International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-14, Issue-1, January 2026
DOI: https://doi.org/10.55524/ijircst.2026.14.1.7
Article ID IRP-1706, Pages 55-62
www.ijircst.org

AI-Powered Anonymous Crime Reporting System with Automated Risk Assessment

Shilpa R[1], Abhisheka C Hegde[2], Anushree Joshi[3], Chinmay Naik[4], Darshan Hegde[5]

[1]Assistant Professor, Department of Computer Science & Engineering, Sri Dharmasthala Manjunatheshwara Institute of Technology, Ujire, Karnataka, India

[2, 3, 4, 5] B.E Scholar, Department of Computer Science & Engineering, Sri Dharmasthala Manjunatheshwara Institute of Technology, Ujire, Karnataka, India

Correspondence should be addressed to Shilpa R; shilparayee@gmail.com

Received: 27 November 2025 Revised: 12 December 2025 Accepted: 26 December 2025

ABSTRACT- Many crimes aren't reported - even though speaking up helps keep communities safe. Fear of being exposed keeps people quiet, along with doubts about law enforcement's reliability. Old-style methods usually demand ID or face-to-face talk, making it harder for victims to act. These processes tend to be slow, paper-heavy, and out of touch. As tech becomes more common in daily life, better tools are needed - ones that ensure secrecy without sacrificing ease. A smarter way would let users share info privately, quickly, through trusted digital path.

This project introduces a smart tool for reporting crimes, using AI to sort incidents, pull out key details, spot emotions in reports, plus judge potential dangers. It lets people share info without giving names, safely manage photos or videos, send instant warnings when danger's high, also offers live-updating screens for police use. With features that protect user identity while delivering sharp insights, the setup tries making it easier to report issues, speed up reactions, yet build a stronger, more reliable way to handle crime.

KEYWORDS- Anonymous Crime Reporting, Artificial Intelligence, Machine Learning, Natural Language Processing, Privacy Protection, Risk Assessment, Sentiment Analysis, Real-Time Alerts, Secure Digital Evidence, Law Enforcement Support Systems

I. INTRODUCTION

Citizens staying safe mostly comes down to how fast and right crime info gets shared. Still, old-school ways of reporting run into issues like worry about getting backlashed, shame from others, or no privacy - so plenty of people stay quiet. That leads to tons of crimes never making it into official records, plus shaky stats that mess up police work. Now tech's moving quick - AI and internet tools have gotten way better - and fresh solutions can tackle problems Folkes've dealt with forever.

The AI-powered crime tool lets people share info safely, no name needed. This setup uses smart tech to understand what users say. Instead of names, it spots places, actions, feelings, and how serious things are. Because of this, reports get sorted fast - urgent ones jump ahead. What makes it work? Clever language reading plus pattern recognition behind the scenes.

The platform runs on the MERN setup - MongoDB, Express.js, React.js, Node.js - for its main structure, while handling AI tasks through a Python-powered FastAPI. It also supports uploading video or audio proof, gives different access levels based on user roles, uses one-time passwords for login security, sends instant alerts via Socket.IO, plus offers cops a special dashboard packed with live insights and features to track cases.

Keeping things private, easy to reach, yet smart in operation, this setup brings a fresh way to report crimes - linking people directly with police through reliable systems.

II. RELATED WORK

Anita Verma's [1] et al. built a web tool letting people report crimes anonymously - no need to share personal details. This setup aimed to get more folks involved, which led to higher numbers of reports, particularly for touchy issues like harassment or mistreatment.

R. Kumar et al. [2] introduced a system powered by machine learning to sort crimes based on written reports. Their work tested three methods - Naive Bayes, Support Vector Machine, plus Random Forest - to see which worked best. Results showed Random Forest outperformed the others when identifying crime types.

P. Singh's et al. [3] built a web tool that lets people report crimes online - using NLP, it pulls out details like where and when an event happened or what kind of incident occurred. This method helps cut down time cops spend on paperwork.

M. Patel et al. [4] proposed a platform for reporting crimes, focusing on keeping users anonymous and their data private. Instead of real names, the setup relied on fake IDs combined with strong login security to shield those who report, while still making sure everything worked smoothly.

S. Banerjee et al. [5] looked into ways to spot emotional strain in crime write-ups through sentiment checks. Their setup used VADER plus alternative tools that picked up intense feelings tied to urgent situations within written records.

N. Joshi's et al. [6] built an AI tool for police that ranks crime reports by how serious they are and surrounding clues - so agents can tackle urgent cases faster.

K. Malhotra et al. [7] developed a tool for spotting key details in messy crime stories - turning them into clear records. The method worked better at pulling out places, people accused, or when things happened.

A. Thomas et al. [8] developed a live crime warning tool that sends alerts to officers when set urgency conditions are met. It actually cut down reaction speed during crises.

S. Gupta's et al. [9] developed a crime-reporting app where people can send photos or clips as proof. This research stressed safe storage of files along with proper data tracking.

R. Naik et al. [10] rolled out a system based on rules to check risk levels by spotting key terms, judging emotional tone, along with how much proof exists. This method worked well enough while staying straightforward when ranking threats.

J. Wilson et al. [11] came up with a private online reporting system built around ease of use and reach. Because of their work, it turned out that easier layouts boost how often regular people report issues.

H. Ahmed et al. [12] tested machine learning tools to spot fake or confusing crime alerts. The setup made systems more trustworthy by marking odd entries so humans could check them later.

T. Rao et al. [13] built a cloud-powered tool for managing crimes, allowing users to send tips without revealing identity while reports get reviewed in one central place. This setup helped police departments handle growing workloads easier and reach info faster.

P. Kaur et al. [14] looked into spotting crime patterns through old records using machine learning groupings. Their study showed that visualizing trends helps police plan better moves.

A. D'Souza et al. [15] developed a crime analysis tool using AI, showing kinds of crimes, danger levels, besides case progress. This system helped police respond faster while making daily operations smoother.

R. Chandra, et al. [16] developed a tool using natural language processing to spot signs of threats or violence in police reports. This approach helped catch risky cases sooner than before.

S. Iqbal, et al. [17] looked at how emotion show up in secret crime tips by using tools that sort feelings. The results showed that when fear or anger levels were strong, those situations often needed fast help.

K. Verma et al. [18] developed a tool that sorts crimes by text, using TF-IDF along with SVMs. This setup worked well in telling crime types apart, showing solid results.

M. Sharma et al. [19] worked to boost data safety in web-based crime reporting by cutting down stored info while using tighter entry rules. Because of this method, people felt more confident using those online services.

V. Ramesh et al. [20] developed a machine learning tool to predict crime risks by judging how urgent each report is. This setup worked well at spotting serious or critical situations, helping police act quicker.

III. PROBLEM STATEMENT

Crime hits places everywhere, but lots of cases never get shared - people fear how they'll be treated or doubt anything would change anyway. Worries about getting hurt after speaking out, being judged, or having private details revealed keep folks from reaching out to cops. On top of that, when trust in police is low or secrecy feels shaky, most choose to stay quiet instead. If someone thinks no one will care or their name might leak, they're far less likely to come forward - so offenses slip under the radar without answers. Old ways of reporting crimes make things worse - using slow, hands-on steps that take ages. Filling out stacks of forms, showing up in person, having to show your ID - it feels like a hassle and scares people off. On top of that, these setups don't have smart tech to spot crime trends, handle digital files, or send alerts fast when something's wrong. Without useful data or quick reactions, police can't act swiftly, which weakens trust over time. In the end, lots of victims get ignored, while chances to stop crime early just slip away.

IV. OBJECTIVES OF THE PROPOSED SYSTEM

The following are the Objectives of the proposed Neo The main aim here is building a fresh crime-reporting system powered by AI, fixing today's weak spots while giving police useful info. Key targets are listed next.
Primary Objectives

A. Develop an Anonymous Crime Reporting Platform:
Build a safe, simple system where people can share concerns while staying anonymous - this boosts the chance they'll speak up about tough situations.

B. Convert Free-Text Reports into Structured Data:
Use NLP methods to spot main details like what crime happened, where it took place, when it occurred, who was there - also pulling out any vital facts from how users describe things.

C. Enable Automated Crime Classification and Urgency Assessment:
Train smart systems to sort reports into set types - like robbery or abuse - while judging how serious each case is by tone, impact, and situation. Instead of relying on fixed rules, these tools learn patterns over time to tell which events need fast action.

D. Provide Advanced Visualization and Analytics:
Make live graphs, area visuals, dropdown picks, or number summaries so cops can spot patterns, find trouble zones, or follow how often crimes pop up in different spots.

E. Ensure End-to-End Security and Anonymity:
Keep data safe by using encryption plus set clear roles for who sees what. Mix in anonymizing tricks so no one's identity gets exposed. Rely on solid API setups instead of weak links during reports or checks.

F. Support Real-Time Alerts and Case Prioritization:
Set up a live alert tool using Socket.IO so cops get instant updates on urgent cases - this helps them respond fast while managing resources smartly.

G. Facilitate Secure Evidence Management:
Upload pics, movies, or files - checks what type is allowed, locks down who can view, keeps everything safe in storage.

H. Enable Real-Time Communication Between System Components:
Foster smooth chat among the report tool, server parts, or the AI helper so tasks run side by side without slowing down.

V. METHODOLOGY

The proposed system works by safely guiding crime reports - from start to finish - using smart tools that keep things fast and protected. Instead of relying on basic methods, it uses strong login checks along with private reporting paths so people can share info without fear. Smart software steps in once a report comes in, sorting details automatically while keeping data locked down tight. Real-time updates link users and police smoothly, making responses quicker when needed most. Each part plays a role: shielding identities, handling files securely, speeding up reviews, plus triggering warnings based on patterns found behind the scenes.

A. User Authentication and Identity Management

The login starts by sending a one-time code to your email so we know it's really you. After that, you get a special token to keep your session safe while using the app. If you're not signed in, the system gives you a unique guest ID instead. Since each person has their own role, they can only use parts of the site meant for them.

B. Report Submission Process

Users can send reports using set templates or just typing out their own words - also attaching photos, clips, or files if needed. The browser checks entries right away, making sure everything's filled in properly. If someone picks anonymous reporting, the server strips personal data before saving anything.

C. AI Processing Pipeline

Each report that comes in runs through several models at once. While VADER checks sentiment, text2emotion picks up emotional tones. The crime type gets guessed by a scikit-learn setup. SpaCy's NER tool pulls out key people or places mentioned. A special formula then rates how risky each case seems. These results combine into one clear overview for police use.

D. Evidence Management

Evidence gets checked for format and dimensions first - then it's uploaded through Multer. Stored files come with indexed details so they're easy to find later. Access is limited to those with proper login keys. The officer's panel shows media right inside the browser, no extra tools needed.

E. Real-Time Notification System

The system relies on Socket.IO to send quick alerts the moment a high-risk report pops up. While active, secure channels handle data flow; if someone's offline, warnings get saved till they reconnect. Once back, those missed messages come through automatically. Marking alerts as read or not helps officers keep track without confusion.

F. Police Dashboard and Crime Analytics

The police dashboard gives cops a clear look at what's happening across the system - like how many cases are waiting, flagged as high-risk, or already sorted. Instead of just numbers, it shows patterns using visuals made with Chart.js, such as slices, bars, or trend lines. Filtering options let users' narrow things down by offense category, timeframe, case status, or danger level. On top of that, pop-out panels offer smart summaries built by AI, alongside attached proof clips.

G. Security Framework

The system keeps things safe using end-to-end encryption, along with password protection powered by bcrypt. It's set up to manage cross-origin access carefully, while API calls get slowed down after too many tries When someone submits a report without logging in, their details stay hidden thanks to tight privacy rules - meaning nothing personal ever gets saved or shown.

VI. SYSTEM DESIGN

In the below figure 1 shows how the whole setup works for the AI-based crime reporting tool that keeps users hidden. It points out how the front part talks with the back end, smart features, and storage spots. Built with React, the visible side includes screens people use, report forms, plus views to check submissions or stats. Instead of direct links, it uses REST APIs along with Socket.IO to swap info between sides - mixing typical calls and instant changes. That way, everything feels quick and steady during use, sending details safely over for handling behind the scenes.

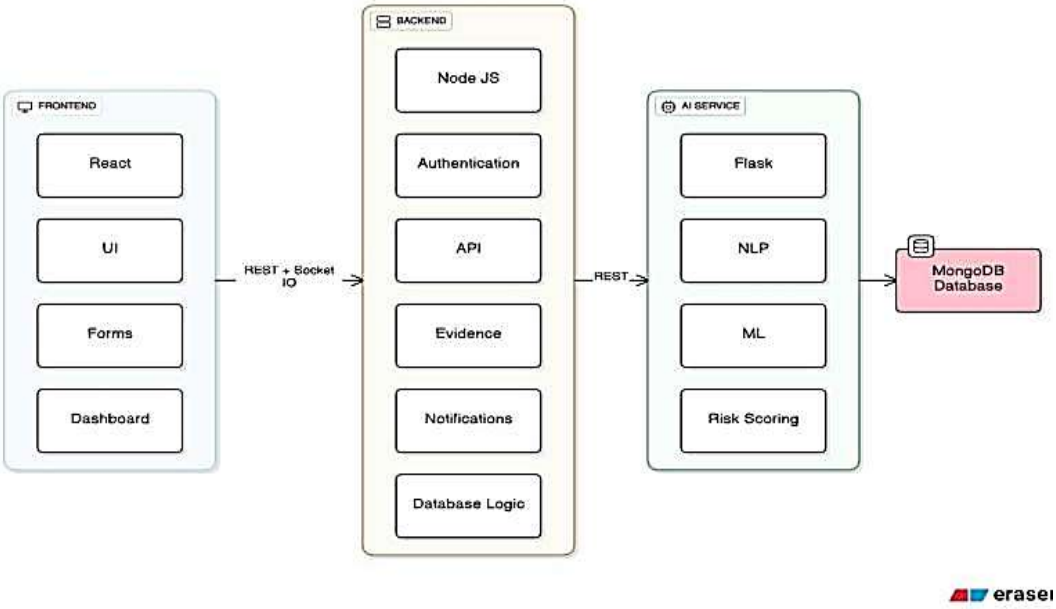

Figure 1: System Architecture of the proposed system

The backend runs on Node.js, taking care of key jobs like login control, API handling, report tracking, alerts, also database operations. It links the front end with the AI system smoothly. The AI part uses Flask to analyze submitted reports through natural language tools and learning algorithms - doing things like sorting crimes, pulling out important details, besides judging threat levels. Finished outputs get saved into MongoDB, holding records of incidents, person info, file tags, along with processed insights. This split design helps grow easily, keeps data moving safely, plus connects smart analysis fast with live reporting for police work.

VII. ALGORITHMS USED

The setup uses various NLP, machine learning, along with rule-based methods. One handles a specific task to build a solid, smart reporting tool.

A. VADER Sentiment Analysis

VADER checks emotions in text people send, spotting if it's good, bad, or just okay. So, the system gets better at sensing when someone's upset or urgent. It uses different signals instead of relying on one clue. Accuracy sits near 96% on F1, which means it works well for feelings in crime reports. Performance stays strong even with messy real-world input.

B. Named Entity Recognition (spaCy)

The spaCy NER tool pulls out important details from messy texts - like people's names, places, times, or items. So, the system turns casual user notes into organized info, hitting over 90% precision, which sharpens understanding while helping automation handle cases faster.

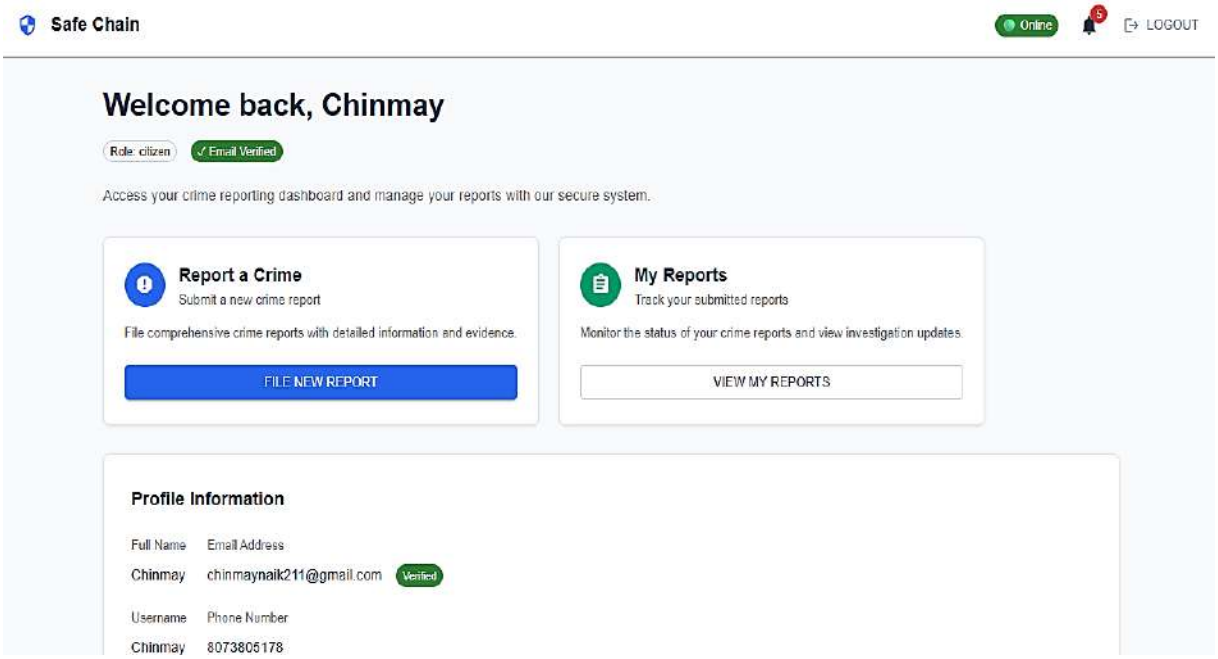

Figure 2: User Dashboard

C. Crime Classification (scikit-learn)

The system uses scikit-learn methods, where models learn from TF-IDF data to sort incidents into set crime types. Because of this setup, it hits over 85% precision on eight key categories. That means the tool can label cases by itself - no person needed. Instead of relying on hand-sorting, it runs predictions based on patterns found during training.

D. Text Emotion Detection (text2emotion)

The text2emotion tool picks up feelings like anger, fear, sadness, joy, or surprise. It helps the system catch subtle emotional hints while offering richer background - really helpful when judging how serious a report is, especially if it involves abuse, threats, or painful experiences.

E. Custom Risk Scoring Algorithm

The risk scoring system gives a number between 0 and 100 based on things like how serious the crime might be, what tone is found in the text, proof that's present, along with certain risky words spotted. Depending on this number, each case gets sorted into low, medium, or high priority - so urgent ones stand out fast.

F. Urgency Detection Rules

The system relies on smart shortcuts to scan for danger signs like certain words, how things are said, or context clues. Because of this setup, it can quickly spot serious threats and send real-time warnings straight to police.

Figure 2 shows a clean layout meant for everyday access. From here, people can file fresh reports about crimes instead of waiting. Past entries sit ready to be reviewed at any time. Checking on how things are going happens right within this space too. Simplicity guides the look so anyone feels comfortable using it without giving identity away.

VIII. SYSTEM IMPLEMENTATION SNAPSHOTS

The pictures walk through how the AI-backed crime reporting tool works - from logging in securely to checking data later. Instead of regular sign-up, it uses OTP verification so only real people get access. Once inside, folks land on a clear home screen where they can start new reports or check ones already sent. They're also able to update personal info whenever needed - no hassle. A

separate section breaks down patterns using charts: what kinds of crimes pop up most, their current state, danger ratings over time, plus latest actions taken. Each incident shows specifics like threat level and whether someone's looking into it now. Then comes the officer's side - a live map and stats feed help them stay updated minute-by-minute. This part lets cops watch unfolding situations, study shifts across areas, follow ongoing probes - all without switching apps. Together, these screens link public tips with faster police response behind one shared setup.

Figure 3: User registration page

Figure 3 shows a sign-up screen relying on OTP checks to confirm identity. Security stays strong here without asking for too much personal detail. Privacy gets preserved because less data is collected up front. What stands out is how access works smoothly yet keeps things locked down.

Figure 4: Login page

Figure 4 shows a sign-up screen relying on OTP checks to confirm identity. Security stays strong here without asking for too much personal detail. Privacy gets preserved because less data is collected up front. What stands out is how access works smoothly yet keeps things locked down.

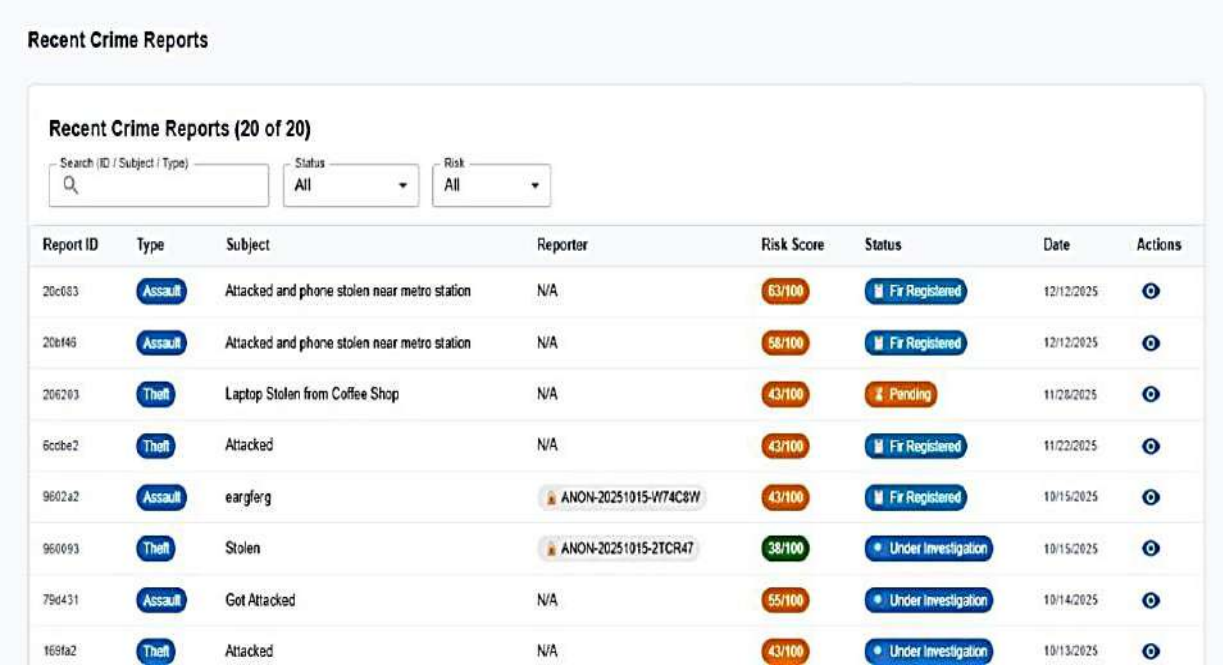

Figure 5: Registered Report Page

Figure 5 shows completed crime submissions. Report type appears alongside when it was sent, its present condition, one of several urgency tags. Tracking happens here while staying anonymous. Progress stays visible without exposing who reported.

Figure 6: Admin dashboard

Figure 6 shows a snapshot of the police command interface shows tools for handling incident data. Officials can sort reports by type, how urgent they are, or where they happened - using filters that adapt fast. Alerts pop up live, pulling attention to serious situations needing quick moves. Charts and maps display patterns so teams see what matters right away.

Figure 7: Analysis Page

In the above Figure 7 shows a snapshot of the analysis screen shows graphs built from crime records. Because visuals help, trend lines highlight how incidents shift over time. Severity levels appear through color-coded markers instead of words. When patterns emerge across hours or days, they guide responses without guesswork. Authorities use these insights to adjust their approach where needed.

IX. CONCLUSION

This crime reporting system keeps things safe, smart, and focused on people's needs - fixing what's broken in old-school reporting. It lets folks speak up without giving their name, uses clever tech to sort tips, also allows instant updates so info flows smooth. No identity needed, brains behind it analyze reports fast, plus live chat boosts how many joins in - all while guarding each person's privacy tight. Things like auto-sorting crimes by type, judging danger levels, locking down proof safely, along with clear data views help cops act quicker, choose better. In the end, this setup builds stronger links from streets to squads, speeds up responses, cuts delays, helps stop trouble before it grows, makes neighbourhoods safer overall.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] Verma *et al.*, "Anonymous web-based crime reporting system," *International Journal of Computer Applications*, vol. 12, no. 4, 2021.

[2] R. Kumar and S. Mehta, "Machine learning approaches for crime classification," *International Journal of Advanced Research in Computer Science*, vol. 10, no. 2, 2020. Available from: https://tinyurl.com/3vx5rv6n

[3] P. Singh *et al.*, "Online crime reporting system using NLP," *International Journal of Engineering Research and Technology*, vol. 9, no. 5, 2020. Available from: https://doi.org/10.48550/arXiv.2412.16614

[4] M. Patel and J. Shah, "Secure anonymous crime reporting platform," *International Journal of Information Security Science*, vol. 8, no. 3, 2019.

[5] S. Banerjee and A. Roy, "Sentiment analysis for detecting emotional distress in crime reports," *International Journal of Data Analytics Science*, vol. 6, no. 1, 2021. Available from: Crime Detection Using Sentiment Analysis

[6] N. Joshi *et al.*, "AI-based crime severity ranking system," *International Journal of Artificial Intelligence Applications*, vol. 7, no. 2, 2022.

[7] K. Malhotra and R. Iyer, "Information extraction from unstructured crime narratives," *International Journal of Natural Language Processing*, vol. 5, no. 4, 2019.

[8] Thomas and L. Fernandez, "Real-time crime alert system for law enforcement," *International Journal of Smart Systems Applications*, vol. 9, no. 1, 2021.

[9] S. Gupta *et al.*, "Multimedia-based crime reporting application," *International Journal of Multimedia Applications*, vol. 11, no. 3, 2020. Available from: https://tinyurl.com/bdd2af73

[10] R. Naik and D. Kulkarni, "Rule-based crime threat assessment system," *International Journal of Information Technology*, vol. 8, no. 2, 2018.

[11] Wilson and P. Carter, "User-friendly online crime reporting systems," *International Journal of Human–Computer Interaction*, vol. 6, no. 1, 2017.

[12] Ahmed *et al.*, "Detection of false crime reports using machine learning," *International Journal of Machine Learning Applications*, vol. 9, no. 4, 2022.

[13] T. Rao and S. Mishra, "Cloud-based anonymous crime management system," *International Journal of Cloud Computing Services*, vol. 7, no. 2, 2021.

[14] P. Kaur and M. Singh, "Crime pattern analysis using machine learning," *International Journal of Big Data Analytics*, vol. 10, no. 1, 2023.

[15] D'Souza *et al.*, "AI-based crime analysis and monitoring system," *International Journal of Advanced Computer Science Applications*, vol. 12, no. 6, 2022.

[16] R. Chandra and N. Bose, "NLP-based threat detection in crime reports," *International Journal of Cyber Security Applications*, vol. 5, no. 3, 2021.

[17] S. Iqbal and F. Rahman, "Emotion analysis in anonymous crime tips," *International Journal of Affective Computing*, vol. 6, no. 2, 2020.

[18] Verma and P. Jain, "Crime text classification using TF-IDF and SVM," *International Journal of Text Mining Applications*, vol. 9, no. 1, 2019.
[19] M. Sharma and R. Gupta, "Enhancing data privacy in online crime reporting systems," *International Journal of Information Privacy and Security*, vol. 8, no. 4, 2022.
[20] V. Ramesh and S. Nandhini, "Machine learning-based crime urgency prediction," *International Journal of Predictive Analytics*, vol. 11, no. 2, 2023.

ABOUT THE AUTHORS

Shilpa R has completed her B.E in the year 2011, MTech in the year 2015 from VTU Belagavi, and Ph.D. in the year 2024 from VTU Belagavi. Her areas of interest are Image Processing and Deep Learning. She has published 13 international research papers. She is the life member of Kannada Sahitya Parishad.

Abhisheka C. Hegde is a B.E. Computer Science graduate from VTU, Belagavi, who studied at SDM Institute of Technology, Ujire. He emphasized hands-on coding over theory, gaining practical experience in Java, full-stack development, MERN stack, and intelligent systems. With experience across both front-end and back-end development, he focuses on building scalable, real-world applications through self-driven learning and implementation.

Anushree Joshi finished her B.E. in Computer Science through VTU, Belagavi - she learned at SDM Institute of Technology in Ujire. Instead of just theory, she picked up real-world app-building experience. That hands-on work helped her get comfortable with Java along with full-stack tools. On top of that, she's gained know-how in cloud platforms.

Chinmay Naik received his B.E. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belagavi. His areas of interest include Java, python programming, full stack development. He was a student in the Department of Computer Science and Engineering at SDM Institute of Technology, Ujire.

Darshan Hegde received his B.E. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belagavi. His areas of interest include Java programming, web development. He was a student in the Department of Computer Science and Engineering at SDM Institute of Technology, Ujire.