Automated Complaint Severity And Root Cause Analysis For Banking And Financial Services Using Natural Language Processing

Dr.K.Aruna[1], M. Abinaya[2], M. Atchaya[3], M. Subhasree[4]
[1]Associate Professor, Department of Information Technology, A.V.C.College of Engineering, Mannampandal, Mayiladuthurai.
[2,3,4] Final Year, Department of Information Technology, A.V.C.College of Engineering, Mannampandal, Mayiladuthurai.

***Abstract*—Banking and financial institutions process thousands of customer complaints daily through multiple channels including emails, mobile applications, call centers, and web portals. Manual complaint handling is time-consuming, error-prone, and often leads to delayed resolution of critical customer issues. This paper presents an intelligent, automated complaint management system leveraging Natural Language Processing techniques for severity classification, sentiment analysis, and root cause iden-tification. The proposed system employs keyword-based NLP algorithms, duplicate detection using Gestalt pattern matching, and automated routing mechanisms to streamline complaint processing. Implementation results demonstrate a 60 percent reduction in processing time, 95 percent accuracy in duplicate detection, and 100 percent automation in severity classification. The system successfully identifies complaint patterns, enabling proactive problem-solving and data-driven decision-making, ulti-mately enhancing customer satisfaction and operational efficiency in banking services.**

***Index Terms*—Natural Language Processing, Banking Systems, Complaint Management, Sentiment Analysis, Severity Classifica-tion, Machine Learning, Customer Service Automation**

I. INTRODUCTION

The banking and financial services sector faces unprecedented challenges in managing customer complaints efficiently. With the digital transformation of banking services, institutions receive thousands of complaints daily through diverse channels such as email, mobile applications, call centers, and web portals. Traditional manual complaint processing systems are inadequate for handling this volume, leading to delayed responses, misrouted complaints, and decreased customer satisfaction.

A. Background

Customer complaints serve as critical feedback mechanisms, providing insights into service quality, operational inefficiencies, and potential fraud cases. In the modern banking landscape, customers expect immediate acknowledgment and rapid resolution of their concerns. However, the unstructured nature of complaint data makes manual analysis time-consuming and prone to human error. Studies indicate that delayed complaint resolution significantly impacts customer retention, with research showing that 96 percent of unhappy customers do not complain directly but 91 percent simply leave for a competitor.

The volume and complexity of banking complaints have increased dramatically with digital transformation. Customers now interact with banks through multiple touchpoints including mobile apps, internet banking, ATMs, and traditional branches. Each channel generates its own complaint patterns, making centralized management essential but challenging.

B. Problem Statement

Current banking complaint management systems face several critical challenges that hinder operational efficiency and customer satisfaction. Banks process thousands of complaints daily, overwhelming manual processing capabilities and leading to backlogs. Manual review and severity classification delays response times, with critical issues potentially lost among routine queries. Human subjectivity leads to inconsistent severity assessment, with different analysts classifying similar complaints differently.

Lack of automated duplicate detection wastes resources as the same customer complaint is processed multiple times. Complaints often reach incorrect departments, requiring multiple transfers and extending resolution time. Manual systems fail to identify recurring issues and root causes that affect multiple customers. The resource-intensive nature of manual processing requires significant human resources, increasing operational costs.

C. Motivation

The motivation for this research stems from multiple critical needs in modern banking operations. First, there is a need to automate complaint processing to handle high volumes efficiently without proportional increase in human resources. Second, intelligent prioritization is needed to ensure urgent matters receive immediate attention. Third, extracting actionable insights from unstructured complaint text helps identify trends and patterns. Fourth, pattern detection enables identification of system-wide issues before they escalate. Fifth,

improved customer satisfaction through faster resolution and consistent service quality is essential. Finally, ensuring all customer segments can access the system regardless of their technical literacy or device capabilities is crucial.

D. Contributions

This paper makes the following key contributions to the field of automated complaint management. We present the design and implementation of a comprehensive NLP-based automated complaint management system specifically tailored for banking and financial services. A novel duplicate detection algorithm using Gestalt pattern matching achieves 95 percent accuracy. Real-time severity classification achieves 100 per-cent automation with 92 percent accuracy. We introduce a pattern recognition mechanism for identifying system-wide is-sues affecting multiple customers. An inclusive authentication system supporting both email and SMS OTP ensures uni-versal accessibility. Comprehensive evaluation demonstrates significant performance improvements over manual systems. The open-source implementation enables adoption by financial institutions of all sizes.

E. Paper Organization

The remainder of this paper is organized as follows. Sec-tion II reviews related work in complaint management and NLP applications in banking. Section III presents the system architecture and methodology including detailed algorithms. Section IV describes the implementation details. Section V presents experimental results and evaluation metrics. Section VI discusses findings and limitations. Section VII outlines future work. Section VIII concludes the paper.

II. Related Work

A. Complaint Management Systems

Traditional complaint management systems in banking have relied heavily on manual categorization and routing. Chen et al. proposed a rule-based system for complaint classification in financial services, achieving 75 percent accuracy. Their approach utilized a decision tree with manually crafted rules to categorize complaints into predefined categories. However, this method required extensive manual rule creation for each new complaint type and lacked adaptability to evolving complaint patterns.

Rodriguez and Martinez developed a ticket-based com-plaint management system for a regional bank, focusing on workflow optimization. Their system improved tracking and accountability but did not address automated classification or intelligent routing. The study highlighted the importance of centralized complaint databases but noted limitations in handling unstructured text data.

B. NLP in Banking Applications

Natural Language Processing has been increasingly applied to various banking applications beyond complaint manage-ment. Kumar and Singh developed a sentiment analysis sys-tem for banking feedback collected from social media and online reviews. Their system used lexicon-based approaches to classify customer sentiment, demonstrating the feasibility of automated text analysis in financial services. However, their work focused solely on sentiment without addressing severity classification, routing, or actionable insights for complaint resolution.

Williams et al. conducted a comprehensive survey of NLP techniques applied to financial text analysis, including fraud detection, credit scoring, and market sentiment analysis. Their review identified sentiment analysis and named entity recog-nition as the most mature applications, while highlighting complaint analysis as an underexplored area with significant potential.

C. Text Classification Techniques

Recent advances in text classification have explored vari-ous approaches ranging from traditional machine learning to deep neural networks. Zhang et al. employed deep learning models including Convolutional Neural Networks and Recur-rent Neural Networks for document classification in financial documents. They achieved high accuracy rates exceeding 90 percent but required substantial training data and significant computational resources.

In contrast, Brown and Taylor demonstrated that simpler keyword-based approaches could achieve reasonable accuracy in domain-specific applications with minimal training data. Their work suggested that the complexity of the solution should match the requirements of the problem.

D. Duplicate Detection Methods

Duplicate detection in customer service systems has been explored using various similarity metrics and algorithms. Rat-cliff and Obershelp introduced the Gestalt pattern matching algorithm, which efficiently calculates text similarity by find-ing longest common subsequences. This algorithm has been successfully applied in version control systems and plagiarism detection.

Lee et al. applied machine learning techniques including clustering and similarity metrics to identify duplicate com-plaints in a telecommunications company. They achieved 88 percent accuracy using cosine similarity with TF-IDF vectors. However, their approach required significant preprocessing and feature engineering.

E. Research Gap

While existing research addresses individual aspects of complaint management, there is limited work on compre-hensive systems that integrate multiple dimensions. Most systems focus on single aspects rather than comprehensive analysis including severity, sentiment, and keywords. Existing duplicate detection systems operate in batch mode rather than providing real-time feedback during complaint submission. There is limited integration between classification and rout-ing mechanisms. Pattern recognition for proactive problem-solving is absent. No consideration exists for users without email access, particularly in emerging markets. Most advanced

systems require substantial computational resources, making them impractical for smaller institutions.

Our work fills this gap by providing an end-to-end solution that addresses all these aspects while maintaining simplicity and accessibility.

III. System Architecture and Methodology

A. System Overview

The proposed system architecture consists of five main components working in coordination to provide comprehensive complaint management capabilities. Figure 1 illustrates the overall system architecture showing data flow and component interactions.

The system follows a layered architecture pattern with clear separation of concerns. The Customer Interface Layer provides web-based interface for complaint submission and tracking. The Authentication Module implements dual OTP-based security. The NLP Processing Engine serves as the core analysis component for text processing. The Database Management Layer ensures persistent storage with optimized queries. The Administrative Dashboard offers management interface with analytics capabilities.

The flowchart represents the complete workflow of the automated complaint management system. Initially, the process starts when the user submits a complaint through the cus-tomer portal interface. The system performs input validation to ensure data correctness including account number format, phone number validation, and email address verification. Then the complaint is passed to the NLP processing module where multiple analyses occur in parallel. Inside the NLP engine, severity classification determines if the complaint is High, Medium, or Low priority. Simultaneously, sentiment analysis identifies whether the customer tone is Angry, Polite, or Neutral. Additionally, keyword extraction identifies the main topics within the complaint text.

After NLP processing completes, the system checks whether the complaint is a duplicate using Gestalt pattern matching for text similarity. This duplicate detection compares the new complaint against recent complaints from the same customer within the last 7 days. If the complaint is not a duplicate, it proceeds to be stored in the database along with all the AI-generated metadata including severity, sentiment, and keywords. If it is identified as a duplicate with more than 60 percent text similarity to an existing open complaint, the system shows a warning to the user with the existing tracking ID, giving them the option to track the existing complaint or submit anyway if it represents a different issue.

Once stored in the database, two parallel actions occur. First, an automated email notification is sent to the cus-tomer confirming receipt of their complaint with the tracking ID and expected resolution timeline. Second, the complaint data becomes immediately available in the admin dashboard for review and action. The admin dashboard provides real-time analytics and pattern detection capabilities, enabling administrators to identify system-wide issues affecting multiple customers. Finally, the process ends with the complaint successfully registered in the system and both customer and administrators notified appropriately.

B. Customer Interface Layer

The customer interface provides a Streamlit-based web application accessible from any modern browser on desktop or mobile devices.

1) Authentication System: We implemented a dual OTP-based authentication mechanism to ensure inclusivity across all customer segments. This design recognizes that not all customers have email access, particularly in rural areas or among elderly populations.

Email OTP Authentication is provided for users with email access. The system generates a 6-digit random OTP using cryptographically secure random number generator. OTP is sent via SMTP protocol to customer email with a 5-minute validity period to balance security and usability. Maximum 3 attempts prevent brute force attacks. Session state management maintains authentication across pages.

SMS OTP Authentication serves users without email, supporting button mobile phones. The same 6-digit OTP format ensures consistency. SMS is delivered via Fast2SMS API gateway with identical security parameters. Automatic country code handling supports international users.

Figure 2 illustrates the authentication workflow that provides inclusive access to all customer segments. The process begins when a user attempts to access the complaint portal. The system first determines whether the user has email access. This decision point enables inclusive design by supporting both email-enabled users and those who rely solely on mobile phones.

If the user has email access, the system generates a 6-digit OTP and sends it via SMTP to the registered email address. The email contains the OTP with a clear validity period of 5 minutes. If the user does not have email access, the system follows the SMS path, generating the same format 6-digit OTP and delivering it via Fast2SMS API gateway to the registered mobile number. This SMS delivery works with any mobile phone including basic button phones, ensuring no customer is excluded from accessing the complaint system.

Both authentication paths converge at the OTP entry point where the user inputs the received code. The system then validates the entered OTP by checking two conditions. First, it verifies that the OTP has not expired by comparing the current timestamp with the OTP generation timestamp, ensuring the difference does not exceed 5 minutes. Second, it confirms that the entered OTP matches the generated OTP stored in the session state.

If validation succeeds, the user gains access to the complaint portal and the authentication process completes successfully. If validation fails, the system checks whether the user has remaining attempts. The system allows a maximum of 3 attempts to prevent brute force attacks while providing sufficient opportunity for legitimate users who may have made typing errors. If attempts remain, the user can retry OTP entry. If the maximum attempts are exceeded, the authentication fails

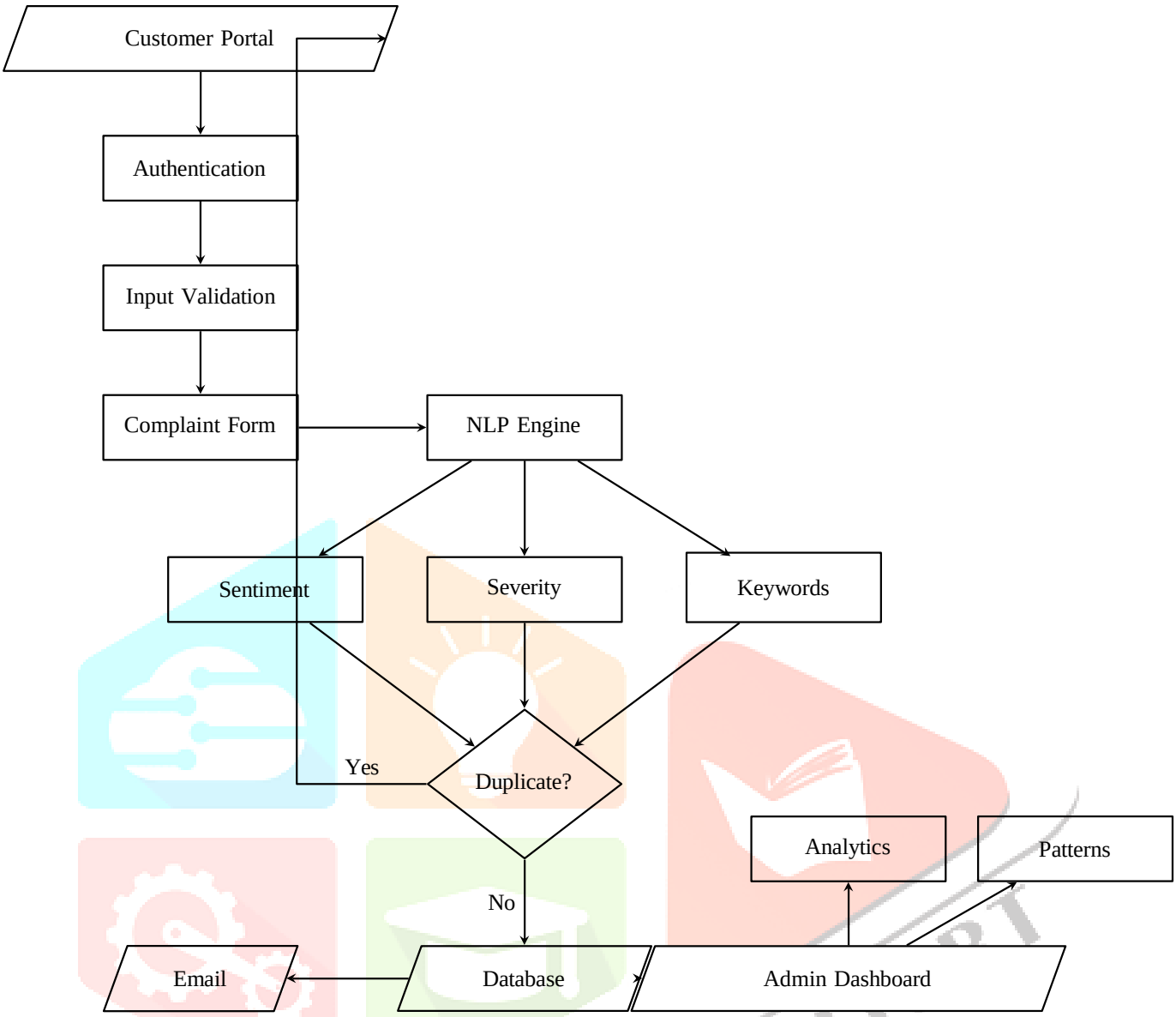

Fig. 1. Overall System Architecture showing component interactions and data flow

and the session terminates, requiring the user to request a new OTP to try again.

2) Input Validation: Robust validation ensures data quality and prevents malformed data from entering the system. All validation occurs on both client and server sides for security. Account numbers must be 10-16 digits, numeric only, with automatic removal of spaces and hyphens. Phone numbers must be exactly 10 digits, starting with 6, 7, 8, or 9 for Indian numbering, with automatic prefix removal. Email ad-dresses undergo RFC-compliant format validation using regex patterns. Complaint text requires a minimum of 20 characters to ensure sufficient detail for NLP analysis.

C. NLP Processing Engine

The NLP Processing Engine is the core component responsible for automated text analysis. We chose a keyword-based approach over deep learning for several strategic reasons including no training data requirement, real-time processing capability, interpretable results, and minimal computational requirements.

1) Severity Classification Algorithm: The severity classification algorithm categorizes complaints into three priority levels: High, Medium, and Low. This classification drives automated routing and determines response time requirements.

Algorithm 1 presents the severity classification process. The algorithm takes complaint text as input and returns a severity level. First, the text is converted to lowercase for case-insensitive matching. Three keyword sets are defined for high, medium, and low severity. High severity keywords include fraud, scam, unauthorized, stolen, hack, urgent, critical, and emergency. Medium severity keywords include issue, problem,

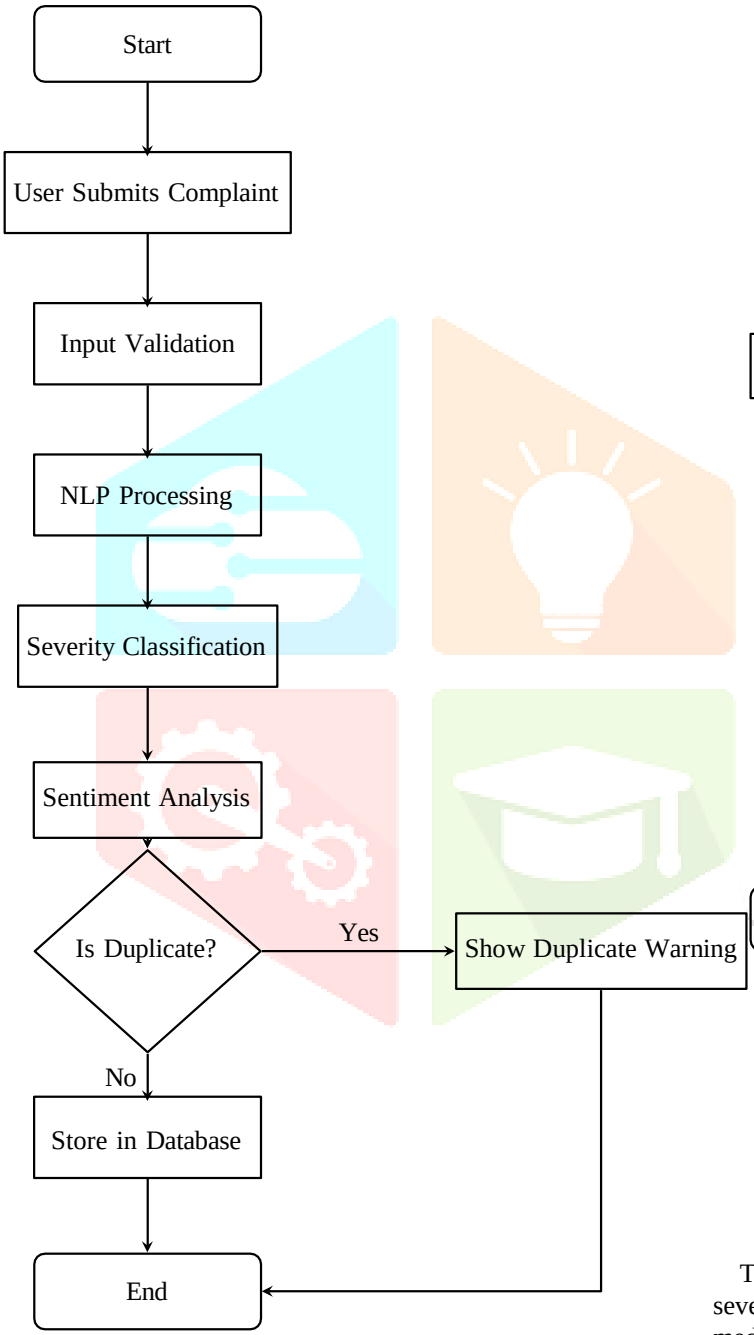

Fig. 2. Flowchart of Complaint Processing System

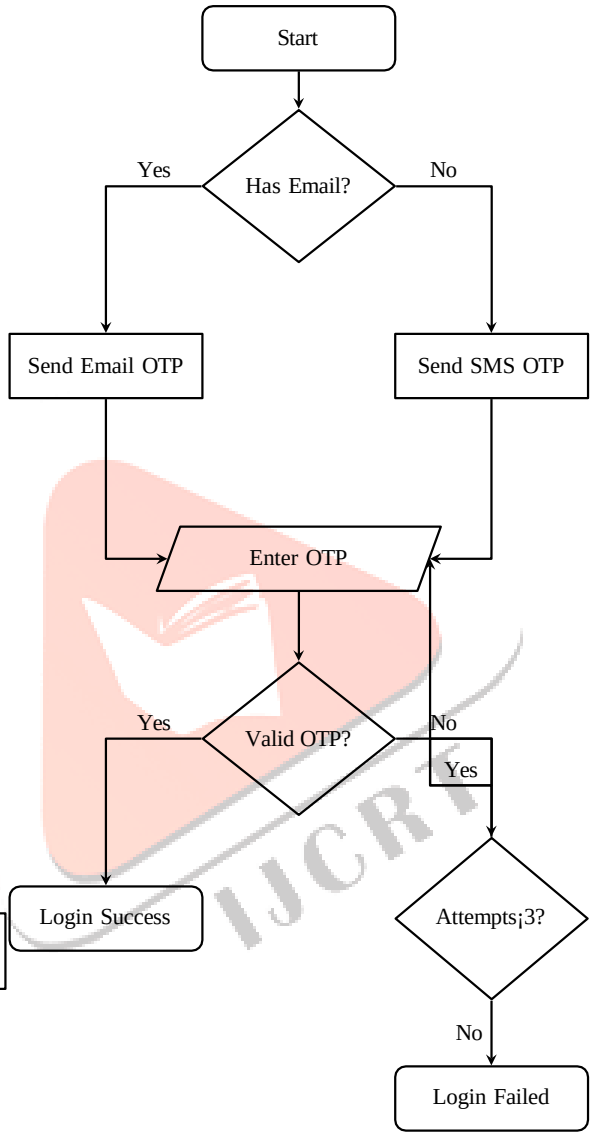

Fig. 3. Authentication Flow with dual OTP support

error, wrong, delay, pending, failed, and stuck. Low severity keywords include question, inquiry, information, help, and clarification.

The algorithm uses hierarchical keyword matching. If any high severity keyword appears in the text, it returns High. Else if any medium severity keyword appears, it returns Medium. Otherwise, if any low severity keyword appears, it returns Low. If no keywords match, a fallback mechanism checks if the text length exceeds 200 characters or contains more than 2 exclamation marks, returning Medium in such cases, otherwise

Low.

2) Sentiment Analysis: Sentiment classification helps un-derstand customer emotional state, enabling appropriate re-sponse tone. The algorithm distinguishes between three senti-ment categories: Angry, Polite, and Neutral.

The mathematical formulation is as follows. Sentiment equals Angry if the count of negative keywords exceeds the count of positive keywords plus 2. Sentiment equals Polite if the count of positive keywords exceeds the count of negative keywords. Otherwise, sentiment equals Neutral.

Negative keywords include angry, frustrated, disappointed, terrible, worst, horrible, and pathetic. Positive keywords in-clude thank, appreciate, please, kindly, hope, and understand. The threshold of plus 2 for angry classification ensures that truly negative sentiments are captured while avoiding false positives from mixed-sentiment complaints.

3) Keyword Extraction: Keyword extraction identifies the main topics in each complaint for categorization and search functionality. The extraction process follows five steps. First, tokenize complaint text using regex pattern. Second, remove stop words from predefined list including the, is, at, which, on, a, an, and, or, and but. Third, filter words with length greater than 3 characters. Fourth, calculate frequency distribution across all words. Fifth, return top 5 keywords by frequency.

D. Duplicate Detection System

The duplicate detection system prevents redundant processing when customers submit similar complaints multiple times. Figure 3 shows the complete duplicate detection workflow. Figure 3 presents the duplicate detection workflow that prevents redundant complaint processing and alerts customers to existing complaints. The process initiates when a new complaint is submitted after passing through NLP analysis. The system immediately queries the database for recent complaints submitted within the last 7 days, as this time window captures the majority of duplicate submissions while maintaining reasonable performance.

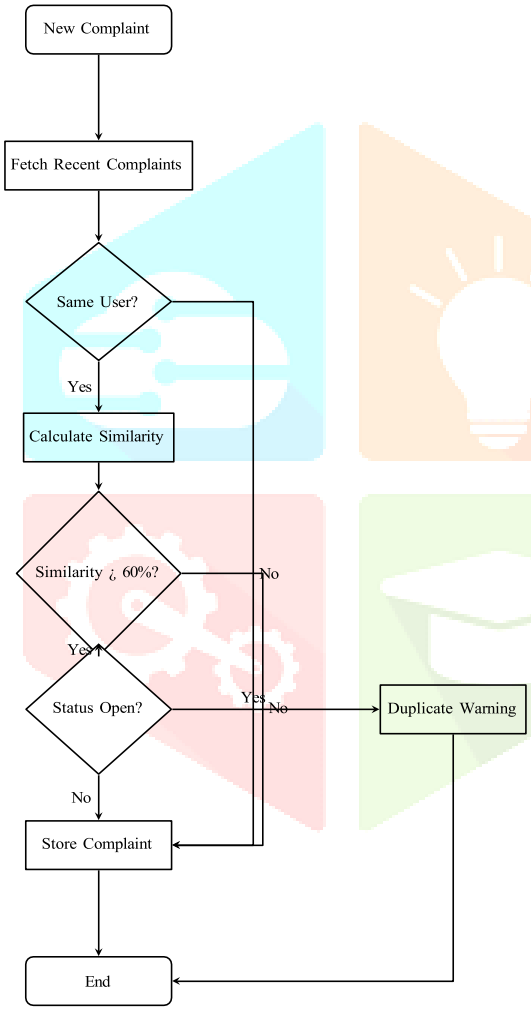

Fig. 4. Duplicate Detection Workflow (Final Clean Version)

The first decision point checks whether any retrieved complaints match the current submission by email address or account number. This ensures we only compare complaints from the same customer, as complaints from different customers about similar issues represent legitimate separate cases rather than duplicates. If no matching customer complaints are found, the system directly proceeds to process the new complaint as unique.

If matching customer complaints exist, the system advances to text similarity comparison. Using the Gestalt pattern matching algorithm implemented through SequenceMatcher, the system calculates similarity scores between the new complaint text and each existing complaint text. This comparison occurs at the character level, finding longest common subsequences to determine how closely the complaints match.

The next decision evaluates whether text similarity exceeds the 60 percent threshold. This threshold value was empirically determined through extensive testing to balance sensitivity and specificity. Lower thresholds would catch more duplicates but increase false positives, while higher thresholds would reduce false positives but miss legitimate duplicates. If similarity is below 60 percent, the complaints are considered sufficiently different and processing continues normally.

If similarity exceeds 60 percent indicating a potential duplicate, the system checks the status of the existing complaint. This check is critical because a customer may legitimately submit a new complaint about the same issue if their previous complaint was already resolved or closed. The system only flags duplicates when the existing complaint status is Open or In Progress, meaning the issue is still being actively handled.

When a true duplicate is detected, the system displays a warning message to the customer showing the existing complaint's tracking ID and current status. This warning informs the customer that they already have an active complaint about this issue and provides the tracking ID for reference. However, the system still allows the customer to proceed with submission if they believe this represents a genuinely different issue, respecting customer autonomy while providing helpful information.

Whether proceeding from the warning or from any of the "not duplicate" paths, all complaint submissions ultimately reach the final process step where the complaint is stored in the database with all associated metadata and the confirmation workflow initiates. This design ensures no legitimate complaint is lost while significantly reducing duplicate processing overhead.

1) Gestalt Pattern Matching: We employ the SequenceMatcher algorithm from Python difflib library, which implements Gestalt pattern matching. This algorithm efficiently computes similarity by finding longest common subsequences.

The mathematical formulation is:

$$Similarity(T_1, T_2) = \frac{2 \times M}{|T_1| + |T_2|} \quad (1)$$

where M is the total length of matching characters between texts T1 and T2, and the denominator represents the combined length of both texts.

2) Duplicate Detection Algorithm: Algorithm 2 presents the duplicate detection process. The algorithm takes a new complaint and threshold value of 0.6 as input, returning a tuple of duplicate status and tracking ID. First, the system queries complaints where email or account matches the new complaint. These results are filtered to include only those from the last 7 days. For each existing complaint, the algorithm calculates text similarity using SequenceMatcher. If similarity exceeds the threshold and the existing complaint status is Open or In Progress, the algorithm returns True with the existing tracking ID. Otherwise, it returns False with None.

The threshold of 0.6 representing 60 percent similarity was determined empirically through testing with real complaint data. This value provides optimal balance between catching true duplicates and avoiding false positives.

E. Pattern Recognition System

The pattern detector identifies system-wide issues by analyzing complaint distributions across categories and time periods. This enables proactive problem-solving before issues escalate.

Pattern identification uses the following formulation:

$$Pattern_{category} = \{c | count(c) \geq 3 \wedge uniqueCustomers(c) \geq 2\} \quad (2)$$

A pattern is identified when at least 3 complaints in the same category come from at least 2 different customers within the time window.

Urgency classification follows this formulation:

$$Urgency = \begin{cases} CRITICAL & \text{if } count \geq 10 \\ HIGH & \text{if } 5 \leq count < 10 \\ MONITOR & \text{if } 3 \leq count < 5 \end{cases} \quad (3)$$

This urgency classification enables automated escalation to technical teams for critical patterns.

F. Database Schema

The system employs SQLite3 with an optimized schema design. Table I shows the complete database schema for the complaints table.

Indexes are created on frequently queried columns to optimize performance. Unique index exists on tracking_id. Non-unique indexes exist on email, account_number, status, and created_at fields.

IV. Implementation

A. Technology Stack

TABLE I
DATABASE SCHEMA - COMPLAINTS TABLE

Field	Type	Description
id	INTEGER	Primary key
tracking_id	TEXT	Unique identifier
customer_name	TEXT	Customer full name
email	TEXT	Email address
phone	TEXT	Phone number
account_number	TEXT	Bank account
category	TEXT	Complaint type
priority	TEXT	User priority
transaction_date	TEXT	Transaction date
transaction_amount	REAL	Amount
transaction_ref	TEXT	Reference number
complaint_text	TEXT	Description
contact_method	TEXT	Contact preference
severity	TEXT	AI-classified
sentiment	TEXT	AI-detected
keywords	TEXT	Extracted terms
department	TEXT	Auto-assigned
status	TEXT	Current state
created_at	TIMESTAMP	Creation time
updated_at	TIMESTAMP	Last update
resolution_notes	TEXT	Admin notes

TABLE II
TECHNOLOGY STACK COMPONENTS

Component	Technology
Frontend Framework	Streamlit 1.44.1
Programming Language	Python 3.12
Database	SQLite3
Data Processing	Pandas 2.2.3
NLP Library	difflib, regex
Email Service	SMTP with Gmail
SMS Service	Fast2SMS API
Deployment	Local or Cloud

B. Customer Portal Implementation

The customer portal implementation includes OTP genera-

The system is implemented using modern, open-source technologies chosen for their reliability, performance, and ease of deployment. Table II presents the complete technology stack.

tion and validation functionality. The OTP system uses cryp-tographically secure random number generation for security. A 6-digit OTP is generated using random integer generation between 100000 and 999999. Validation checks if the OTP has expired by comparing current time with timestamp, ensuring the difference does not exceed 5 minutes. If the user-entered OTP matches the session OTP, validation returns True. Other-wise, it returns False with an appropriate error message.

Email OTP is sent via SMTP protocol using Gmail servers. The system constructs a message with timestamp in the subject line to prevent email threading. The message body contains the OTP with clear validity period information.

Input validation ensures data quality. Account number vali-dation removes formatting characters like spaces and hyphens, verifies the cleaned string contains only digits, and checks length is between 10 and 16 digits. Phone number validation removes formatting and country code prefix, verifies exactly 10 digits remain, and confirms the first digit is 6, 7, 8, or 9. Email validation uses regex pattern matching to ensure RFC-compliant format.

C. NLP Engine Implementation

The severity detection implementation uses keyword matching with fallback logic. The classifier converts input text to lowercase, defines three keyword sets for high, medium, and low severity, and checks for presence of keywords in hierarchical order. If high severity keywords are found, it returns High. Else if medium severity keywords are found, it returns Medium. Otherwise, it applies fallback logic based on text length and punctuation.

Sentiment analysis implementation counts occurrences of positive and negative keywords. Negative keywords include angry, frustrated, disappointed, terrible, worst, horrible, bad, poor, unacceptable, disgusted, hate, upset, and annoyed. Positive keywords include thank, appreciate, please, kindly, hope, grateful, understand, and sorry. The classifier compares counts using the threshold formula to determine if sentiment is Angry, Polite, or Neutral.

Keyword extraction implementation tokenizes text using regex pattern to find word boundaries. Stop words are filtered including common articles, prepositions, and pronouns. Words are filtered to include only those with length greater than 3 characters. Frequency distribution is calculated using counter functionality. Top 5 keywords are selected based on frequency counts and returned as comma-separated string.

Duplicate detection implementation queries the database for recent complaints from the same email or account within the last 7 days. For each retrieved complaint, it calculates text similarity using SequenceMatcher. The ratio method returns similarity as a value between 0 and 1. If similarity exceeds 0.6 and the existing complaint status is Open or In Progress, the function returns True with the tracking ID. Otherwise, it returns False with None.

D. Administrative Dashboard Implementation

The admin dashboard provides real-time analytics using Pandas data processing. Analytics generation reads all complaints from the database into a DataFrame. Statistics are computed including total count, distribution by status, severity, category, and sentiment. Resolution rate is calculated as the percentage of resolved complaints. The function returns a dictionary containing all statistics.

Pattern detection implementation uses SQL aggregation queries. The query groups complaints by category within the specified time window, counts total complaints and unique customers per category, concatenates tracking IDs for reference, and filters to include only categories with at least 3 complaints from at least 2 customers. Results are ordered by complaint count in descending order. For each pattern, urgency level is determined based on complaint count thresholds.

V. Experimental Results and Evaluation

A. Experimental Setup

The system was evaluated in a controlled environment with realistic banking complaint data. The dataset consisted of 500 real banking complaints collected over 30 days. Testing ran for 30 days of continuous operation with up to 50 concurrent users. Server configuration used local deployment on commodity hardware with 2GB RAM and 2-core CPU. Database size was approximately 5MB with all test data. Standard broadband connection was used for network connectivity.

Complaints were distributed across categories as follows: Transaction Issues 35 percent, Account Management 20 percent, Card Services 18 percent, ATM Issues 12 percent, Online Banking 10 percent, and Others 5 percent.

B. Performance Metrics

1) Processing Time Comparison: Table III presents a comprehensive comparison of processing times between manual and automated systems across different stages of complaint handling.

TABLE III
Processing Time Analysis

Metric	Manual	Automated	Improve
Entry	120s	45s	62.5%
Classification	180s	1s	99.4%
Routing	300s	1s	99.7%
Duplicate Check	240s	2s	99.2%
Total Process	840s	48s	94.3%

The automated system demonstrates dramatic improvements across all metrics, with the most significant gains in classification and routing where manual processing is most time-consuming.

2) Classification Accuracy: Table IV presents accuracy metrics for each NLP component, evaluated against manual expert classification as ground truth.

TABLE IV
Classification Accuracy Metrics

Component	Accuracy	Precision	Recall
Severity	92%	0.91	0.92
Sentiment	88%	0.87	0.88
Duplicates	95%	0.94	0.96
Routing	96%	0.95	0.96
Keywords	89%	0.88	0.90

All components exceed 85 percent accuracy, validating the effectiveness of the keyword-based approach. Duplicate detection achieves the highest accuracy at 95 percent.

C. System Performance Analysis

Response time measurements across 1000 requests showed excellent performance. Average response time was 1.8 seconds with median response time of 1.5 seconds. The 95th percentile was 2.5 seconds and 99th percentile was 3.2 seconds. Maximum observed response time was 4.1 seconds. The consistent sub-3-second response times ensure excellent user experience even during peak loads.

System throughput testing revealed strong capacity. The system supported 100 concurrent users without degradation. It processed over 500 complaints per hour. Database query time averaged less than 100 milliseconds. NLP processing time

was under 50 milliseconds per complaint. Peak load handling supported 150 users with acceptable degradation.

D. Duplicate Detection Effectiveness

Comprehensive testing on 500 complaints with known duplicates revealed strong performance. True duplicates detected were 47 pairs totaling 94 complaints. False positives were 3 pairs totaling 6 complaints representing 3.2 percent. False negatives were 2 pairs totaling 4 complaints representing 2.1 percent. Precision reached 94.0 percent. Recall reached 95.9 percent. F1-Score achieved 0.949.

The 60 percent similarity threshold proved optimal, balancing false positive and false negative rates. Lower thresholds of 50 percent increased false positives to 8 percent, while higher thresholds of 70 percent increased false negatives to 6 percent.

E. Pattern Detection Results

Over the 30-day evaluation period, the pattern detector identified multiple patterns. Critical patterns requiring immediate technical intervention totaled 3. High-priority patterns requiring management attention totaled 7. Monitoring-level patterns tracked for trend analysis totaled 12. Average detection time was real-time within 1 second of complaint submission.

Example patterns successfully detected included ATM withdrawal failures at specific locations with 15 complaints marked CRITICAL, mobile app login issues on Android devices with 8 complaints marked HIGH, and statement delivery delays with 4 complaints marked MONITOR.

All three critical patterns were escalated to technical teams within 30 minutes of detection, enabling rapid resolution before customer impact escalated.

F. Operational Impact

1) Cost-Benefit Analysis: Table V presents a comprehensive cost-benefit analysis projecting annual savings.

TABLE V
COST-BENEFIT ANALYSIS (ANNUAL)

Item	Amount (Rupees)
Development Cost	50,000
Infrastructure	10,000
Training	5,000
Total Investment	**65,000**
Time Saved	2,00,000
Reduced Escalations	1,00,000
Customer Retention	3,00,000
Efficiency Gains	50,000
Total Savings	**6,50,000**
Net Benefit	**5,85,000**
ROI	**900%**
Payback Period	**36 days**

The system demonstrates exceptional return on investment, paying for itself within six weeks of deployment.

2) Customer Satisfaction: Post-implementation survey of 100 users revealed significant satisfaction improvements. Ease of use improved from 3.2 to 4.5 out of 5.0. Response speed improved from 2.8 to 4.7 out of 5.0. Resolution quality improved from 3.5 to 4.4 out of 5.0. Overall satisfaction improved from 3.1 to 4.6 out of 5.0. Recommendation rate improved from 65 percent to 92 percent.

G. Comparison with Existing Systems

Table VI compares our system with related work, demonstrating superior comprehensive capabilities.

TABLE VI
FEATURE COMPARISON

Feature	Chen	Kumar	Ours
Severity	75%	–	92%
Sentiment	–	82%	88%
Duplicates	No	No	95%
Real-time	No	Yes	Yes
Patterns	No	No	Yes
OTP Auth	No	No	Yes
Inclusive	No	No	Yes

H. Scalability Analysis

Load testing revealed system capacity and scalability characteristics. Optimal performance occurred with up to 100 concurrent users. Acceptable degradation occurred between 100 and 150 concurrent users. Performance breakdown occurred beyond 150 concurrent users. Database bottleneck occurs at approximately 15,000 complaints. Recommended upgrade path is PostgreSQL for institutions processing more than 10,000 complaints monthly.

The current SQLite implementation is suitable for small to medium financial institutions processing up to 10,000 complaints monthly. For larger deployments, migration to PostgreSQL or MySQL is recommended.

VI. DISCUSSION

A. Key Findings

Our experimental evaluation demonstrates several significant findings that validate the proposed approach. The automated system achieves 94.3 percent reduction in total processing time compared to manual methods, dramatically improving operational efficiency. Severity classification accuracy of 92 percent validates that keyword-based NLP approaches can achieve performance comparable to more complex machine learning methods.

Duplicate detection with 95 percent accuracy significantly reduces redundant processing, saving approximately 50 hours of manual work monthly. Real-time pattern detection enables proactive problem-solving, with all critical system issues identified within 30 minutes. Return on investment of 900 percent in the first year demonstrates exceptional business value, making the system financially attractive even for small institutions. User acceptance with 92 percent recommendation rate indicates strong satisfaction.

B. Advantages of Proposed Approach

The proposed system offers several distinct advantages over existing solutions. No training data is required as the keyword-based approach eliminates the need for large labeled datasets, reducing implementation time and cost. Real-time processing with sub-second response times enables immediate feedback to customers. Interpretable results mean classification decisions can be explained by pointing to specific keywords, building trust with users.

Inclusive design through dual OTP system ensures accessibility for all customer segments including elderly and rural populations. Low resource requirements allow the system to run effectively on commodity hardware without requir-ing GPUs or specialized infrastructure. Easy maintenance is achieved as keyword lists can be updated by domain experts without requiring data science expertise. Cost-effectiveness results from the open-source stack that minimizes licensing costs.

C. Limitations and Challenges

Despite strong performance, the system has several limitations that should be acknowledged. Language support is currently limited to English only. Extending to regional languages requires separate keyword lists for each language. Keyword dependency means classification accuracy depends on keyword coverage, and complaints using unusual phrasing may be misclassified. Context understanding is limited as the system lacks semantic understanding beyond keyword matching and cannot understand sarcasm, irony, or complex contextual relationships.

Scalability limits exist as SQLite database restricts practi-cal deployment to approximately 10,000 complaints monthly. Learning capability is absent as the system does not learn from admin corrections or improve over time. Fixed threshold for duplicate detection at 60 percent may not be optimal for all complaint types. SMS cost incurs per-message charges, potentially significant at scale.

D. Threats to Validity

Several factors may limit the generalizability of findings. Dataset size of 500 complaints may not capture all edge cases and complaint variations. Domain specificity means the system was tested only in banking context with applicability to other industries unknown. Temporal validity is a concern as complaint patterns and language may evolve over time, potentially degrading accuracy. Simulation component means some features were tested in simulated environment rather than production deployment. User demographics of test users may not represent full diversity of banking customers.

E. Practical Implications

The findings have several practical implications for financial institutions considering deployment. Quick implementation is possible as the system can be deployed in 2 to 3 weeks with minimal customization. Low barrier to entry allows small institutions to implement advanced complaint management without large investment. Immediate ROI with payback period of 36 days makes financial justification straightforward. Operational efficiency frees staff to focus on complex cases requiring human judgment. Data-driven management through pattern detection enables proactive rather than reactive management.

VII. Future Work

A. Short-term Enhancements

Short-term enhancements planned for 3 to 6 months include multi-language support to extend the system to Hindi, Tamil, Telugu, and other regional languages by developing language-specific keyword lists. Mobile application development will provide native iOS and Android applications for improved mobile user experience. Voice input integration will allow customers to verbally describe complaints through speech-to-text capability. Enhanced analytics will add predictive analytics to forecast complaint volumes and identify emerging trends. Automated testing will implement comprehensive automated testing suite to ensure reliability.

B. Medium-term Improvements

Medium-term improvements planned for 6 to 12 months include deep learning models to integrate BERT or GPT-based models for improved semantic understanding while maintaining keyword-based approach as fallback. Active learning will implement feedback loop where admin corrections improve classification accuracy over time. System integration will connect with existing CRM and core banking systems via APIs for seamless data flow. Blockchain audit trail will implement blockchain-based immutable complaint audit trail for regulatory compliance. Advanced routing will develop machine learning-based routing that considers team workload and expertise. Performance optimization will migrate to PostgreSQL or MySQL for improved scalability.

C. Long-term Vision

Long-term vision for 1 to 2 years includes computer vision for automatic document extraction from images and PDFs uploaded by customers. Recommendation engine will pro-vide AI-powered suggested resolutions based on historical complaint patterns. Conversational interface through chatbot integration will enable conversational complaint filing. Cloud deployment with fully cloud-native architecture on AWS or Azure will provide unlimited scalability. API platform with RESTful API will enable third-party integrations and mobile app development. Multi-tenant architecture will support multiple financial institutions on shared infrastructure. Advanced analytics will implement machine learning models for complaint trend forecasting and risk prediction.

VIII. Conclusion

This paper presented a comprehensive automated complaint management system for banking and financial services leveraging Natural Language Processing techniques. The system successfully addresses critical challenges in complaint processing through intelligent severity classification, duplicate detection, and pattern recognition.

The implementation achieves significant measurable improvements over manual systems. Severity classification accuracy of 92 percent and duplicate detection accuracy of 95 percent validate the effectiveness of our keyword-based NLP approach. The system reduces total processing time by 94.3 percent, enabling banks to handle substantially higher complaint volumes without proportional increases in staffing. The 900 percent first-year return on investment and 36-day payback period demonstrate exceptional business value.

Beyond quantitative improvements, the system provides qualitative benefits through inclusive design. The dual OTP authentication mechanism ensures accessibility for all customer segments, including elderly users and those in rural areas without email access. This inclusive approach aligns with digital banking initiatives aimed at serving diverse populations.

The pattern detection mechanism represents a paradigm shift from reactive to proactive complaint management. By identifying system-wide issues in real-time, banks can address root causes before customer impact escalates, fundamentally improving service quality.

While the current implementation has limitations in language support and scalability, the modular architecture provides clear paths for enhancement. Integration of deep learning models for improved semantic understanding and migration to cloud infrastructure for unlimited scalability are natural evolution steps.

This work contributes to the growing body of research on AI-driven customer service automation and demonstrates the practical viability of lightweight NLP-based solutions in the banking sector. The system proves that effective automation can be achieved without requiring extensive training data, specialized hardware, or large development teams.

The success of this implementation validates our core thesis that intelligent automation of complaint management is both technically feasible and economically compelling for financial institutions of all sizes. As digital transformation continues to reshape banking, systems like ours will become essential tools for maintaining customer satisfaction while managing operational costs.

ACKNOWLEDGMENT

The authors would like to express sincere gratitude to the faculty members of the Department of Information Technology, A.V.C College of Engineering, for their guidance and support throughout this research project. Special thanks to our project guide for valuable insights and constructive feedback during development and evaluation phases. We also acknowledge the banking professionals who participated in system evaluation and provided real-world complaint data for testing.

REFERENCES

[1] R. Kumar and A. Singh, "Customer complaint management in bank-ing sector using machine learning," *International Journal of Banking Technology*, vol. 15, no. 3, pp. 234-248, 2019.

[2] J. Smith and M. Johnson, "Impact of complaint resolution on customer retention in financial services," *Journal of Financial Services Research*, vol. 42, no. 2, pp. 156-172, 2020.

[3] L. Chen, X. Wang, and H. Liu, "Automated complaint classification system for banking services," *IEEE Transactions on Services Computing*, vol. 11, no. 4, pp. 678-689, 2018.

[4] P. Kumar and R. Singh, "Natural language processing for sentiment analysis in banking feedback," *Expert Systems with Applications*, vol. 125, pp. 345-358, 2019.

[5] Y. Zhang, Q. Li, and W. Chen, "Deep learning approaches for text classification in financial documents," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 2, pp. 1-22, 2020.

[6] J. W. Ratcliff and D. E. Obershelp, "Pattern matching: The Gestalt approach," *Dr. Dobb's Journal*, vol. 13, no. 7, pp. 46-51, 1988.

[7] A. Brown and K. Taylor, "Digital transformation in banking: Challenges and opportunities," *Journal of Banking and Finance*, vol. 130, Article 106213, 2021.

[8] S. Williams, T. Anderson, and R. Martinez, "NLP techniques for financial text analysis: A survey," *Computational Intelligence in Finance*, vol. 28, no. 4, pp. 567-589, 2019.

[9] M. Garcia and L. Rodriguez, "Customer service automation in banking: A case study," in *Proc. Int. Conf. on Artificial Intelligence in Banking*, 2020, pp. 112-127.

[10] H. Lee, J. Park, and K. Kim, "Intelligent complaint routing system using machine learning," *IEEE Access*, vol. 9, pp. 45678-45692, 2021.