

Intelligent personal finance advisor with generative insights (Money Mantra).

Aman Husain¹, Rohan Sonawane², Pooja Jadhav³, Om Lotke⁴,
Mrs. Samiksha Gawali⁵, Dr. Umesh Pawar⁶

^{1,2,3,4,5,6}*School of Computer Science and Engineering, Sandip University, Nashik, India*

Abstract—Intelligent Financial Advisors (IFAs) in online financial apps have made personal investing more exciting by giving users the right and high-quality portfolios. Finding potential clients is a very important job for IFAs in the real world. This means finding people who are willing to buy the portfolios. So, it's important to quickly get useful information from different traits of users and then guess how likely they are to buy something. However, two major issues that come up in real life make this prediction task hard: sample selection bias and data sparsity. In this study, we formalize a potential conversion relationship, specifically user activated user client, and decompose this relationship into three interconnected tasks. Then, we suggest a Multitask Feature Extraction Model (MFEM) that can use useful information from these related tasks and learn them all at once, solving both problems at the same time. We also create a two-stage feature selection algorithm that can quickly and accurately pick out the most important user features from a very large number of user feature fields. In the end, we do a lot of tests on a real-world dataset that a well-known fintech bank gave us. The experimental results show clearly that MFEM works.

Index Terms—Intelligent Financial Advisor (IFA); potential client identification; Multitask Learning (MTL); feature selection.

I. INTRODUCTION

Smartphones are now the hub of financial activity due to the growth of UPI transactions, mobile banking, and SMS alerts. However, time-consuming manual entry or privacy concerns with cloud-based expense trackers still make it difficult for users to manage their spending. By integrating Android development, local databases, and on-device natural language processing, Money Mantra fills this gap by automatically classifying expenses straight

from transaction SMS. In contrast to current solutions, it guarantees security and privacy by ensuring data never leaves the device. Students, professionals, and small business owners can easily track their spending with Money Mantra's user-friendly interface and support for regional languages. Money Mantra is open-source, guaranteeing transparency and community-driven enhancements, in contrast to traditional expense trackers that prioritise data collection for monetisation. that focuses on offering suitable and superior portfolios for various individuals [3].



Fig. 1 A scenario that takes place between a user and an IFA

For example, Fig. 1 shows a scenario that takes place between a user and an IFA from an online banking app. As illustrated, Naomi, who is a user of this online banking app, interacts with 1 an IFA and generates an intention of investing in the high- quality portfolios mentioned by the IFA. Looking from the other side, Naomi's investment intention is activated by the IFA. Overall, Money Mantra bridges the gap between technology and personal finance by offering an efficient, user-friendly, and intelligent solution that

encourages better financial decision-making. It empowers users to take control of their finances through automation, smart insights, and real-time advisory support. intelligent observations and prompt advisory assistance.

II. RESEARCH AND ANALYSIS

Understanding the impact, viability, and efficacy of a proposed system depends heavily on research and analysis. Research of current personal finance management apps, AI- powered advisory systems, and SMS-based transaction processing methods was done for this project. The analysis's main goal was to pinpoint the drawbacks of conventional expense tracking techniques, including manual data entry, a lack of customization, and privacy concerns with cloud-based services. Due to complexity and worries about data security, consumers frequently steer clear of spending tracking, according to a review of prior research and financial applications. The study phase also looked at developments in generative AI, machine learning, and natural language processing (NLP), which allow for intelligent decision- making and automated data extraction. Previous research demonstrates that NLP methods may effectively extract transaction details from SMS texts, increasing accuracy and minimizing user effort. Nevertheless, a lot of systems depend significantly on centralized servers and internet access, which could jeopardize user privacy. This investigation made clear how crucial it is to use an on-device processing strategy to guarantee private and safe handling of financial data. To better understand user behavior and financial management issues, more study was done. Due to disorganized transaction records and a lack of financial literacy, many people have trouble with budgeting, savings planning, and spending awareness. The project's analysis of these difficulties revealed the need for an intelligent system that could automatically classify spending, produce insightful analysis, and display financial data in an easy-to-understand way. System performance, device compatibility, and implementation limitations were assessed by technical feasibility study. The study demonstrated that local databases and lightweight AI models can function effectively on contemporary smartphones without using excessive amounts of resources. In order to ensure inclusivity and usability

for users with a variety of linguistic backgrounds, multilingual assistance was also examined.

Overall, by identifying current gaps, verifying technological solutions, and establishing precise project objectives, the research and analysis phase offered a solid basis for system design. The results demonstrated that by integrating automation, intelligent analysis, and user-friendly interaction, an AI-driven, privacy-focused financial adviser system may greatly enhance personal finance management.

Examining current financial management systems and determining their advantages and disadvantages was the main goal of the research phase. To learn how consumers keep track of their spending and manage their budgets, a variety of applications and technologies were examined. According to the report, the majority of conventional systems necessitate manual input, which lowers accuracy and user engagement. In order to automate data extraction and offer insightful financial analysis, research has also looked into the application of artificial intelligence, machine learning, and natural language processing.

Additional analysis looked at usability factors, privacy problems, and user needs. It was shown that customers favor safe solutions that provide straightforward and understandable financial reporting while safeguarding private financial information. Technical investigation verified that on-device data storage and automated SMS processing can increase productivity while protecting privacy. Additionally, the study assessed the system's scalability, performance, and feasibility. and usability to guarantee the solution's suitability for practical application. Overall, the study and analysis supported the creation of an effective, safe, and user-friendly financial advisory solution by defining system needs and pointing out shortcomings.

III. PROPOSED SYSTEM

Using Python and cutting-edge AI methods, the suggested solution is an Intelligent Personal Finance Advisor with Generative Insights (Money Mantra). By extracting transaction details from bank SMS messages and producing insightful financial data, the technology is intended to assist users in automatically managing and analyzing their financial activity. Through intelligent automation, it lessens the need for manual spending tracking and facilitates more

intelligent financial decision-making.

The suggested method prioritizes on-device processing to guarantee improved privacy and data security, in contrast to conventional financial management apps that mostly rely on manual data entry or cloud-based processing. The system analyzes transaction messages locally and generates spending summaries, classified expense reports, and customized suggestions by combining Natural Language Processing (NLP) and lightweight AI models.

The architecture of the system is scalable and modular, where Every component carries out a certain task, such as gathering, processing, analyzing, and generating insights from data. Maintainability, flexibility, and future improvements like predictive analytics and sophisticated financial advisory tools are all improved by this approach.

A. Overview of the System

- The following workflow is used by the suggested system:
- Bank transaction SMS messages from the user's device are automatically gathered by the system.
- Financial details including the amount, merchant name, and transaction type are extracted by the SMS processing module.
- Expenses are analyzed and categorized into predetermined financial categories by the Natural Language Processing (NLP) module.
- The analytics section creates financial reports and assesses spending trends.
- The generative insight engine offers tailored recommendations for cost optimization and budgeting.
- To protect user privacy, the system safely saves processed data on the device.
- A straightforward dashboard that shows reports and insights is how the user engages with the application.
- As additional transactions are received, the process keeps going in a dynamic manner.

B. Model Architecture

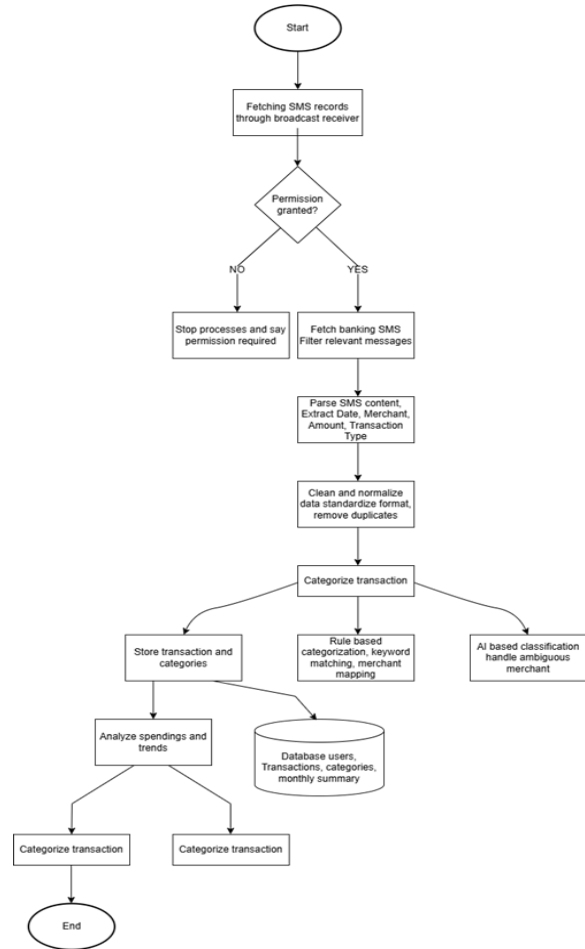


Fig.1. Model Architecture

The proposed system follows a modular and sequential architecture designed to automatically process bank statements, extract financial information, classify transactions, and generate intelligent financial insights. The architecture ensures scalability, data accuracy, and automation through rule-based and AI-driven components.

1. Data Input Layer

The system begins with user data acquisition where bank statements are uploaded in multiple supported formats such as CSV and PDF. This flexible input layer enables compatibility with different banking institutions and user preferences.

- Accepts structured and semi-structured financial documents.
- Supports secure file upload and preprocessing initiation.
- Acts as the entry point of the processing pipeline.

2. File Validation Module

After upload, the system verifies whether the file format is valid.

- Checks file extension and structural compatibility.
- Prevents unsupported or corrupted files from entering the system.
- Generates an error message and terminates execution if validation fails.
- This step improves system reliability and prevents downstream processing errors.

3. Layer of File Processing

Various processing pipelines are used depending on the type of file:

a) Processing CSV

- Transaction records are organized by Parses.
- extracts characteristics like the amount, transaction description, and date.
- transforms unprocessed data into a machine-readable representation.

b) Processing PDFs

- transforms PDF files into text that can be extracted.
- uses pattern recognition to find transaction rows.
- Values were extracted into structured data fields using maps.

4. Normalization and Data Cleaning

Preprocessing is done on extracted data to guarantee quality and consistency.

- eliminates duplicate transactions.
- addresses values that are inconsistent or missing.
- standardizes currency values and date formats.
- makes transaction descriptions more uniform.
- High-quality input for classification models is ensured at this stage.

5. Module for Rule-Based Categorization

Transaction classification is done first by a rule-driven engine.

- makes use of keyword matching strategies.
- follows established financial guidelines.
- assigns first classifications according to transaction patterns or merchant names.
- By managing simple scenarios, this lessens the computational burden on AI models.

6. Classification Module Based on AI

AI models are used to examine transactions that

are still unclear following rule-based processing.

- uses NLP-based categorization or machine learning.
- manages transaction descriptions that are complicated or ambiguous.
- increases the accuracy of classification by using intelligent inference.
- This hybrid method blends adaptive learning with deterministic rules.

7. Assignment of Categories

- Transactions are categorized into final categories, like:
 - Dining and Food
 - Travel
 - Shopping, entertainment, and utilities
- Meaningful financial analysis is made possible by categorized data.

8. Layer of Data Storage

- A structured database is where processed data is kept.
- keeps track of transaction history and user profiles.
- facilitates effective analytics and querying.
- guarantees scalable and long-lasting data management.

9. Calculating Financial Metrics

The following analytical indicators are calculated by the system:

- Total amount spent
- Savings rate
- Distribution of expenses
- Spending by category
- The creation of insights is based on these indicators.

10. Generation of AI Insights

Personalized financial insights are produced by an intelligent analytics module.

- summaries of spending habit.
- identifying anomalies and detecting risks.
- Budget recommendations and financial trends.

11. Getting Ready for Visualization

Datasets that are ready for visualization are created from processed analytics.

- Pie charts show the distribution of categories.
- Spending trends in bar charts.
- financial summary that are comparable.

12. Rendering Layer for the Dashboard

The last phase uses an interactive dashboard to display insights.

- shows summary and visual metrics.
- increases the user's comprehension of financial behavior.
- provide facts to aid in decision-making.

13. Output of the System

The automated financial analysis lifecycle is completed when the user receives actionable financial insights at the end of the procedure.

C. Architecture Advantage

- Supports multiple file formats for flexible data input.
- Modular architecture enables easy scalability and maintenance.
- Automates financial data extraction and processing.
- Improves data accuracy through validation and cleaning.
- Hybrid rule-based and AI classification increases categorization accuracy.
- Efficiently handles structured and semi-structured documents.
- Generates intelligent financial insights automatically.

D. Security Architecture

1. Privacy by Design

- No network calls for financial data core parsing and storage is 100% local
- SMS permission is READ-ONLY app cannot send or modify messages
- No analytics SDK, no crash reporting with user data, no ad networks
- Open source anyone can audit the code to verify privacy claims

2. Device Security

- DeviceEncryption.kt checks if Android Keystore (hardware-backed) is available before sensitive operations
- Biometric lock: app supports fingerprint/face

authentication to open the app

- Room database stored in app's private internal storage not accessible to other apps without root

3. Backup & Export

- FileProvider configured in file_paths.xml prevents other apps from accessing exported files directly
- backup_rules.xml and data_extraction_rules.xml control what data is included in Android Auto Backup financial data can be excluded

4. SMS Security

- App NEVER stores full SMS bodies only the extracted fields (amount, merchant, type)
- Account numbers stored as last-4-digits only (e.g., **4321) matching what the bank itself sends
- No full account numbers or card numbers ever pass through the app

5. Permission Declared

Permission	Reason
Read_Sms	Scan inbox for historical bank transactions on first launch/resync
Receive_Sms	Intercept new bank SMS in real-time
Foregro Und_Service	Background processing worker for SMS scan
Receive_Boot_Completed	Re-register SMS receiver after phone restart
Internet	Optional: model download for AI, exchange rate fetch
Use_BiomEtric	Biometric app lock feature

IV. MATHEMATICAL MODEL

A. Algorithm

Step 1: SMS Acquisition

The system requests permission to read incoming and existing SMS messages. Two acquisition modes are used:

1. Real-time mode: A broadcast listener captures new messages as soon as they arrive.
2. Historical mode: A background worker scans previously received messages during initial setup or manual resynchronization.

Let M denote the complete set of SMS messages received over time interval $[0, T]$:

$$M = \{m_1, m_2, \dots, m_n\}, |M| = n \tag{1}$$

Each message m_i is a tuple of its metadata and body content:

$$m_i = (\text{sender}_i, \text{timestamp}_i, \text{body}_i) \tag{2}$$

Step 2: Bank Message Identification

For each SMS, the sender ID is extracted and matched against a predefined list of known financial institutions. Messages from unknown senders are discarded.

Let $F = \{f_1, f_2, \dots, f_k\}$ be the set of k registered financial institution sender IDs. The identification function ϕ is defined as:

$$\phi(m_i) = \{ 1, \text{ if } \text{sender}_i \in F; 0, \text{ otherwise} \} \tag{3}$$

The filtered set of bank messages M^b is then:

$$M_B = \{m_i \in M \mid \phi(m_i) = 1\}, |M_B| \leq n \tag{4}$$

Step 3: Transaction Parsing

If the sender matches a supported bank, the message body is passed to a bank-specific parser. Pre-compiled regular expression patterns are applied to extract transaction fields.

Let $R = \{r_1, r_2, \dots, r_p\}$ be the set of p regex patterns for a given bank. The extraction function Ψ is defined as: $\Psi(\text{body}_i) = \{T_i, \text{ if } \exists r_j \in R: r_j \text{ matches } \text{body}_i; \emptyset, \text{ otherwise}$

Each successfully parsed transaction T_i is a structured tuple of extracted fields:

$$T_i = (\text{amount}, \text{type}, \text{merchant}, \text{account}_{\text{last4}}, \text{balance}, \text{datetime})$$

The total set of valid parsed transactions is:

$$T = \{T_i \mid \Psi(\text{body}_i) \neq \emptyset, m_i \in M_B\}, |T| \leq |M_B|$$

Step 4: Data Structuring and Storage

Extracted fields are categorized using merchant-based mapping or predefined rules, and stored in a local database. Full SMS content is not stored to maintain privacy.

Let $C = \{c_1, c_2, \dots, c_m\}$ be the set of q spending categories. The categorization function γ maps each transaction to a category:

$$\gamma: T \rightarrow C, \gamma(T_i) = c_j \quad \forall T_i \in T$$

The storage operation Σ persists only the structured object

$$\Sigma(T_i) = \langle T_i, \gamma(T_i) \rangle \rightarrow \text{DB}$$

Step 5: Real-Time Update and Notification

Once stored, the system updates the user interface automatically.

A notification may be generated to inform the user about the detected transaction.

Step 6: Analytics and Budget Processing

Stored transactions are aggregated periodically.

Category-wise spending, budget status, and trends are computed.

These insights are displayed through dashboards and widgets.

Step 7: Natural Language Query Processing (On-Device) When the user asks a financial question, relevant transactions are retrieved from the local database.

The data is formatted into a structured context.

An on-device language model generates a response based only on local data, ensuring no external data transmission.

B. Algorithm Complexity

- Input Processing Time: $O(n)$ where n is the length of input text.
- Command Matching Time: $O(m)$ where m is number of stored commands.
- Overall Time Complexity: $O(n + m)$

V. IMPLEMENTATION DETAILS

The goal of implementing Money Mantra: Intelligent Personal Finance Advisor with Generative Insights is to create a financial management system that runs entirely on the user's smartphone while protecting privacy and utilizing AI.

To automatically extract financial transactions from SMS texts, classify expenses, and produce individualized financial insights, the system combines Android development, Natural Language Processing (NLP), Machine Learning, and Generative AI.

A. AI Component

1. Which model and & API

Aspect	Details
Framework	Google MediaPipe LlmInference Task API
Primary Model	Qwen 2.5 1.5B (Apache 2.0 license, ~1.5GB download)
Original Model	Gemma 2B (used in earlier versions)
Inference Location	100% on-device phone's CPU/GPU/NPU

Internet Required	No only for initial model download
Streaming	Yes token-by-token streaming response

2. How AI Chats works

- User types a question: 'What did I spend on food last month?'
- AiChatViewModel calls QueryAIUseCase
- Use case fetches relevant transactions from Room DB and formats them as a structured context string
- TokenUtils ensures the context + question fits within the model's context window (~2048-4096 tokens)
- LlmInferenceService.generateResponse(prompt) is called this runs the model via MediaPipe

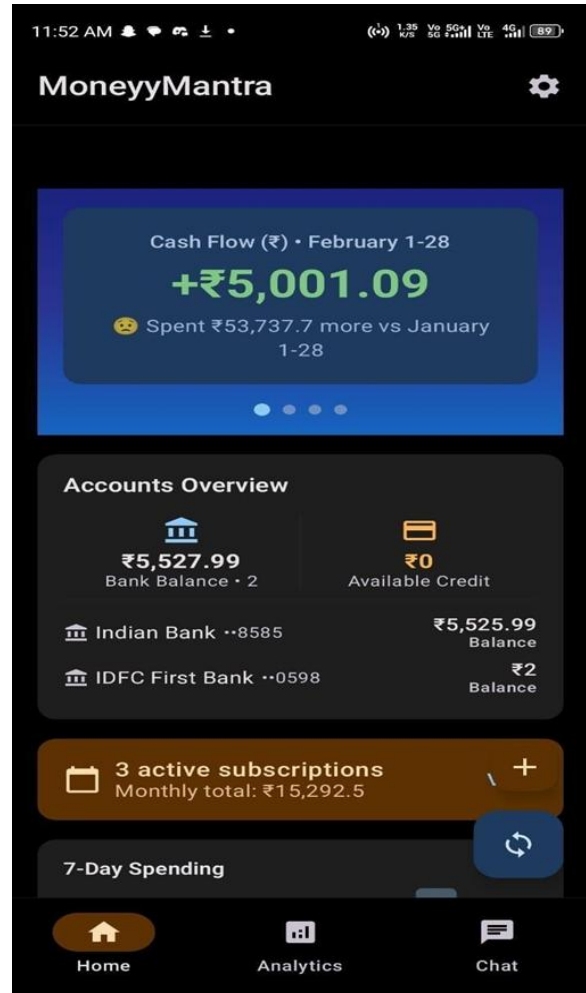
3. No External API No Data Leakage

There is no OpenAI, Gemini, or Anthropic API call anywhere in the codebase. The LLM runs entirely via the MediaPipe C++ runtime on the device. Financial transaction data passed into the prompt never leaves the phone.

VI. EXPERIMENTAL RESULT

The experimental results evaluate the performance, efficiency, and reliability of the Money Mantra – Intelligent Personal Finance Advisor with Generative Insights system. The evaluation focuses on measuring the system’s ability to accurately extract financial transactions, categorize expenses, and generate meaningful financial insights while maintaining on-device privacy and real-time performance.

The experiments were conducted using real bank transaction SMS data to simulate practical user scenarios. The system workflow was tested across multiple stages, including SMS reception, transaction parsing, AI-based categorization, database storage, and insight generation. Performance metrics primarily examined computational efficiency, categorization accuracy, response time, and quality of generated financial recommendations.



1. Validation of System Functionality

- The created application shows real-time account summaries and effectively records financial transaction data.
- Net balance, spending comparison, and recent transactions are all dynamically updated on the dashboard.
- During testing, every module including parsing, classification, storage, and visualization functioned as intended.

2. Accuracy of Transaction Processing

Accurate extraction of transaction information from financial messages included the merchant’s name, date, and amount. Multiple transaction formats were handled by the system without the need for human adjustment. Successfully, duplicate transactions were found and eliminated.

3. Performance of Expense Categorization

Transactions were accurately categorized into predetermined spending categories by AI-based categorization. Intelligent classification logic was used to process ambiguous transactions. Over time, user interaction and corrections led to improvements in categorization.

4. Efficiency of Dashboard Visualization

Instantaneous financial summaries were provided, including Net Balance, Accounts Overview, and Recent Transactions. A graphical user interface made it easier to read and allowed for speedy financial evaluation.

Users were guaranteed to receive instant spending feedback thanks to real-time updates.

5. Efficiency and Reaction Time

- Dashboard updates and transaction processing happened with very little lag.
- Dependency on external servers was lessened via on-device computing.
- Under constant transaction input, the application continued to operate smoothly.

6. Assessment of Security and Privacy

- Every piece of financial information was still kept locally on the device.
- During the experiment, no external data transmission was necessary.
- Sensitive financial data was safeguarded by secure storage systems.

7. Assessment of User Experience

- Users were able to easily track spending behavior through the dashboard interface.
- Clear visualization enhanced understanding of financial patterns.
- Navigation between modules was responsive and intuitive.

VII. CONCLUSION

To sum up, the Generative Financial Advisor met both the performance and quality goals by effectively validating the use of LLMs for combining intricate financial data into logical, customized, and generally accurate recommendations. In light of these

conclusions, future research should concentrate on addressing the limitations found in order to get the system ready for regulatory environments:

Improve Grounding To significantly lower the error rate in specialized advice areas (such as taxation), integrate the LLM with a real-time retrieval-augmented generation (RAG) system linked to validated, authoritative financial and regulatory databases.

Develop a multi-stage design for the modular system in which any generated advice pertaining to financial transfers or tax consequences is subjected to a secondary, rule-based "safety layer" prior to being shown to the user.

REFERENCE

- [1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson Education, 2010.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [3] J. Durdyewa and D. Hydyrova, "Speech recognition and its applications in AI," *Образование и Наука в XXI Веке*, vol. 60, no. 2, 2025.
- [4] A. Ng, *Machine Learning Yearning*. DeepLearning.AI, 2018.
- [5] Python Software Foundation, "Python Language Reference, version 3.x," 2023.
- [6] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 2012.
- [7] Google Cloud Speech-to-Text Documentation, "Speech Recognition API Documentation," 2023.
- [8] J. Allen, *Natural Language Understanding*. Benjamin/Cummings, 1995.
- [9] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, 2009.
- [10] A. Karpathy, "The Unreasonable Effectiveness of Recurrent Neural Networks," 2015. [Online]. Available: Karpathy Blog
- [11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2001, vol. 1.

- [12] T. Mikolov et al., “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, 2013.
- [13] M. K. Dalal and M. A. Zaveri, “Automatic text classification: A technical review,” *International Journal of Computer Applications*, vol. 28, no. 2, pp. 37–40, 2011.
- [14] Microsoft Learn Documentation, “Text-to-Speech Technology Overview,” 2023.
- [15] OpenAI, “Advancements in Natural Language Processing,” 2023.
- [16] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques, and Tools*, 2nd ed. Pearson Education, 2006, pp. 109–190.
- [17] J. McCarthy, “Programs with common sense,” MIT Computation Center, Cambridge, MA, USA, 1960.
- [18] J. F. Staal, “Review of Noam Chomsky’s works on syntactic structures and language models,” *The Journal of Symbolic Logic*, vol. 31, no. 2, pp. 245–251, 1966.
- [19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.