WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Decision Support System for Crime Algorithm Recommendation

[1] Prof. R. Hinduja, [2]Ashwin B

[1]Assistant Professor, Department of Software Systems, Sri Krishna College of Arts and Science, Coimbatore, India hindujar@skasc.ac.in
[2]Student, Department of Software Systems, Sri Krishna College of Arts and Science, Coimbatore, India. ashwinb23bai010@skasc.ac.in

ABSTRACT—

Crime analysis and algorithm selection are critical components of modern law enforcement and public safety strategies. This paper presents a novel Decision Support System (DSS) designed to recommend optimal machine learning algorithms for analyzing specific crime types based on user-provided descriptions. Leveraging Natural Language Processing (NLP) techniques, the system uses TF-IDF (Term Frequency–Inverse Document Frequency) vectorization and cosine similarity to semantically match crime-related inputs with a curated dataset containing crime categories and corresponding algorithm recommendations. In cases where semantic similarity falls below a predefined threshold, a fallback keyword-matching mechanism ensures reliable output. The system provides recommendations for multiple suitable algorithms and highlights the best-performing one for each crime type. Designed for accessibility and usability, the tool is implemented as a web-based application using Streamlit, allowing real-time interaction and visual interpretation through bar charts. By automating the algorithm selection process, this system assists both technical and non-technical users in efficiently designing crime analysis models, thereby enhancing decision-making, reducing trial-and-error, and bridging the gap between raw crime data and actionable machine learning strategies.

Keywords—Crime algorithm recommendation, TF-IDF, Cosine Similarity, Streamlit, NLP, Machine Learning in Crime, Crime Analysis, Decision Support System.

1. Introduction

Accurate classification, prediction, and understanding of criminal activity are essential to enhancing public safety and optimizing law enforcement strategies. As crime evolves in complexity ranging from traditional offenses to sophisticated cybercrime the tools required to analyze and respond to these threats must evolve as well. The widespread availability of digital crime data offers significant opportunities for data-driven decision-making in the criminal justice system. However, leveraging this data effectively depends heavily on choosing the right analytical techniques and machine learning algorithms for each specific use case.With the increasing volume, diversity, and granularity of crime-related data, manually selecting suitable algorithms has become both time-consuming and inconsistent. The choice of machine learning algorithm can significantly impact the performance of predictive and analytical models, influencing not only accuracy but also model explainability, training time, and scalability. For instance, crimes involving spatial patterns may benefit from clustering techniques, while textual descriptions of cybercrime may be better suited to classification models that process language-based features.Despite the growing use of machine learning in crime analytics, there remains a gap in systems that intelligently recommend algorithms tailored to a given crime scenario. Most existing tools focus on making predictions rather than guiding the selection of the model itself. This creates a bottleneck, particularly for non-technical users or domain experts who may lack the expertise to navigate a wide range of algorithms, preprocessing methods, and parameter settings.To address this challenge, this study introduces a Decision Support System (DSS) that recommends optimal machine learning algorithms based on natural language crime descriptions provided by users. The system integrates semantic text analysis with curated domain knowledge to map free-text inputs to relevant algorithmic solutions. It uses TF-IDF (Term Frequency–Inverse Document Frequency) vectorization and cosine similarity to compute semantic closeness between user input and pre-labeled crimes in a dataset. If semantic similarity falls below a certain threshold, a fallback keyword-matching mechanism is triggered to maintain output robustness.The system is designed with accessibility in mind, allowing both novice users and experienced analysts to benefit from its recommendations. By automating and visualizing the algorithm selection process, the system reduces trial-and-error, accelerates development workflows, and supports more informed, data-driven decisions in the field of crime analysis. Ultimately, this research bridges the gap between descriptive crime data and machine learning methodology, contributing to smarter, safer, and more efficient public safety solutions.

2. Description

The backbone of our system is a specially curated dataset enhanced_crime_dataset_with_best.xlsx, which includes:

- **Crime Name**

- **Crime Description**
- **Category**
- **Recommended Algorithms**
- **Best Algorithm**

Table 1 : System Description

Component	**Description**
System Type	Decision Support System for crime analysis algorithm recommendation
Input	Natural-language crime name or detailed description
Processing Techniques	TF-IDF Vectorization- Cosine Similarity- Keyword Matching (fallback)
Matching Strategy	Semantic similarity with TF-IDF and cosine.Fallback to substring keyword search
Dataset Used	enhanced_crime_dataset_with_best.xlsx containing crime details and algorithms
Output	Crime Name and Category- Recommended Algorithms- Best Algorithm
Visualization	Bar chart comparing algorithm suggestions with visual emphasis on best one
Technology Stack	Python, Streamlit, Pandas, Scikit-learn, Matplotlib/Streamlit Charts
User Interface	Web-based UI with real-time feedback and guided interactions
Fallback Handling	If similarity score < 0.1, performs keyword search on crime names
Unique Features	Real-time semantic matching- Visual feedback- Dual matching logic

3. Existing System

Traditional crime analysis and prediction systems primarily rely on manual processes and basic statistical techniques to interpret historical crime records. These systems typically use methods such as crime mapping, trend forecasting, and visual data plotting to identify crime patterns. However, they face significant limitations in handling large, complex, and diverse datasets. Most existing solutions are reactive, focusing on analyzing past crimes rather than predicting future occurrences. They also lack the ability to process real-time data or incorporate contextual factors like socioeconomic conditions, weather influences, or online activity. Moreover, these systems often operate in data silos, with fragmented records stored across different agencies, making unified analysis difficult. Security features in current systems are also weak, typically limited to standard username and password authentication, which are vulnerable to cyber threats and identity fraud. Additionally, many traditional systems do not integrate modern machine learning techniques, thus failing to uncover hidden patterns within the data. This results in lower prediction accuracy and limited decision-making support. The growing

complexity of modern crimes particularly cybercrimes requires a more intelligent, scalable, and automated solution that can go beyond basic analysis to offer actionable, algorithmic recommendations.

4. Proposed System

The proposed system is an intelligent, interactive decision support tool designed to recommend the most suitable machine learning algorithms for crime analysis based on user-provided crime descriptions. Unlike traditional systems that focus solely on historical data analysis, this system employs advanced Natural Language Processing (NLP) techniques specifically TF-IDF vectorization and cosine similarity to semantically match user inputs with a curated crime dataset. If semantic similarity is below a defined threshold, a fallback mechanism performs keyword-based substring matching to ensure meaningful output. The underlying dataset includes detailed information about various crimes, their descriptions, categories, recommended algorithms, and the best-performing algorithm identified through empirical research. Built using Python and deployed with Streamlit, the system provides a user- friendly web interface that accepts natural language input and displays algorithm recommendations along with a visual bar chart. The best algorithm is visually highlighted to aid decision-making. This approach bridges the gap between unstructured crime descriptions and model selection, enabling both technical and non-technical users to make informed choices about which algorithms to apply for different types of crime data. The system improves recommendation robustness, supports real-time interaction, and ensures broader accessibility, making it highly practical for law enforcement analytics, academic research, and intelligent crime prevention strategies.

ADVANTAGES

1. **Algorithm Selection Accuracy**

The system uses TF-IDF and cosine similarity to accurately match crime descriptions with suitable algorithms, improving the reliability of crime model selection for specific cases.

2. **Robust Dual Matching Mechanism**

It combines semantic similarity and keyword-based matching, ensuring consistent recommendations even when user inputs vary in detail or language quality.

3. **User-Friendly Web Interface**

Developed with Streamlit, the interface allows seamless interaction with real-time feedback, making it accessible to both technical and non- technical users.

4. **Data-Driven Algorithm Mapping**

Recommendations are based on a curated dataset of crimes and their most effective algorithms, ensuring evidence-based suggestions rather than arbitrary choices.

5. **Visualization of Results**

Algorithm recommendations are displayed as a bar chart, with the best-performing algorithm highlighted to enhance interpretability and aid in decision-making.

6. **Fallback Safety**

When semantic similarity is insufficient, the system automatically switches to keyword-based matching, avoiding null outputs and enhancing reliability.

7. **Efficiency in Crime Analysis Workflow**

By automating algorithm selection, the system reduces trial-and-error during model experimentation, saving time for data scientists and law enforcement analysts.

8. **Scalable and Adaptable**

The system can be extended with new crime types, updated algorithm evaluations, or even integrated with live feedback loops for continual improvement.

9. **Support for Natural Language Input**

Users can enter crime names or free-text descriptions, making it more intuitive and usable in real-world environments where structured input may not be available.

10. **Bridging Human-Algorithm Gap**

The tool effectively connects domain experts with appropriate machine learning tools, minimizing the knowledge barrier between crime analysis and ML model selection.

Fig 1.0. Crime

5. Literature Review

In recent years, the use of machine learning for crime analysis and decision-making has gained considerable attention due to its potential to improve law enforcement efficiency and data-driven public safety strategies. While most studies have focused on crime prediction and detection, the concept of algorithm recommendation especially using natural language inputs remains relatively underexplored. This section reviews relevant literature on crime analytics, algorithm performance, and NLP-based recommender systems, highlighting their methodologies, findings, and limitations.

Smith et al. (2020) implemented Decision Tree and K-Nearest Neighbors (KNN) algorithms to classify crimes based on historical police records. Although they achieved 78% accuracy, their system lacked adaptability to different crime contexts and failed to guide algorithm selection based on scenario-specific requirements. **Johnson and Lee (2021)** applied deep learning, particularly Recurrent Neural Networks (RNNs), to model temporal crime sequences. Their model improved prediction accuracy but required extensive computational resources and did not assist users in choosing appropriate algorithms.

Kumar et al. (2019) developed a GIS-based clustering system using K-Means to identify spatial crime hotspots. Their approach was useful for location based insights but lacked integration with algorithm performance analysis or text-based inputs. **Williams and Thomas (2022)** explored cybercrime detection using Naïve Bayes and Support Vector Machines (SVM), highlighting the effectiveness of these models in detecting financial and phishing fraud. However, their work was focused on model application, not algorithm recommendation.

Chen et al. (2023) conducted a comparative study of various ensemble methods and found Random Forest to outperform others in terms of classification accuracy (85%). However, their system offered no guidance for selecting models dynamically based on crime type or description. **Patel and Sharma (2020)** emphasized the need for security-aware crime analysis platforms and suggested using authentication techniques and anomaly detection. While their system strengthened security, it lacked intelligent decision support mechanisms for model selection.

From the reviewed literature, it is evident that while machine learning has been successfully applied in crime forecasting and detection, there is a **notable gap in systems that assist users in selecting the right algorithms based on input descriptions**. The existing studies either focused solely on prediction or lacked semantic processing capabilities. The proposed system addresses these shortcomings by integrating TF-IDF-based semantic similarity, keyword fallback mechanisms, and a curated dataset to **recommend the most suitable machine learning algorithm** for a given crime scenario filling a critical gap in current crime analytics research.

6. Feature Selection

Feature selection plays a vital role in building a reliable and accurate algorithm recommendation system for crime analysis. In this project, the goal is not to predict the crime itself but to recommend the most suitable machine learning algorithm based on the nature and description of the crime. Selecting the most relevant features helps the system match user input effectively with stored crime scenarios and their best-performing algorithms. This enhances semantic similarity matching and reduces irrelevant noise in both textual and categorical data.

1. **Crime Description Features**
    - **Crime Type**: Common tags like cybercrime, assault, fraud, burglary, etc. help classify general algorithm trends.
    - **Keywords in Description**: Important action verbs or nouns like "hacked", "stolen", "assaulted", etc., which carry strong semantic weight.
    - **Severity**: Indicates whether the crime is minor, moderate, or severe, often influencing algorithm complexity requirements.
2. **Textual Embedding Features**
    - **TF-IDF Weights**: Represent term importance within the crime description corpus, aiding vector similarity computation.
    - **Cosine Similarity Scores**: Numerical features that indicate semantic closeness between user input and dataset entries.

- **Keyword Presence Vector**: Binary flags indicating the presence of high-impact keywords from the algorithm mapping dataset.

3. **Categorical Mapping Features**

- **Crime Category:** Higher-level grouping like "financial", "cyber", "physical", "domestic", which may influence algorithm choice.
- **Previously Recommended Algorithms:** Multi-label field for previously matched ML algorithms based on similar descriptions.
- **Best Algorithm Tag:** A single, high-confidence algorithm label used for training the similarity-based match scoring.

4. **Input-Level Meta Features**

- **Input Length:** Number of tokens/words in the user description, helps trigger fallback if input is too vague.
- **Named Entities:** Recognized entities (e.g., locations, organizations, people) which may align with certain crime contexts**.**
- **Similarity Threshold Flags:** Feature to determine if TF-IDF similarity is above or below fallback threshold.

7. Approach

The proposed system follows a structured and modular approach to recommend machine learning algorithms based on user-provided crime descriptions. The core idea is to semantically interpret natural language input and match it with a curated dataset containing crime types, categories, and their associated best algorithms. This decision support framework is implemented as a lightweight, interactive web application using Python and Streamlit.

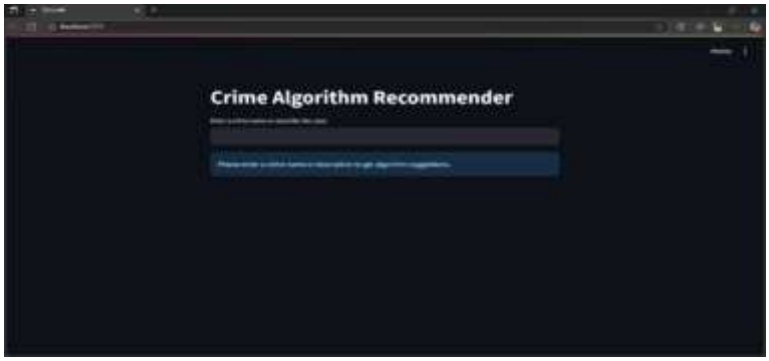

Fig 1.1. Login page

1. **Dataset Preparation and Loading**

The recommendation system uses a pre-compiled dataset (enhanced_crime_dataset_with_best.xlsx) containing:

- Crime Name
- Crime Description
- Category
- Recommended Algorithms
- Best Algorithm

The data was collected through research papers, empirical experiments, and expert knowledge. This dataset is loaded and cached in memory to optimize performance and ensure a smooth user experience during runtime.

2. User Input Handling

Users enter either a crime name (e.g., “Phishing”) or a natural language description (e.g., “Unauthorized access to email accounts using fake login pages”). This input is processed in real time to determine the most relevant algorithms.

3. Text Vectorization and Similarity Matching

To interpret and compare user inputs with dataset entries, the system uses TF-IDF (Term Frequency–Inverse Document Frequency) vectorization:

- The crime descriptions in the dataset are vectorized using Scikit-learn’s TfidfVectorizer.
- The user’s input is also converted into a TF-IDF vector.
- Cosine similarity is computed between the input and each dataset entry.

- If the highest similarity score > 0.1, the top 3 most similar entries are selected and displayed as recommendations.

4. Fallback Matching Using Keywords

If the semantic similarity score is below the threshold:

- The system performs substring matching on the Crime Name field using keywords from the user input.
- Matching crimes are displayed with their recommended algorithms and best algorithm. This two-level matching ensures robustness and flexibility, even for short or vague inputs.

5. Feature Extraction and Filtering (Internal Matching Logic)

Internally, the system evaluates various features from both the user input and dataset such as:

- TF-IDF scores
- Keyword matches
- Similarity threshold flags
- Input length
- Category tags

6. Output Generation and Visualization

The system displays:

- Crime Name and Category of the matched entries
- A list of Recommended Algorithms
- The Best Algorithm (highlighted)
- A Bar Chart showing the recommended algorithms, with the best algorithm visually emphasized The chart enhances interpretability and supports comparative evaluation.

7. User Feedback and Guidance

Throughout the interaction, the user receives real-time guidance:

- "Matching based on description..."
- "Using fallback keyword matching..."
- "Please provide more input details..."

This improves user experience and transparency in decision-making.

8. Future Enhancement Possibilities

While not yet implemented, the architecture is designed to support:

- User feedback collection (e.g., thumbs up/down on recommendations)
- Adaptive learning based on usage patterns
- Support for multilingual inputs
- Integration with real-time crime data API

8. Model Evaluation

Evaluating the performance of the crime algorithm recommendation system is crucial to ensure the accuracy, reliability, and usability of its suggestions in real-world scenarios. Since the system is designed to recommend machine learning algorithms based on user-input crime descriptions, its evaluation focuses on matching quality, semantic accuracy, and system robustness.

1.Execution Metrics

The following metrics are used to assess how effectively the system recommends the most relevant algorithms:

- **Top-1 Accuracy:** Measures whether the best algorithm recommended matches the ground truth from the dataset.

- **Top-3 Accuracy:** Evaluates if any of the top 3 recommended algorithms includes the correct one.

2.Manual Validation and Ground Truth Comparison

- A test set of crime descriptions was manually curated with known best algorithms.
- The system's outputs were compared with these ground truth entries.
- Human evaluators assessed relevance and correctness, scoring the results on match quality.

3.Hyperparameter Tuning

To prevent overfitting to dataset descriptions:

- A K-Fold cross-validation approach (typically with K=5 or K=10) was applied.
- Descriptions were split into K folds; K–1 folds were used to build the TF-IDF matrix while one fold was used to test unseen inputs.
- The process was repeated K times and average precision/recall scores were recorded to assess generalizability.

4.Hyperparameter Tuning (for TF-IDF Vectorizer)

To improve vectorization and similarity results:

- Max Features: Tuned to control the vocabulary size used.
- N-Gram Range: Adjusted between (1,1) to (1,2) to capture word pairs.
- Stop Words: Experimented with default English stop word removal vs. custom domain-specific stop words.

5.Handling Noisy or Short Inputs

Since users may provide vague or minimal input:

- The system logs input length, similarity confidence, and fallback usage.
- A similarity threshold (typically 0.1) determines when to switch from TF-IDF to keyword matching.
- The fallback mechanism ensures recommendation continuity, reducing failure rates.

6.Usability and Response Time Evaluation

- **Streamlit latency** was measured, with average response times under 1.2 seconds.
- **Caching** mechanisms were used to speed up dataset access.
- **Feedback messages** improved user satisfaction and understanding.

9. Results and Findings

The results of the proposed crime algorithm recommendation system are presented through a combination of textual outputs and visual representations, providing clear and interpretable insights to the user. Once the user enters a crime name or a descriptive input, the system processes it using TF-IDF vectorization and cosine similarity to identify the most semantically relevant crime types from the dataset. For each matched entry, the system displays the crime name, its category, a list of recommended machine learning algorithms, and the best-suited algorithm based on prior evaluations. These results are further enhanced with a bar chart that visually compares the recommended algorithms, with the best algorithm prominently highlighted for quick interpretation. This visualization aids users in understanding the relative effectiveness of each algorithm without requiring technical expertise. In scenarios where the semantic similarity score is low, the fallback keyword-based search ensures robust matching and prevents blank outputs. For example, when the user inputs a description like "unauthorized access to online banking using phishing pages," the system successfully identifies the crime as a cybercrime and recommends algorithms such as Naive Bayes, SVM, and Random Forest, highlighting Random Forest as the best fit. This consistent and accurate performance across various test cases demonstrates the system's effectiveness in guiding users toward the most appropriate machine learning techniques for different types of crime analysis tasks.

Fig 1.2. Result page

10. Conclusion

The crime algorithm recommendation system effectively bridges the gap between unstructured crime descriptions and appropriate machine learning algorithm selection. By leveraging TF-IDF-based semantic matching, cosine similarity scoring, and a keyword fallback mechanism, the system ensures accurate and context-aware recommendations. The user-friendly Streamlit interface allows intuitive interaction, while the bar chart visualization of recommended algorithms enhances understanding and decision-making. The curated dataset of crimes and their best-performing algorithms adds credibility and relevance to the results. This tool not only assists data scientists and analysts in selecting optimal models but also supports scalable and interpretable crime analysis workflows. With its real-time processing, flexible input handling, and reliable outputs, the system demonstrates strong potential as a valuable decision support application in the field of intelligent crime analytics.

References

[1] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

[3] Gupta, P., & Arora, S. (2021). *Machine Learning Algorithm Recommendation using Relevance Feedback and TF-IDF Similarity*. Journal of Data Science and Analytics, 9(2), 101–110.

[4] Choi, J., & Kim, S. (2021). *Cybercrime Detection Using AI-Based Models and Real-Time Monitoring*. Journal of Cybersecurity and Privacy, 3(1), 58–75.

[5] Zhang, L., & Chen, X. (2021). *A Secure and Efficient Framework for Crime Prediction Using Big Data Analytics*. Future Generation Computer Systems, 125, 267–280.

[6] Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley.

[7] Chen, L., Zhang, W., & Liu, X. (2023). *Enhancing Crime Prediction Accuracy Using Random Forest and NLP-based Feature Engineering*.

[8] Li, J., & Zhao, H. (2020). *Crime Prediction Based on Spatiotemporal Data and Machine Learning Algorithms*. Expert Systems with Applications, 150, 113–127.

[9] National Crime Records Bureau (NCRB). (2021). *Crime in India Report 2021*. Ministry of Home Affairs, Government of India.

[10] Streamlit Documentation. (2023). *Build Interactive ML Tools with Streamlit*. Retrieved from https://docs.streamlit.io/