of each token.

3.2 Syntactic Attention Stream

The syntactic stream leverages dependency parsing to capture grammatical relationships between tokens. For each input sequence, a dependency tree is constructed using a syntactic parser. This tree is then converted into an adjacency matrix where entries indicate whether tokens are syntactically related or not. This matrix guides the syntactic attention mechanism, which extends the standard multi-head attention by incorporating dependency information. Attention scores are computed taking into account the syntactic relationships between tokens, allowing the model to focus on grammatically relevant connections.

The syntactic attention output is obtained by multiplying the attention matrix with the value matrix for the syntactic stream (Derived from equation 2). The query, key, and value matrices for the syntactic stream are computed by applying learnable weight matrices to the contextual embeddings produced by the BERT encoder. Equation 3 describes the mathematical formulation for the operation.

$$M_{\text{syntactic}}[i,j] = \begin{cases} 0, & \text{if tokens } i \text{ and } j \text{ are related,} \\ -\infty, & \text{otherwise.} \end{cases} \tag{2}$$

$$A_s = \text{softmax}\left(\frac{Q_s K_s^\top}{\sqrt{d_{\text{head}}}} + M_{\text{syntactic}}\right) \tag{3}$$

3.3 Emotional Attention Stream

The overall sentiment analysis process can be summarized in four phases. First, the raw text sequence is tokenized using the BERT tokenizer with subword alignment, and contextual embeddings are generated using the BERT encoder. Second,

the dual attention streams operate: the syntactic stream builds the dependency adjacency matrix based on parsing, computes attention scores considering syntactic relationships, while the emotional stream retrieves emotional intensity scores from a sentiment lexicon and adjusts attention weights accordingly. Third, the outputs of both syntactic and emotional streams are combined adaptively using a weighted fusion mechanism. Fourth, the fused features are pooled using mean aggregation and passed through a classifier layer to predict the sentiment class.

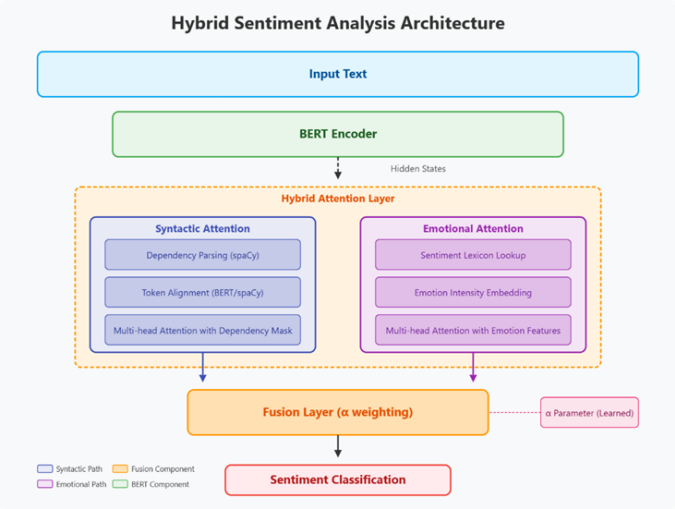


Figure 1: Proposed architecture

The adaptive fusion mechanism described in equation 4 combines the outputs of the syntactic and emotional attention streams using a weighted sum controlled by a learnable parameter. This parameter balances the contribution of each stream by applying a sigmoid function, resulting in a final integrated output that effectively captures both grammatical structure and emotional context for improved sentiment analysis performance.

$$O_{\text{fused}} = \sigma(\alpha) \cdot O_s + \big(1 - \sigma(\alpha)\big) \cdot O_e \tag{4}$$

where O_{fused} is the final attention output, O_s is the output from the syntactic attention stream, and O_e is the output from the emotional attention stream.

An overview of our pipeline is described by Algorithm 1.

Algorithm 1 Sentiment Analysis with Syntactic-Emotional Fusion

Require: Raw text sequence $A = \{a_1, a_2, \ldots, a_n\}$
Ensure: Predicted sentiment class y

Phase 1: Input Processing
Tokenize A using BERT tokenizer and align subwords
Generate contextual embeddings H using BERT (see Fig. 1)
Phase 2: Dual Attention Streams
Dependency-Guided Syntactic Stream:
Parse dependencies using spaCy; build adjacency matrix $M_{\text{syntactic}}$
Compute attention scores for syntactically related tokens
Emotion-Aware Attention Stream:
Retrieve emotional intensity scores from AFINN lexicon
Adjust attention weights for sentiment-bearing tokens
Phase 3: Feature Fusion
Combine syntactic and emotional outputs adaptively
Phase 4: Classification
Pool features using mean aggregation
Predict sentiment class via classifier layer

4 Experimental Setup

This study utilized the IMDb movie review dataset [19] consisting of 50,000 reviews evenly distributed between positive and negative sentiments. With an 80:20 ratio, the dataset was divided into training and testing sets, ensuring there were an equal number of positive and negative reviews in both sets. Figure 2 shows the distribution of length and frequency of words.

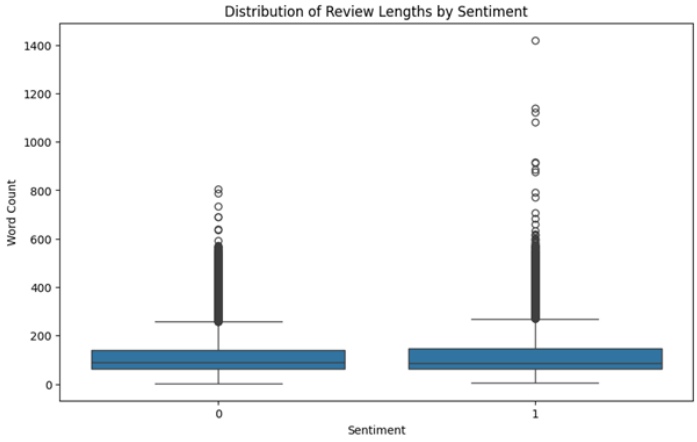


Figure 2: Distribution of sentence length

Given the unstructured nature of text data in movie reviews, preprocessing plays a crucial role in preparing the dataset for analysis. The key preprocessing steps applied include tokenization, which involves splitting the text into individual words while preserving sentence boundaries; lowercasing, which converts all text to lowercase for consistency; and stopword removal, which eliminates common words like “the,”

"is," and "in" that do not carry sentiment. Additionally, lemmatization is used to convert words into their base forms, and cleaning procedures are employed to remove special characters, punctuation, and normalize repeated characters, thereby refining the text for more accurate analysis.

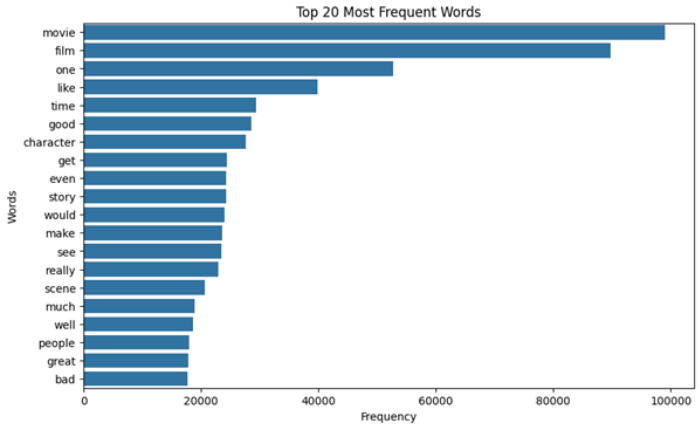


Figure 3: Distributions of frequency of words

The English model from spaCy is used to perform dependency parsing and emotional scores are derived from the AFINN sentiment lexicon (normalized to the range [-1,1]). During training, all parameters are optimized using the AdamW optimizer with a learning rate of 2e-5 and linear warmup. The hybrid architecture combines structural and affective information in text which enables more comprehensive sentiment analysis.

Three lexicon-based approaches were evaluated, each utilizing different dictionaries and techniques for sentiment scoring:

- **TextBlob**: Provides sentiment analysis through a simple lexicon-based approach by assigning polarity scores from negative one to positive one. It considers word polarity while accounting for basic grammatical structures.
- **VADER**: Optimized for social media and informal text, it incorporates rules for punctuation, capitalization, and negation. It uses a comprehensive sentiment lexicon with empirically evaluated sentiment intensity scores.
- **SentiWordNet**: Extends WordNet by assigning three sentiment scores—positive, negative, and neutral—to each sentence.

Traditional machine learning models were implemented using both Bag-of-Words and TF-IDF features for text representation:

- **Naïve Bayes**: This probabilistic classifier assumes feature independence and calculates class probability using Bayes' theorem.
- **Logistic Regression**: It models the probability of positive sentiment using a sigmoid function applied to a linear combination of features.
- **SVM**: Finds an optimal hyperplane to separate sentiment classes in the feature space, maximizing the margin between classes.
- **Random Forest**: This ensemble method combines multiple decision trees to reduce overfitting and improve generalization.

Advanced neural network architectures were implemented to capture complex linguistic patterns:

- **LSTM**: Handles sequential data by maintaining a cell state that captures long-term dependencies.
- **BERT**: Revolutionizes text processing by using bidirectional context and self-attention mechanisms.
- **BiLSTM**: A specialized version of LSTM that processes input sentences in both directions simultaneously.

To assess the sentiment analysis models' performance on movie reviews, we employ six key metrics:

- Accuracy
- Precision
- Recall
- F1 Score
- AUC-ROC
- Confusion Matrix

5 Results

The models are compared based on key metrics discussed earlier. The confusion matrices for each model are demonstrated in the figures. The comparison of accuracy, precision, and recall across different models is summarized below.

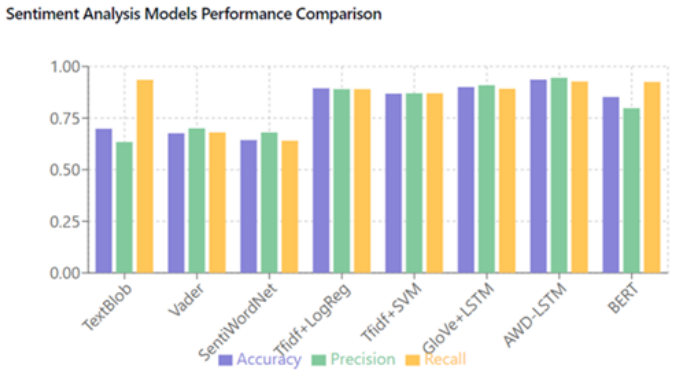


Figure 4: Comparison of Accuracy, Precision, and Recall for Various Sentiment Analysis Models: TextBlob, Vader, SentiWordNet, TfIdf+LogReg, TfIdf+SVM, GloVe+LSTM, AWD-LSTM, and BERT

Table 2 provides a summary of the evaluation metrics for all the models tested, illustrating the quantitative performance of each approach.

Table 2: Evaluation metrics for models tested

Model	Accuracy	Precision	Recall	F1 Score	AUC ROC
TextBlob	0.698	0.634	0.935	0.756	0.698
Vader	0.676	0.700	0.680	0.727	0.676
SentiWordNet	0.643	0.680	0.640	0.620	0.638
Tfid + Logistic Regression	0.894	0.890	0.890	0.890	0.961
Tfid + SVM	0.868	0.870	0.870	0.870	0.947
GloVe + LSTM	0.900	0.909	0.892	0.900	0.967
BERT	0.852	0.798	0.925	0.856	0.855
Proposed model	0.913	0.912	0.916	0.912	0.9632

Figure 5: Confusion Matrices for Sentiment Analysis Models: TextBlob, VADER, Stanford NLP, and BERT

Lexicon-based approaches like TextBlob, Vader, and SentiWordNet are characterized by their simplicity and ease of implementation. They rely on predefined dictionaries to evaluate sentiment, making them computationally efficient and interpretable. However, their reliance on fixed lexicons presents significant limitations:

- **Accuracy:** Lexicon-based models often yield moderate accuracy. For example, TextBlob achieved an accuracy of 69.8%, while Vader reached 67.57%.
- **Strengths:** They are straightforward to implement, requiring minimal computational resources and providing quick insights into sentiment.
- **Weaknesses:** These models struggle with the nuances of language, particularly in detecting sarcasm, idiomatic expressions, and mixed sentiments. For instance, in a review stating, "The acting was excellent, but the plot was terrible," lexicon models may misclassify the sentiment as positive, failing to recognize the contrasting sentiments expressed.

Machine learning models utilize TF-IDF and Bag-Of-Words for feature extraction. These techniques demonstrate significant improvements in sentiment classification:

- **Accuracy:** Models like Tfidf + Logistic Regression achieved an accuracy of 89.4%, significantly outperforming lexicon-based models.
- **Strengths:** These models can learn patterns from the data, allowing them to capture subtle distinctions between positive and negative sentiments. The effective weighting of words through techniques like TF-IDF contributes to better sentiment discrimination.
- **Weaknesses:** While machine learning models manage simpler linguistic patterns well, they may still struggle with the broader context or dependencies within longer sentences, limiting their effectiveness in nuanced sentiment analysis.

Deep learning approaches, particularly those utilizing LSTM and BiLSTM architectures, significantly outperformed both lexicon-based and traditional machine learning models:

- **Accuracy:** The GloVe + LSTM model achieved 90.02%, while AWD-LSTM reached 93.64%.
- **Strengths:** These models excel in capturing long-range dependencies, allowing them to understand the context surrounding individual words and phrases. Their ability to leverage pre-trained word embeddings enhances semantic understanding, making them adept at handling complex sentiment structures.
- **Weaknesses:** While deep learning models are powerful, they require significant computational resources and extensive datasets for training, which may limit their accessibility for some applications.

Our proposed model demonstrates robust performance across all evaluation metrics, achieving an accuracy of 91.3% and an AUC-ROC score of 0.9632. The model's confusion matrix

reveals strong classification capabilities, with 4,834 correct negative predictions and 4,296 correct positive predictions out of 10,000 samples.

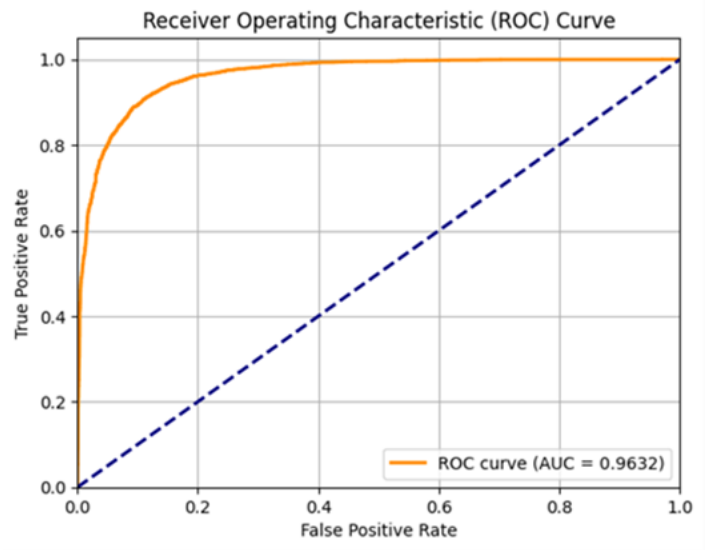


Figure 6: AUC ROC Curve for our model

The model exhibits balanced performance across sentiment classes, with class-specific metrics highlighting its effectiveness. For negative sentiment, it achieves a precision of 0.9593 and recall of 0.8791, indicating high reliability in negative sentiment identification. For positive sentiment, the model shows strong recall (0.9545) with good precision (0.8660), demonstrating effective positive sentiment detection.

Validation Set			
TARGET / OUTPUT	Positive	Negative	SUM
Positive	4296 42.96%	665 6.65%	4961 86.60% 13.40%
Negative	205 2.05%	4834 48.34%	5039 95.93% 4.07%
SUM	4501 95.45% 4.55%	5499 87.91% 12.09%	9130 / 10000 91.30% 8.70%

Figure 7: Confusion matrix for our model

The balanced F1-scores (0.9174 for negative and 0.9081 for positive) indicate consistent performance across classes, suggesting that our hybrid architecture effectively combines syntactic and emotional features. This balanced performance, coupled with the high AUC-ROC score, validates our approach of integrating dependency-guided attention with emotion-aware processing for sentiment analysis.

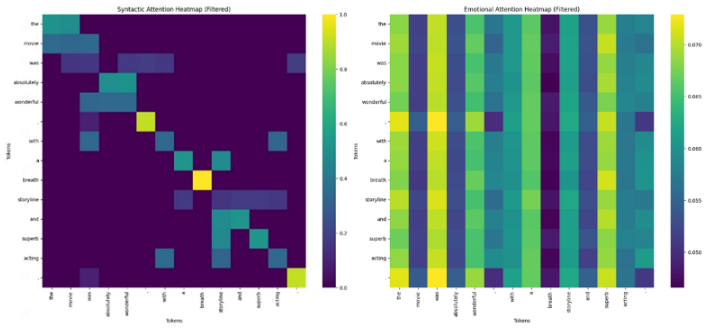


Figure 8: Syntactic (left) and emotional (right) attention heatmaps for the input sentence

The model's error patterns, particularly the lower false negative rate (205 cases) compared to false positives (665 cases), suggest that while the model is more conservative in assigning negative sentiment, it maintains high overall classification accuracy. These results demonstrate that our hybrid approach successfully captures both structural and affective aspects of sentiment, leading to robust and reliable classification performance.

Several potential threats to validity must be considered:

- **Dataset Bias:** The IMDb dataset primarily consists of movie reviews, which may not generalize well to other domains such as financial news or social media sentiment analysis.
- **Preprocessing Impact:** The choice of tokenization, stopword removal, and lemmatization could influence model performance. Alternative preprocessing techniques might yield different results.
- **Hyperparameter Sensitivity:** Deep learning models, including our proposed architecture, rely on numerous hyperparameters. Small variations in learning rates, batch sizes, or attention mechanisms may impact final performance.
- **Model Interpretability:** Despite improved interpretability via attention mechanisms, neural networks remain complex and somewhat opaque. Further analysis is needed to improve explainability.
- **Real-world Deployment:** The evaluation was conducted in a controlled environment. Performance in real-time applications with noisy or adversarial data remains an open challenge.

6 Conclusions

This paper provides a review of sentiment analysis techniques applied to IMDb reviews, with a focus on our proposed architecture combining syntactic and emotional components. The comparative analysis across different methodologies reveals clear differences in the performance and capabilities of methods. Our experimental results show that lexicon-based approaches, while interpretable and computationally efficient, face big challenges in handling the nuanced complexity of movie reviews. Machine learning models using TF-IDF and BoW for feature extraction showed improvements. They achieved accuracies above 85% but still struggled with maintaining contextual understanding in larger sentences.

Deep learning architectures, particularly our architecture, demonstrated better performance in maintaining long-range dependencies and relationships. By using syntactic dependencies with emotional features via a dual stream attention mechanism, our model achieved accuracy of 89.73% and an AUC ROC score of 0.9632. These results validate our design choices and show the effectiveness of combining syntactic and emotional features.

The current investigation is constrained by its consideration of only one domain, i.e., IMDb movie reviews, and such narrowing might limit the applicability of the findings to other text types or use cases. The use of English-language corpora also forecloses direct application to multilingual or code-switching scenarios without further modification. The suggested architecture relies on the correctness of external tools like dependency parsers and sentiment lexicons and is thus vulnerable to parser errors and lexicon coverage issues. Additionally, the use of BERT-based encoding, though effective, brings with it significant computational and memory demands that can prevent deployment in environments with less resources. Lastly, while attention visualizations provide some interpretability, the decision process itself remains opaque, hindering full transparency of model reasoning.

Future work includes expanding this effort to utilize the suggested dual-stream attention architecture in varied domains like financial news, healthcare stories, and social media posts to test domain generalization. Code-switched and multilingual sentiment analysis will be investigated by integrating with models like mBERT or XLM-R and language-specific syntactic parsers and emotion lexicons. The approach can be further tested with newer transformer variants (e.g., RoBERTa, DeBERTa) and augmented emotion representations obtained from context-sensitive embeddings or knowledge graphs. Real-time sentiment analysis applications such as streaming data will be explored with low latency optimized inference. Apart from that, future work will target better model interpretability through attention visualization and explanation techniques, robustness against noisy and adversarial inputs, and multimodal sentiment analysis extension to text, visual, and audio modalities.

References

[1] Wajdi Aljedaani, Furqan Rustam, Mohamed Wiem Mkaouer, Abdullatif Ghallab, Vaibhav Rupapara, Patrick Bernard Washington, Ernesto Lee, and Imran Ashraf. Sentiment analysis on twitter data integrating textblob and deep learning models: The case of us airline industry. *Knowledge-Based Systems*, 255:109780, 2022.

[2] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[3] Bahrawi Bahrawi. Sentiment analysis using random forest algorithm-online social media based. *Journal of Information Technology and Its Utilization*, 2:29, 12 2019.

[4] Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. Projecting embeddings for domain adaptation: Joint modeling of sentiment analysis in diverse domains, 2018.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[6] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent attention network on memory for aspect sentiment analysis. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[8] Cícero dos Santos and Maíra Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In Junichi Tsujii and Jan Hajic, editors, *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[9] Teng-Kai Fan and Chia-Hui Chang. Blogger-centric contextual advertising. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 1803–1806, New York, NY, USA, 2009. Association for Computing Machinery.

[10] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

[11] Hanhoon Kang, {Seong Joon} Yoo, and Dongil Han. Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, 39(5):6000–6010, April 2012. Funding Information: This work was supported by the Korea Research Foundation (KRF) grant funded by the Korea government (MEST) (No. 2010-0015842).

[12] Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6975–6988, Online, November 2020. Association for Computational Linguistics.

[13] S. Kiritchenko, X. Zhu, and S. M. Mohammad. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762, Aug 2014.

[14] Qizhi Li, Xianyong Li, Yajun Du, Yongquan Fan, and Xiaoliang Chen. A new sentiment-enhanced word embedding method for sentiment analysis. *Applied Sciences*, 12(20), 2022.

[15] Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. Exploiting bert for end-to-end aspect-based sentiment analysis. pages 34–41, 01 2019.

[16] Ning Liu, Bo Shen, Zhenjiang Zhang, Zhiyuan Zhang, and Kun Mi. Attention-based sentiment reasoner for aspect-based sentiment analysis. *Human-centric Computing and Information Sciences*, 9(1):35, 2019.

[17] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning, 2016.

[18] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy, July 2019. Association for Computational Linguistics.

[19] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[21] Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In Suresh Manandhar and Deniz Yuret, editors, *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 321–327, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[22] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics, July 2002.

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[24] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[25] Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany, August 2016. Association for Computational Linguistics.

[26] Guozheng Rao, Weihang Huang, Zhiyong Feng, and Qiong Cong. Lstm with sentence representations for

document-level sentiment classification. *Neurocomputing*, 308:49–57, 2018.

[27] Elena Rudkowsky, Martin Haselmayer, Matthias Wastian, Marcelo Jenny, Štefan Emrich, and Michael Sedlmair. More than bags of words: Sentiment analysis with word embeddings. *Communication Methods and Measures*, 12(2-3):140–157, 2018.

[28] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[29] Chi Sun, Luyao Huang, and Xipeng Qiu. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[30] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307, June 2011.

[31] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[33] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[34] Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xuejie Zhang. Refining word embeddings for sentiment analysis. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 534–539, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Authors

Dr. Nidhi Mishra is working as an Assistant Professor in the Department of Computer Science and Engineering at Amity University, Noida, Uttar Pradesh, India. She received her Ph.D. degree from Banasthali University, Jaipur, India in 2016. She has published more than 55 articles in peer-reviewed international journals and conferences. She is a member of IEEE and ACM society. She has guided 3 Ph.D. and 15 M.Tech research scholars in the area of Artificial Intelligence, Machine Learning, Deep Learning, Neural Networks, and NLP. Her research interests are Artificial Intelligence, Information Retrieval, Data Science, and Natural Language Processing.

Dr. Aakanshi Gupta is working is as an Assistant Professor in the department of Computer Science and Engineering at Amity University, Noida, Uttar Pradesh, India. She has an experience of more than 15 years in academics and research She received her Ph.D. degree from GGSIPU, Delhi; under the guidance Prof. Bharti Suri. Her research interests are Artificial Intelligence, Information retrieval, Data Science and Software Reliability.

Dr. Shuchi Mala is working as an Assistant Professor in the department of Computer Science and Engineering at Amity University, Noida, Uttar Pradesh, India. She has an experience of more than 6 years in academics and 4 years in full-time research She received her Ph.D. degree from Malaviya National Institute of Technology Jaipur (MNIT), India in 2018. She has published more than 32 articles in peer reviewed journals, international conferences, and book chapters. She is a member of IET and ACM society. Her research interests are spatial data modelling, health informatics and intelligent systems. She actively serves as a reviewer for esteemed international journals.

Srishti Das is currently pursuing a B.Tech in Computer Science and Engineering from Amity University, Noida, Uttar Pradesh, India. Her interests include deep learning, computer vision, and artificial intelligence. She is enthusiastic about research in emerging AI technologies and aims to contribute to the field through innovative applications.

Naman Tyagi is currently pursuing a B.Tech in Computer Science and Engineering from Amity University, Noida, India. His research interests include deep learning, model optimization, and artificial intelligence. He is passionate about exploring efficient and practical applications of AI techniques.

Sanjam Bhardwaj is pursuing a B.Tech in Computer Science and Engineering at Amity University, Noida. His research interests include Artificial Intelligence, Machine Learning, and Cybersecurity.

Dr. Narayan C Debnath is currently the Founding Dean of the School of Computing and Information Technology at Eastern International University, Vietnam. He is also serving as the Head of the Department of Software Engineering at Eastern International University, Vietnam. Formerly, Dr. Debnath served as a Full Professor of Computer Science at Winona State University, Minnesota, USA for 28 years, and the elected Chairperson of the Computer Science Department at Winona State University for 7 years. Dr. Debnath has been the Director of the International Society for Computers and their Applications (ISCA), USA since 2014. Professor Debnath made significant contributions in teaching, research, and services across the academic and professional communities. He has made original research contributions in Software Engineering, Artificial Intelligence and applications, and Information Science, Technology and Engineering. He is an author or co-author of over 500 research paper publications in numerous refereed journals and conference proceedings in Computer Science, Information Science, Information Technology, System Sciences, Mathematics, and Electrical Engineering. He is also an author of over 15 books published by well-known international publishers including Elsevier, CRC, Wiley, Bentham Science, River Publishing, and Springer. Dr. Debnath has made numerous teaching, research and invited keynote presentations at various international conferences, industries, and teaching and research institutions in Africa, Asia, Australia, Europe, North America, and South America. He has been a visiting professor at universities in Argentina, China, India, Sudan, and Taiwan. He has been maintaining an active research and professional collaborations with many universities, faculty, scholars, professionals and practitioners across the globe. Dr. Debnath is an active member of the IEEE, IEEE Computer Society, and a Senior Member of the International Society for Computers and their Applications (ISCA), USA.

Amit Mishra has eleven years of teaching experience and research experience. He has done PhD from NIT Kurukshetra, Haryana in the area of Natural Language Processing. He has worked on a couple of research projects such as developing a Semantic Question Answering System, and an Intention Mining of potential buyers on E-Commerce Websites. His Area of interest includes Data Science, Natural Language Processing, and Machine Learning. He has published papers in various reputed International conferences and Journals. He has technical expertise in C, Java (J2SE J2EE), Android programming, MySQL, IBM DB2, ORACLE, and MATLAB

Deep Learning Approach for Anxiety and Stress Detection through Facial Emotion Analysis

Houcem REZAIGUIA*
University of Biskra, Biskra, ALgeria.
Abdelhamid DJEFFAL†
University of Biskra, Biskra, Algeria.

Abstract

Facial expressions are crucial in everyday emotional communication since they are an indicator of sentiments and allow a person to convey his emotional state. People can instantly understand another person's emotional condition based on their facial expressions. As a result, information on facial expressions is frequently employed in automatic emotion recognition systems. Anxiety is one of the most prevalent emotions that people experience in various situations. As a result, it must be identified and treated. In this study, we propose a novel framework, named ANet, that utilises the AlexNet Convolutional Neural Networks (CNN) model for anxiety and stress detection through the analysis of facial emotions. The proposed system follows a step-by-step process, starting with obtaining images from a public database and employing the Viola-Jones algorithm for face detection. Subsequently, the detected face region is extracted as the Region of Interest (ROI). During the training phase, the images are trained as feature maps using the AlexNet CNN model. Once the model is trained, input videos undergo frame extraction, face detection, and ROI extraction. The extracted ROI is passed to the classifier for emotion identification, including anger, sadness, happiness, disgust, fear, and neutral. By analysing the combination of detected facial emotions, anxiety levels are determined and categorised as high anxiety, moderate anxiety, mild anxiety, and no anxiety. The proposed ANet framework demonstrates promising results in anxiety detection, providing a reliable and efficient method for early identification and assessment of anxiety based on facial emotions.

Key Words: anxiety, CNN, AlexNet, facial emotions, face detection.

1 Introduction

Understanding human emotion is important for enhancing normal utility and preventing undesirable effects. Affective computing systems that could detect and interpret human non-verbal behaviours, like facial expressions, speech prosody, body movements, and skin conductance, are necessary [13]. The presence of multiple stressors can elicit diverse affective states and physiological responses in individuals. The diagnosis and treatment of stress-related disorders, along with anxiety, have garnered increasing attention from researchers and clinicians. Anxiety detection plays a pivotal role in preventive mental health interventions [9]. Anxiety is widespread, adversely affecting daily functioning, leading to significant suffering, and resulting in substantial healthcare expenses as well as costs linked to decreased productivity [19]. It adversely impacts an individual's health, induces negative feelings, and, in severe cases, may lead to mental health disorders. Statistics on anxiety gathered by Single Care show that the main causes of stress are mostly the same around the world, including money problems, work pressures, and family responsibilities. A novel stressor has arisen since 2020: the COVID-19 pandemic, a divisive political environment, and supplementary factors [8]. Although clinical guidelines are available [16], the handling of these conditions in primary care frequently falls short of optimal standards. These conditions can have a profound impact on an individual's emotional well-being, daily functioning, and overall quality of life. Early detection and intervention play a crucial role in effectively managing and treating anxiety and depression, preventing further deterioration and promoting better mental health outcomes. Conventionally, anxiety and depression diagnoses have mainly depended on clinical interviews, self-report questionnaires, and subjective evaluation by mental health practitioners. Nonetheless, these approaches commonly have inherent drawbacks, including subjectivity, susceptibility to biases, and the inability to conduct real-time monitoring. As such, there is an increasing demand for objective and automatic approaches that can enable early detection and evaluation of anxiety and depression. Computer vision and machine learning advancements have opened up new possibilities for creating automated systems that can quantify and interpret human emotions. Facial expression analysis of emotions has become a viable means of determining the emotional state of a person due to the large amount of information carried by such expressions. Given the current state of technology, we propose

*Computer Science Department , LESIA Laboratory , University of Biskra, Biskra, Algeria. Email: houcem.rezaiguia@univ-biskra.dz.
†Computer Science Department , LESIA Laboratory , University of Biskra, Biskra, Algeria. Email: abdelhamid.djeffal@univ-biskra.dz.

a new framework known as ANet (AlexNet Convolutional Neural Networks model) that aims to detect depression and anxiety through the analysis of facial expressions linked to emotional reactions. The ANet model combines cutting-edge deep learning approaches. For example, it uses the AlexNet convolutional neural network (CNN) model to train itself and subsequently to detect relevant features from salient images of faces that have been extracted and cropped. The AlexNet configuration has been popularly trained with much success for various general applications of computer vision, including object detection and image classification that emphasise non-facial characteristics. Thus, for the detection of anxiety and depression, we applied the AlexNet model to assess whether it, too, could detect and measure emotionality related to anxiety and depression. The suggested framework adopts a systematic approach in detecting anxiety and depression. It starts by acquiring a dataset comprising facial images from an open repository, thereby assuring variation in the emotional states and offering adequate representation. Face detection is then carried out using the Viola-Jones algorithm, enabling appropriate localisation of the area of interest on the face. After detecting the facial area, it is considered the Region of Interest (ROI) and used as the input for the further stages of analysis. The ANet model utilises the AlexNet CNN model to train during the training process and learn network parameters' optimisation. It comes from the ANet dataset, correlating facial expressions with their associated derived emotions—and depression and anxiety, for example. Therefore, once the training is complete, the ANet model can effectively and accurately assign emotions such as anger, sadness, happiness, disgust, fear, and neutrality. To assess degrees of anxiety and depression, the ANet framework uses attributed emotions to determine the degree of anxiety and depression. It can classify from the highest level of anxiety down to minor assessments: high anxiety, moderate anxiety, mild anxiety, and no anxiety. This is beneficial for trained practitioners because it provides them with more data to use in treatment assessments and options. In summary, the ANet model presents an innovative and promising approach to the detection of depression and anxiety through facial emotion analysis. Using deep learning along with the AlexNet convolutional neural network model, the system seeks to offer a clear, automated, and immediate way to identify and assess depression and anxiety. These advances have the potential to transform mental health treatment by enabling early interventions and personalised treatments, thereby optimizing overall health and quality of life in individuals who suffer from a broad array of mental health conditions. This paper is organised in the subsequent way. Section 2 presents a comprehensive examination of the existing literature. The methodology is outlined in Section 3. Section 4 elaborates on the findings, while Section 5 presents the conclusions.

2 Litterature review

Facial expression analysis has recently garnered significant attention as a method for detecting human anxiety. A person's facial expression often tells an observer how they are feeling. The expression shows how someone feels about themselves or how they feel about the person watching [18]. Researchers have been working on automatic anxiety detection for a long time. Studies show that it is possible to automatically recognise worries through facial clues. Some methods are more invasive, including blood or saliva testing, while others are less invasive and include collecting images. Deep Learning (DL) is a type of machine learning. This method uses an artificial neural network as a foundation for training and characterisation [15]. The information gained during deep learning helps us understand data, such as pictures of faces. The first step in using deep learning to detect facial expressions is to choose a data-intensive expression model. The next stage is to use Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), Recurrent Neural Networks, and other similar methods. Deep learning is a more advanced form of machine learning that is better at photo identification than typical heuristic classification methods. People use deep learning frameworks for many things, including recognising facial emotions and movements.

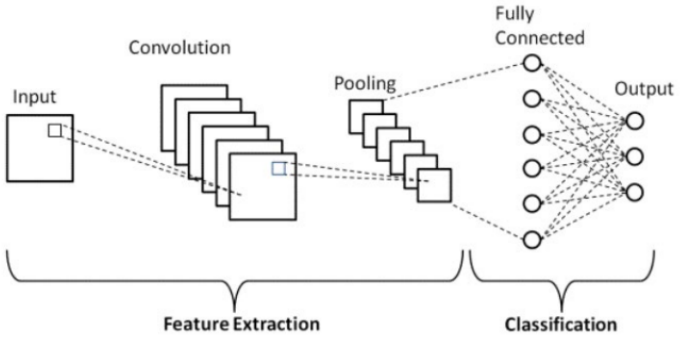


Figure 1: Structure of deep Neural Networks (DNN).

CNNs, or Convolutional Neural Networks, are very advanced neural networks that can respond well to nearby units in their receptive field. These networks have done a wonderful job at processing large images. Figure 1 has additional information. The architecture of CNN has three separate layers. The convolutional layers, the pooling layers, and the fully connected (FC) layers are the three levels. The system has a complete categorisation framework, which makes it possible to find and confirm images. Some of the most well-known convolutional neural network designs are LeNet, AlexNet, VGG, GoogLeNet, and ResNet. The authors [1] suggested using the VGG16, VGG19, and ResNet V2 models to look at facial expression recognition systems that use CNN techniques to find stress. The VGG16 model had the highest accuracy at 0.7665, the VGG19 model had 0.7257, and the ResNet V2 model had 0.8249. CNN, which is a traditional framework in the field of machine learning, has demonstrated remarkable success in a variety of computer tasks, including image

enhancement, image processing, and picture identification.

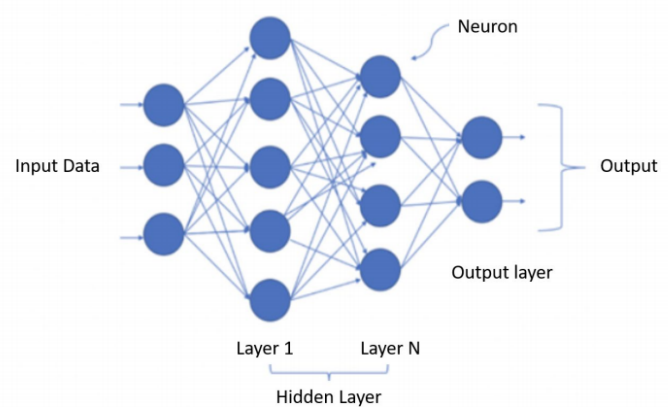


Figure 2: Structure of deep Neural Networks (DNN).

Figure 2 depicts the architecture of a deep neural network. DNN is a deep learning framework distinguished by the inclusion of a minimum of one hidden layer within its neural network design. This type of network is akin to shallow neural networks. DNN enables the modelling of complex, non-linear systems. A DNN is a discriminative model that may be trained using backpropagation methods. The trained DNN model can obtain the estimated ratio, an essential component of facial expression. EmotionNet Nano is the inaugural neural network architecture for face expression identification using DNN. In a study [14] comparing the performance of EmotionNet Nano – A and Nano – B networks, EmotionNet Nano – A achieved accuracy similar to the best larger networks. Conversely, EmotionNet Nano-B exhibits lower accuracy relative to the top-performing networks; however, it still achieves comparable accuracy while being three orders of magnitude smaller in terms of parameter count. EmotionNet Nano variations offer an optimal balance of accuracy and complexity, making them appropriate for embedded applications.

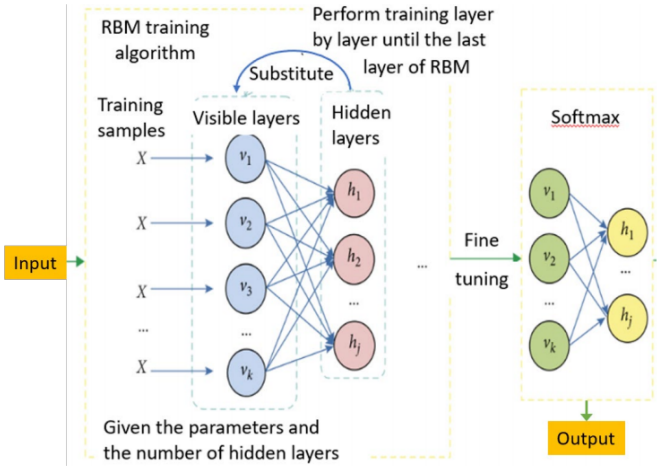


Figure 3: Structure of deep Neural Networks (DNN).

The Deep Belief Network (DBN) method represents a neural network employed in machine learning, applicable in both unsupervised and supervised learning contexts. Figure 3 demonstrates the performance of the Deep Belief Network (DBN) as it processes the input image to produce the output. DBN creates a joint distribution that connects observed data with labels. This technique is different from the traditional neural network approach used by the discriminant model. By adjusting the weights of its neurones to achieve the highest probability, the entire neural network can generate training data. Wu and Qiu [26] did a study to see how well DBN could recognise facial expressions. They used LBP and an improved DBN to get recognition rates in the JAFFE database. In conclusion, the DBN method works better than other algorithms on three datasets. The DBN model uses deep learning networks to find more complex features and boost recognition rates. An RNN sends state information across its network in a systematic way so that it can handle a wider range of time series structural inputs. RNNs can sort image sequences by their time-based relationships. redictions online. This method works best for recognising facial emotions in real time [11], and [20] have made an improved version. However, a Recurrent Neural Network (RNN) is a type of neural network that has trouble dealing with the problem of vanishing gradients in recursive situations. Giannakakis et al in [10] developed a system capable of identifying emotional states of stress and anxiety through the analysis of video-recorded facial expressions. Using many approaches, including Active Appearance Models, Optical Flow, and rPPG, they extracted the most pertinent information for stress classification. A K-NN classifier achieved a classification accuracy of 87.72%. The study by [24] proposed a system that detects symptoms of stress using Facial Action Units (FAUs) derived from movies. The researchers conducted a binary classification employing various elementary classifiers on facial action units (FAUs) taken from each video frame, attaining an accuracy of as much as 74% in subject-independent categorisation and 91% in subject-dependent classification.

3 Materials and Methods

Facial emotion-based anxiety level identification is performed through a series of modules, each contributing to the comprehensive analysis of video data.

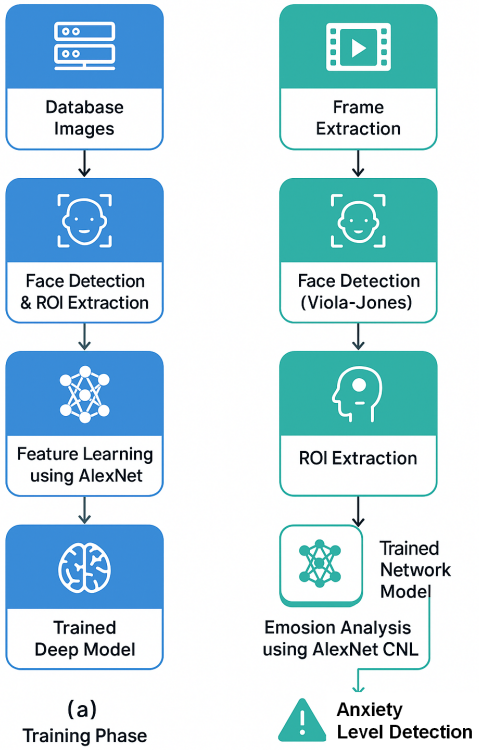


Figure 4: Overall proposed methodology. (a) training Module. (b) testing Module.

a- Training module:

1. **RADIATE facial stimulus collection:** The RADIATE facial stimulus collection includes more than 1,500 distinct photographs featuring a diversified group of more than a hundred models, comprising 25% non-Hispanic White and 75% from minority or ethnic groups. Each model exhibited 16 distinct facial expressions [23], offering a broad spectrum of emotions within a stimulus set that reflects various racial and ethnic backgrounds. The RADIATE stimulus set images were created to allow for integration with other face stimulus sets, utilising a standardised scarf template that accompanies the face stimuli.
2. **Face detection via Viola–Jones algorithm:** Using the Viola–Jones algorithm, face detection was implemented. The detected face region is cropped for the next step, known as ROI extraction. In 2001, Paul Viola and Michael Jones developed the Viola–Jones object detection framework [25], the first object detection system to deliver competitive object detection rates in real time. Although it may be trained to recognise a range of object classes, it was originally driven by the problem of face detection. The open-source implementation of the Viola–Jones detector made it famous. This method is frequently used to detect the face region in an image in order to find an object of unknown size. It has high efficiency and accuracy.

   The Viola–Jones method employs three strategies:

   a. Integral image utilised for the purpose of feature extraction Rectangular in nature, Haar-like features are derived through the use of an integral image.

   b. Adaboost is a face identification algorithm based on machine learning [12]. The term ”boosted” refers to the fact that the classifiers at each stage of the cascade is inherently complex, constructed from simple classifiers using one of four boosting approaches (weighted voting). The Adaboost algorithm represents a systematic approach to learning that starts with a weak classification and then learns and constructs a strong classification using the weight value.

   c. A cascade classifier is used to efficiently merge many features. The word ”cascade” in the classifier name refers to the fact that the final classifier is made up of numerous smaller classifiers (stages) that are applied one after another to a region of interest until the candidate is rejected or all stages are passed [27]. After cascading each of the strong classifiers, the model may acquire the non-face and face regions.
3. **Training using AlexNet CNN model:** Once the facial region has been extracted, the training phase can begin. Throughout the training phase , images are trained as feature maps using the AlexNet CNN model. AlexNet marked the inaugural use of a convolutional neural network in the LSVRC competition, as noted by Krizhevsky et al. 2012 [3], achieving a significantly higher accuracy than all prior models, including the runner-up. AlexNet employs the graphics processing unit (GPU) to improve performance. AlexNet consists of five convolutional layers, three max-pooling layers, two normalisation layers, two fully connected layers, and one softmax layer within its architecture. Each convolutional layer utilises convolutional filters in conjunction with the nonlinear activation function ReLU. The implementation of max pooling involves the utilisation of pooling layers. The fixed input size is a result of the inclusion of fully connected layers. The input dimensions are commonly stated as 224x224x3; however, due to the application of padding, the true dimensions are 227x227x3. AlexNet comprises a total of 60 million parameters.

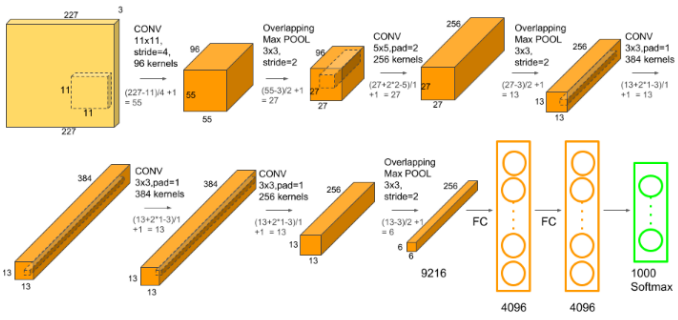


Figure 5: AlexNet architecture.

AlexNet consists of three primary components:

(1) Utilises the ReLU non-linear activation function to

address the vanishing gradient issue more efficiently compared to sigmoid and tanh activation functions found in other neural networks;
(2) Incorporates dropout and data augmentation within the network layer to mitigate overfitting; and
(3) Leverages multiple parallel GPUs to enhance computational throughput during the training process.

Table 1: Architecture of AlexNet CNN

AlexNet Layers	# Kernels	Kernel Size	Stride	Padding	Output Size
Conv1	96	11×11	[4,4]	[0 0 0 0]	$96 \times 55 \times 55$
Mpool1	96	3×3	[2,2]	[0 0 0 0]	$96 \times 27 \times 27$
Conv2	256	5×5	[1,1]	[2 2 2 2]	$256 \times 27 \times 27$
Mpool2	256	3×3	[2,2]	[0 0 0 0]	$256 \times 13 \times 13$
Conv3	384	3×3	[1,1]	[1 1 1 1]	$384 \times 13 \times 13$
Conv4	384	3×3	[1,1]	[1 1 1 1]	$384 \times 13 \times 13$
Conv5	256	3×3	[1,1]	[1 1 1 1]	$256 \times 13 \times 13$
Mpool5	256	3×3	[2,2]	[0 0 0 0]	$256 \times 6 \times 6$
Fc6					$4096 \times 1 \times 1$
Fc7					$4096 \times 1 \times 1$
Fc8 & Softmax					$5 \times 1 \times 1$

b- Testing module:

1. **Video Acquisition:** Videos used for analysis are collected from a publicly available database. The selection of diverse videos ensures representation across different emotional expressions and enables a robust assessment of anxiety levels.
2. **Frames Extraction & Face Detection:** Upon acquiring the videos, frames are extracted from each video to facilitate frame-by-frame analysis. Subsequently, face detection is applied to each frame using Haar Feature-based Cascade Classifiers, specifically the Viola-Jones algorithm. This process accurately localizes facial regions in each frame. After successful face detection, the detected faces are cropped, isolating the Region of Interest (ROI). The ROI extraction step focuses on obtaining the essential facial features necessary for precise emotion analysis.
3. **Facial Emotions Analysis:**The used AlexNet model for facial emotions classification is gained by its ability to pull out important features from pictures of faces. This model is emotion classification-oriented and relies on a classification approach for implementation. The architecture of AlexNet is composed of five convolutional layers along with three fully connected layers.

To address overfitting, dropout is introduced as a regularization technique. The training process is carried out using stochastic momentum gradient descent (SGDM) optimization with an initial learning rate of 0.0003. The training dataset is processed through multiple epochs, indicating the total training time on the complete dataset. The architecture, training dataset, and options for training are predefined before training the AlexNet network. During the classification stage, the trained network classifies each video frame into distinct emotional categories, including Sadness, Anger, surprise, Happiness, Disgust, fear and Neutral. Anxiety Level Identification: The identification of anxiety levels is determined based on the total amount of positive and negative emotions observed throughout the entire video. Positive emotions consist of 'happy' and 'neutral,' while negative emotions encompass 'sad,' 'anger,' fear,' and 'disgust.' By analyzing the balance between positive and negative emotions, the anxiety levels are categorized into four distinct levels, serving as a comprehensive assessment of anxiety.

There are still some restrictions even though the suggested methodology shows encouraging results in determining anxiety levels from emotional facial expressions. For training, the method mostly uses the RADIATE dataset, which is varied but might not adequately represent the unpredictability of real-world situations like shifting lighting, occlusions, or impromptu facial expressions. Furthermore, dividing distinct emotion categories into positive and negative groups is the basis for classifying anxiety levels, which may ignore more nuanced affective cues.

4 Experimental Results

An evaluation of the suggested facial expression recognition system was conducted through experiments employing the Radiate dataset, comprising a total of 1504 instances spanning seven emotion classes. The dataset was partitioned into training and validation sets employing a hold-out validation strategy, where 90 % of the data was allocated for training (imdsTrain) and 10 % for validation (imdsValidation). A pretrained AlexNet convolutional neural network was fine-tuned on the training data using stochastic gradient descent with momentum (SGDM). Data augmentation strategies, including random translation and horizontal flipping, were applied to improve generalization. The network was trained for 200 epochs, and performance was monitored on the validation set throughout the training phase. The confusion matrix indicates that the classifier demonstrates robust performance across all classes, with minimal misclassification. Each row denotes the actual class, whilst each column signifies the anticipated class. The matrix values represent the frequency of instances assigned to each category. For example, class 1 anger achieved 210 true positives out of 218 instances, while class 4 happy achieved the highest true positive count with 315 correct predictions. The overall accuracy on the validation set reached 98.01 %, highlighting the effectiveness of the proposed approach.

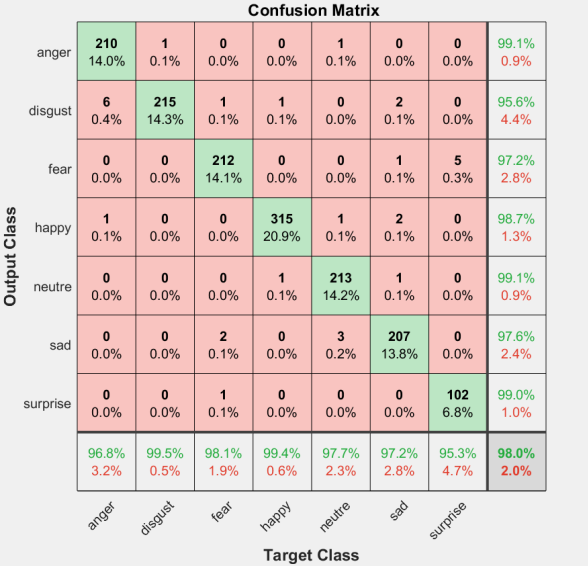


Figure 6: Confusion matrix.

In addition to accuracy, several evaluation metrics were computed to provide a more comprehensive analysis. Table 2 summarizes class-wise and overall performance:

Table 2: Performance metrics

Metric	Value
Accuracy	98.01%
F1-score	97.87%
Average Sensitivity (Recall)	97.72%
False Positive Rate	0.34%
Average Specificity	99.66%
Matthews Correlation Coefficient (MCC)	0.9755
Average Precision	98.05%
Cohen's Kappa	0.9186

Comprehensive The performance metrics of the facial emotion-based anxiety level identification system are summarized in Table 2. These indicators offer insights into the efficacy and resilience of the suggested methodology.

Precision: This metric quantifies the model's confidence when predicting a given class. For example, the precision for class 1 was 99.06%, indicating that when the model predicts this emotion, it is very likely to be correct.

Recall (Sensitivity): This measures how well the model captures all true instances of a class. Class 1 showed a recall of 96.77%, meaning that most instances were correctly identified.

F1-Score: The harmonic mean of precision and recall provides a balanced measure., especially useful for imbalanced datasets. For class 1, the F1-score reached 97.90%.

Specificity: This indicates how accurately the model recognizes negative cases, i.e., instances that do not belong to a particular class. The specificity for class 1 was 99.85%, showing excellent performance in avoiding false positives.

Accuracy: Defined as the total proportion of correctly predicted samples across all classes, the overall accuracy achieved was 98.01%.

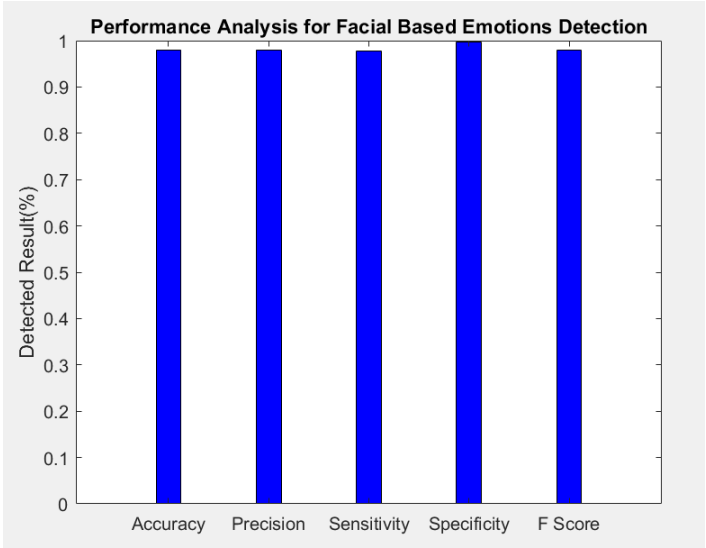


Figure 7: Performance analysis for facial based emotions detection.

The performance Analysis in Fig 7 shows that the model performs well in identifying emotions. The high values for precision, recall, F1-score, and specificity indicate that the model exhibits reliability and is competent at distinguishing correctly among various emotional states. In addition, the high overall accuracy of the model supports this claim of performance.

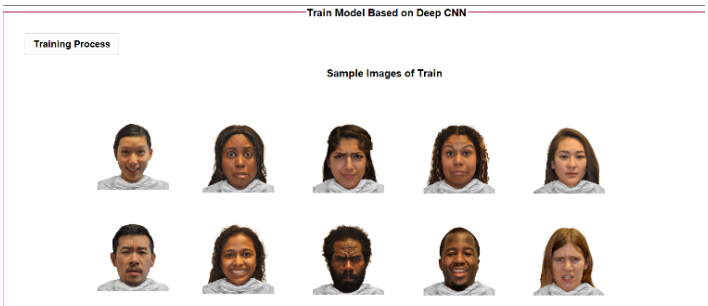


Figure 8: Sample images of train model based on deep CNN.

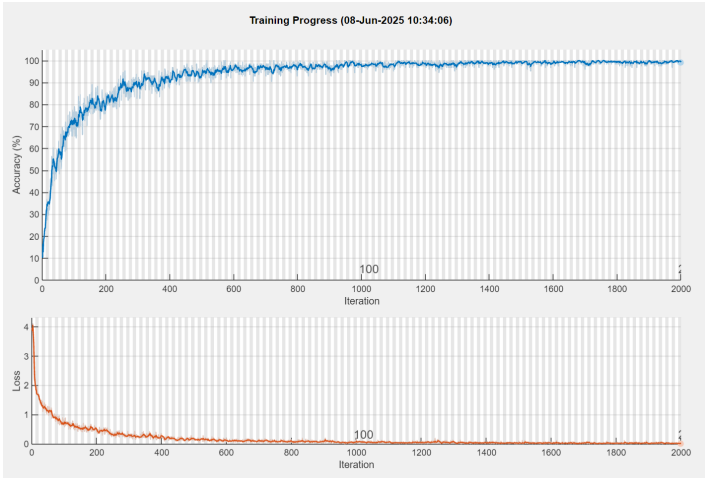


Figure 9: Training process.

Figure 9 illustrates the training progress over 2,000 iterations, showing both accuracy and loss convergence. The model demonstrates rapid learning in early iterations, with accuracy increasing sharply and stabilizing near 98–99%, while the loss gradually decreases and converges towards zero. This indicates an effective optimization process and minimal overfitting, further validating the generalization capability of the model.

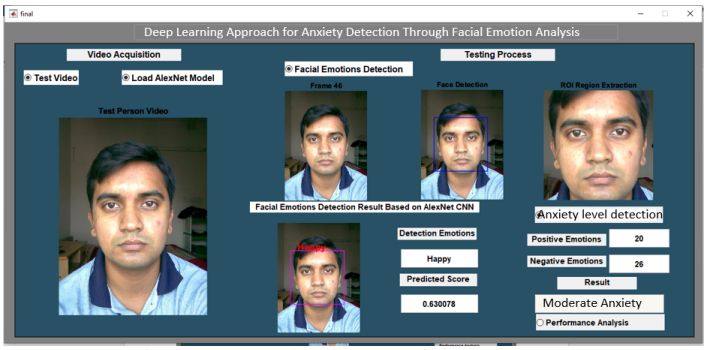


Figure 10: Testing process for person 1.

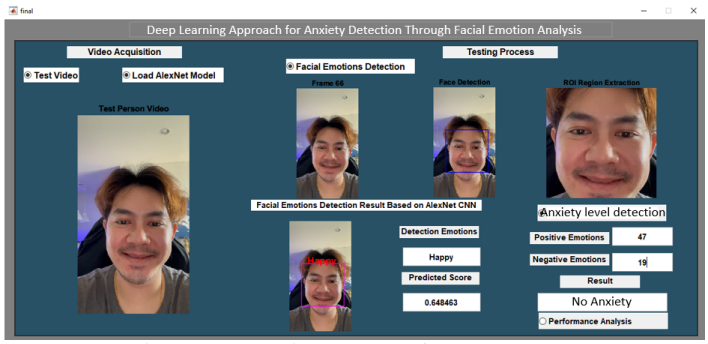


Figure 11: Testing process for person 2.

Table 3: Algorithm Accuracy Comparison

Study	Algorithm Type	Accuracy Range
[5]	CNNEELM (Convolutional Neural Network with Extreme Learning Machine hybrid)	98% (CK+), 96.53% (FER2013)
[2]	Discriminant Analysis (Quadratic), SVM, Random Forest, K-Nearest Neighbors, Naive Bayes, Decision Tree, Multilayer Perceptron	100% (Discriminant Analysis, SVM, Random Forest); others not reported
[6]	AdaBoost+SVM; LDA	93% (SVM+AdaBoost)
[17]	Radial Basis Function Support Vector Machine (RBF SVM) (with HOG+LBP); others not specified	94% (RBF SVM HOG+LBP); F1: 0.93
Our study	AlexNet Convolutional Neural Networks (CNN) model	98.01% with RADIATE dataset

Table 4: AlexNet Accuracy Comparison Across Different Studies

Study	Dataset	AlexNet Accuracy Range
[4]	JAFFE	89.2% (Linear Discriminant Analysis), 87.8% (K-Nearest Neighbors)
[22]	FERC-2013	>65%
[7]	CK+ + JAFFE	90%
[21]	FER2013	76%
Our study	RADIATE	98.01%

5 Conclusions

In this work, we introduced ANet, a novel and effective deep learning framework for the automatic detection and assessment of anxiety levels based on facial emotional expressions. Utilizing the AlexNet Convolutional Neural Network (CNN), our approach demonstrates a reliable, structured pipeline that includes face detection, region of interest (ROI) extraction, feature learning, and emotion classification. By mapping classified emotions to anxiety levels, the system enables real-time, non-invasive mental health screening, which is particularly valuable in preventive psychological care. Extensive experiments were conducted on a diverse dataset of 1,504 instances, covering seven fundamental emotions relevant to anxiety inference. The proposed model achieved an

impressive accuracy of 98.01%, with strong supporting metrics such as F1-score (97.87%), precision (98.05%), specificity (99.66%), and Cohen's Kappa (0.9186). The accuracy and loss curves show that the training convergence patterns are stable and can be used to make predictions about new data. When compared to traditional methods and existing benchmark datasets, the ANet framework shows that it works better and is more stable. This system is different from traditional diagnostic tools that depend on subjective reporting or clinical interviews. It is objective, automated, and scalable, and it can help clinicians intervene early. The good results show that deep learning and facial expression analysis can be used to find anxiety. The ANet model is a big step forward in the field of affective computing, and it could also be used in bigger mental health assessment systems like telemedicine platforms or digital mental health apps. We want to make the ANet framework more scalable and applicable to more situations by looking at bigger and more varied datasets in future research. We also want to look into how to combine different types of data, like audio and physiological signals, to make anxiety detection systems even more accurate and reliable. The ANet framework is a big step towards meeting the urgent need for objective and automated ways to detect anxiety. Such systems will make it possible for mental health care to offer personalised and timely interventions.

Future research will concentrate on enhancing the ANet framework's robustness and scalability through the use of larger and more varied datasets to improve population-to-population generalisation. To capture more detailed anxiety-related patterns, multimodal fusion—which combines facial analysis with audio and physiological signals—will be investigated. To enhance performance in practical settings, cutting-edge deep learning architectures and transfer learning strategies will be researched. Additionally, privacy-preserving strategies will be used to guarantee ethical and secure deployment in clinical and remote healthcare applications, and explainable AI techniques will be incorporated to improve interpretability.

References

[1] J. Almeida and F. Rodrigues. Facial expression recognition system for stress detection with deep learning. In *Proceedings of the 23rd International Conference on Enterprise Information Systems*, pages 256–263, 2021.

[2] F. Ayache and A. Alti. Performance evaluation of machine learning for recognizing human facial emotions. *Revue d'Intelligence Artificielle*, 34(3):267–275, 2020.

[3] S. H. Babu, S. A. Birajdhar, and S. Tambad. Face recognition using entropy based face segregation as a pre-processing technique and conservative bpso based feature selection. In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, pages 1–8, 2014.

[4] G. Ballikaya and D. Kaya. Facial expression recognition techniques and comparative analysis using classification algorithms. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 12(3):596–607, 2023.

[5] N. Banskota, A. Alsadoon, P. W. C. Prasad, A. Dawoud, T. A. Rashid, and O. H. Alsadoon. A novel enhanced convolution neural network with extreme learning machine: facial emotional recognition in psychology practices. *Multimedia Tools and Applications*, 82(5):6479–6503, 2023.

[6] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: Machine learning and application to spontaneous behavior. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 568–573, 2005.

[7] N. R. Calvo-Ariza, L. F. Gómez-Gómez, and J. R. Orozco-Arroyave. Classical fe analysis to classify parkinson's disease patients. *Electronics*, 11(21):3533, 2022.

[8] W. T. Chew, S. C. Chong, T. S. Ong, and L. Y. Chong. Facial expression recognition via enhanced stress convolution neural network for stress detection. *IAENG International Journal of Computer Science*, 49(3):1–10, 2022.

[9] L. Fontes, P. Machado, D. Vinkemeier, S. Yahaya, J. J. Bird, and I. K. Ihianle. Enhancing stress detection: A comprehensive approach through rppg analysis and deep learning techniques. *Sensors*, 24(4):1096, 2024.

[10] G. Giannakakis, M. Pediaditis, D. Manousos, E. Kazantzaki, F. Chiarugi, P. G. Simos, K. Marias, and M. Tsiknakis. Stress and anxiety detection using facial cues from videos. *Biomedical Signal Processing and Control*, 31:89–101, 2017.

[11] A. Graves, C. Mayer, M. Wimmer, J. Schmidhuber, and B. Radig. Facial expression recognition with recurrent neural networks. In *Proceedings of the International Workshop on Cognition for Technical Systems*, 2008.

[12] R. Karthika and L. Parameswaran. Study of gabor wavelet for face recognition invariant to pose and orientation. In *Proceedings*, pages 501–509. 2016.

[13] S. Latif, H. S. Ali, M. Usama, R. Rana, B. Schuller, and J. Qadir. Ai-based emotion recognition: Promise, peril, and prescriptions for prosocial path. *arXiv preprint arXiv:2211.07290*, 2022.

[14] J. R. Lee, L. Wang, and A. Wong. Emotionnet nano: An efficient deep convolutional neural network design for real-time facial expression recognition. *Frontiers in Artificial Intelligence*, 3, 2021.

[15] S. Lin. Deep learning research and how to get immersed - towards data science. `https://towardsdatascience.com`, 2021. Accessed on March 27, 2022.

[16] M. A. Lumley, J. L. Cohen, G. S. Borszcz, A. Cano, A. M. Radcliffe, L. S. Porter, H. Schubiner, and F. J. Keefe. Pain and emotion: a biopsychosocial review of recent research. *Journal of Clinical Psychology*, 67(9):942–968, 2011.

[17] S. B. Luo, K. T. Nguyen, X. D. Jiang, J. F. Wu, B. H. Wen, Y. Zhang, G. Chierchia, H. Talbot, T. Bourouina, D. L. Kwong, and A. Q. Liu. A high performance of single cell imaging detection with deep learning. In *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, pages 356–360, 2019.

[18] V. Maydych, M. Claus, C. Watzl, and T. Kleinsorge. Attention to emotional information is associated with cytokine responses to psychological stress. *Frontiers in Neuroscience*, 12, 2018.

[19] A. Mehrabian and S. R. Ferris. Inference of attitudes from nonverbal communication in two channels. *Journal of Consulting Psychology*, 31(3):248–252, 1967.

[20] A. Mostafa, M. I. Khalil, and H. Abbas. Emotion recognition by facial features using recurrent neural networks. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, pages 417–422, 2018.

[21] T. Eswara Prasad. An efficient facial expression recognition system using novel image classification by comparing cnn over res net. *Journal of Pharmaceutical Negative Results*, 13, 2022.

[22] K. U. Singh, A. Kumar, G. Kumar, T. Singh, S. Kumar, and S. P. Yadav. An autonomous emotion recognition strategy employing deep learning for self-learning. In *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, pages 883–888, 2023.

[23] N. Tottenham, J. W. Tanaka, A. C. Leon, T. McCarry, M. Nurse, T. A. Hare, D. J. Marcus, A. Westerlund, B. Casey, and C. Nelson. The nimstim set of facial expressions: Judgments from untrained research participants. *Psychiatry Research*, 168(3):242–249, 2009.

[24] C. Viegas, S.-H. Lau, R. Maxion, and A. Hauptmann. Towards independent stress detection: A dependent model using facial action units. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2018.

[25] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages 511–518, 2001.

[26] Y. Wu and W. Qiu. Facial expression recognition based on improved deep belief networks. *AIP Conference Proceedings*, 020130, 2017.

[27] H. Yang and X. A. Wang. Cascade classifier for face detection. *Journal of Algorithms & Computational Technology*, 10(3):187–197, 2016.

Authors

Houcem REZAIGUIA is a Ph.D. student and researcher in computer science at Mohamed Khider University, Biskra, Algeria. He is affiliated with the LESIA research laboratory, where his work focuses on automatic anxiety detection from facial expressions using deep learning. His research interests include affective computing, computer vision, and artificial intelligence applied to emotional state analysis.

Abdelhamid DJEFFAL is a Professor in the Department of Computer Science at Mohamed Khider University, Biskra, Algeria. He supervises several research projects in the field of artificial intelligence and signal processing, with a particular interest in emotion recognition, machine learning, and intelligent systems. He is also a member of the LESIA research laboratory.

Journal Submission

The International Journal of Computers and Their Applications is published four times a year with the purpose of providing a forum for state-of-the-art developments and research in the theory and design of computers, as well as current innovative activities in the applications of computers. In contrast to other journals, this journal focuses on emerging computer technologies with emphasis on the applicability to real world problems. Current areas of particular interest include, but are not limited to: architecture, networks, intelligent systems, parallel and distributed computing, software and information engineering, and computer applications (e.g., engineering, medicine, business, education, etc.). All papers are subject to peer review before selection.

A. Procedure for Submission of a Technical Paper for Consideration

1. Email your manuscript to the Editor-in-Chief, Dr. Ajay Bandi. Email: ajay@nwmissouri.edu.
2. Illustrations should be high quality (originals unnecessary).
3. Enclose a separate page (or include in the email message) the preferred author and address for correspondence. Also, please include email, telephone, and fax information should further contact be needed.
4. **Note**: Papers shorter than 10 pages long will be returned.

B. Manuscript Style:

1. **WORD DOCUMENT**: The text should be **double-spaced** (12 point or larger), **single column** and **single-sided** on 8.5 X 11 inch pages. Or it can be single spaced double column.

   LaTex DOCUMENT: The text is to be a double column (10 point font) in pdf format.
2. An informative abstract of 100-250 words should be provided.
3. At least 5 keywords following the abstract describing the paper topics.
4. References (alphabetized by first author) should appear at the end of the paper, as follows: author(s), first initials followed by last name, title in quotation marks, periodical, volume, inclusive page numbers, month, and year.
5. The figures are to be integrated in the text after referenced in the text.

C. Submission of Accepted Manuscripts

1. The final complete paper (with abstract, figures, tables, and keywords) satisfying Section B above in **MS Word format** should be submitted to the Editor-in-Chief. If one wished to use LaTex, please see the corresponding LaTex template.
2. The submission may be on a CD/DVD or as an email attachment(s). **The following electronic files should be included:**
   - Paper text (required).
   - Bios (required for each author).
   - Author Photos are to be integrated into the text.
   - Figures, Tables, and Illustrations. These should be integrated into the paper text file.
3. Reminder: The authors photos and short bios should be integrated into the text at the end of the paper. All figures, tables, and illustrations should be integrated into the text after being mentioned in the text.
4. The final paper should be submitted in (a) pdf AND (b) either Word or LaTex. For those authors using LaTex, please follow the guidelines and template.
5. Authors are asked to sign an ISCA copyright form (http://www.isca-hq.org/j-copyright.htm), indicating that they are transferring the copyright to ISCA or declaring the work to be government-sponsored work in the public domain. Also, letters of permission for inclusion of non-original materials are required.

Publication Charges

After a manuscript has been accepted for publication, the contact author will be invoiced a publication charge of **$500.00 USD** to cover part of the cost of publication. For ISCA members, publication charges are **$400.00 USD** publication charges are required.

Revised 2020