International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 11 | Nov 2026 www.irjet.net p-ISSN: 2395-0072

Crime Pattern Analysis and Hotspot Prediction Using Machine Learning

Prof. V. A. Gahire[1], Sanchi Bansode[2], Prapti Jagtap[3], Swati Pawar[4]

[1]*Professor, Department Of CSE ICEEM, Maharashtra, India*

[2,3,4] *Students, Department of CSE ICEEM, Maharashtra, India*

Abstract - *Crime Investigating organized crime requires analysts to connect people, organizations, phone numbers, vehicles, financial accounts and locations scattered across unstructured case narratives, a process that is largely manual and difficult to scale. This paper presents, a prototype system for criminal-network analysis from crime-related text. Six categories of entities can be extracted using regex and keyword rules. Relationships can also be inferred from these entities using heuristic rules that look at how often entities appear together in sentences and what order they appear in. The entities and relationships are saved to an SQLite database optionally synced with Supabase. This is then modelled as a directed graph. Using NetworkX metrics can be calculated such as degree, betweenness centrality, density, connected components. A heuristic metric to find influential entities is also added. Three rule based methods are used to find command entities (hubs), bridges and cyclic entities. A FastAPI backend serves a React interface offering interactive graph visualization, entity inspection, filtering and manual entity entry. The current implementation is not using any machine learning or trained models, and does not predict geographical hotspots. However, the tool can be useful as it provides a clear, explainable and scalable framework for investigators to create structure within their cases, and identify key entities within the structure. Further work will include quantitative analysis of information extraction and relationships using annotated data, and comparing performance against learning methods.*

Key Words: Criminal network analysis, Rule-based entity extraction, Crime related text, Betweenness Centrality, Social Network Analysis, Graph Visualization

1.INTRODUCTION

Crime analysis provides law enforcement with insights that help them make decisions regarding resource allocation, investigation targets and the structure of criminal activity.. A large amount of work in the computational field concerns understanding spatial patterns, such as the clustering of crime and where crime may occur. Studies of place show that crime is highly concentrated in small geographic units [3], and a range of statistical and machine-learning methods have been proposed to estimate or forecast crime levels [4], [5]. A complementary and equally important question is relational: who is connected to whom, and through which phones, vehicles, accounts, organizations and locations.

Organized crime, such as narcotics trafficking, arms supply and money laundering, is carried out by collaborating offenders who form groups and play different roles, including leaders who issue commands and gatekeepers who control the flow of information and goods [1]. The evidence for these structures typically sits in free-text sources such as police reports, intelligence summaries and call-log narratives. The conversion of text into a structured network is mostly done manually. Xu and Chen describe a progression from manually built association matrices and link charts, through tools that draw graphs automatically, to social network analysis that supports deeper structural analysis [1].

This paper describes a prototype web-application which can automate the initial stages of the workflow. It extracts entities using regular expressions and keyword rules, forms relationships between entities using rule-based methodology and stores it. The data is then analysed and displayed in the form of a network. Entities in the text are identified using regular expressions and keywords. Entities are then joined together based on rule based relationships, saved and analysed as a network. The system has been designed to use rule-based methods to extract entities. It does not use any machine learning, deep learning or trained language models nor does it make any predictions regarding geographical hotspots. This contributes to the field by providing a clear baseline of how to perform this task end-to-end where all entities and relationships can be traced back to a rule and sentence within the source text.

The remainder of this paper is organized as follows: Sections 2 and 3 state the objectives and scope; Section 4 reviews related work; Sections 5 to 13 describe the design, methodology, data handling, analysis, tools and algorithms; Section 14 describes the current results and what remains to be evaluated; Section 15 concludes.

1.1 OBJECTIVES

- The primary objective is to give insight into the network of criminals.
- It will also identify structurally prominent entities and patterns (hubs, bridges and circular flows) using network analysis and transparent rules.
- On the web application, the extracted network can be viewed as an interactive network with metrics including degree, betweenness, and density.
- IAnalysts will be able to easily explore, filter and manually correct the individual entities and relationships in the network in order to maintain traceability back to the original text.

1.2 SCOPE

The overall aim of this system is to perform an analysis of criminal networks by structuring information from text based sources related to crime into a graph containing people, organisations, places, phone numbers, vehicles and financial accounts It will display the extracted entities and the relationships between them, together with network measures such as degree, betweenness centrality and density. This will help police forces & other related investigating agencies to understand how a criminal group is organized and identify the entities that connect or control it. The system highlights the structures within these networks such as control hubs by using clear rule based algorithms. This only allows text based network analysis. The system is limited to text-based network analysis. It does not use machine learning, and it does not track crime rates or predict crime types, locations or hotspots. These remain possible future extensions that would require historical, geocoded and time-stamped crime data. Its outputs are indicators that can be used by human review, not evidence.

2. LITERATURE REVIEW

Spatial and predictive crime analysis is a field that looks at where and when crimes occurs. Much of the research on crime asks these questions. Weisburd states that crime happens mainly in a small places, which pushes researchers to focus on those places. Data-driven models use this concept. Wang and colleagues use city data such as points of interest and taxi trips to guess the crime rate for each neighbourhood. Chen and colleagues use an ARIMA time-series model to predict how many crimes will happen in the future. A newer framework that is the background for this work mixes DBSCAN clustering, XGBoost classification and an interactive map. I find this approach intriguing.

Bias and transparency are concerns for predictive crime tools. Predictive methods rely on data and Lum and Isaac warn that systems trained on police records can keep old biases alive. If we act on the results those biases can become stronger. That is why transparency and human oversight must be goals for any crime-analysis tool. I worry that we may repeat mistakes if we do not watch closely.

Criminal network analysis studies how offenders connect, than where they are. Xu and Chen show that there are three waves of tools. The earliest use tables and link charts. The next wave uses graph-drawing software like Analyst's Notebook, Netmap and XANALYS Link Explorer. The latest wave adds social-network analysis to give structure. Their CrimeNet Explorer framework has four parts: building the network splitting it analysing its structure and showing it. It mixes a concept-space approach, clustering social-network analysis and multidimensional scaling. In a study people recognised the members and how groups interact much faster when they had the structural analysis compared to just looking at a picture. I think having structure really helps.

Centrality is used to identify how important each node is within a network. Betweenness was proposed by Freeman to measure how many times a node appears in the shortest paths between other nodes.Brandes later made a way to compute betweenness. That faster method is, in the NetworkX library.

3. SYSTEM DESIGN

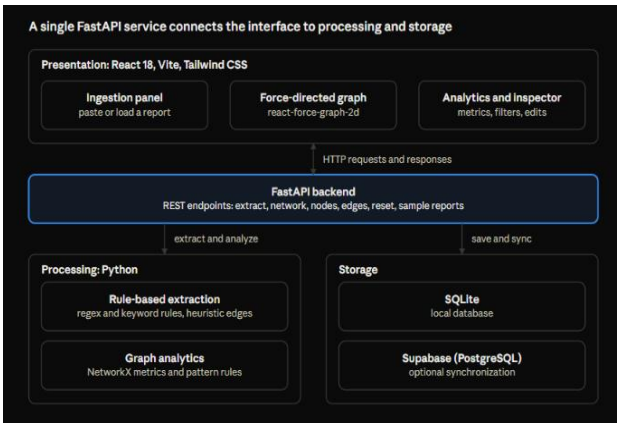


Fig.3.1. System Architecture

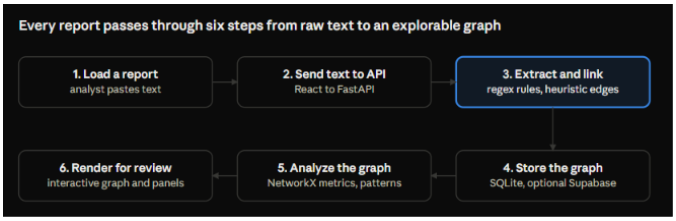


Fig.3.2. Data Flow

4. METHODOLOGY

This methodology is split into four parts:

Text to graph: Text to graph is achieved using a rule based approach to extract entities and infer relationships between them. (See section 8 and 12).

Storage: persistence of nodes and edges (Section 8)

Graph analysis: computation of standard network measures and application of pattern rules (Section 9).

Visualization and interaction: an interactive web interface (Section 11).

4.1. DATA COLLECTION

The prototype currently uses two input sources.

The first source is a series of 3 example case stories which have been included within the project. They have been written to show examples of how they would be used. These cases are a narcotics and hawala case, an arms supply case and a ransomware escrow case. The second source will be text provided by the user, this can be pasted into the ingestion panel and passed along to the extraction endpoint.

No real law-enforcement records are used, and no external dataset is currently integrated. For the evaluation stage, a corpus of crime-related narratives will be assembled and annotated by hand with the six entity categories and the relationships between them. Its source, size and annotation procedure will be reported once it has been created, together with the ethical or permission arrangements under which any real text is used.

4.2 DATA PREPROCESSING

The input text is pre-processed. This involves splitting it into sentences after normalising line breaks. Entity mentions are identified by matching regular expressions and keywords. Non entities are filtered out by applying exclusion lists. For each entity a normalised identifier is created; this helps group repeated entities and reduce redundancy by creating a single node. No missing data imputation was carried out as the input data is free text. The data is structured into a graph of nodes and edges.

4.3 DATA ANALYSIS

A NetworkX directed graph is generated then degree, betweenness [7], density and connected components are calculated. An influence score is calculated using a heuristic function that uses the degree multiplied by 1.5 and the betweenness multiplied by 10 and then multiplied by the risk multiplier. The nodes are sorted according to their score. There are three rules that are applied to highlight any structures. If a command centre exists (a highly ranked node has degree >=3 and connects two or more categories), if there is a bridge (node has betweenness >=0.25 and degree >=2) or a loop (directed cycle of size 2-6).The flags are indicators for analyst review and have not been validated.

4.4. DATA VISUALIZATION

The React frontend shows an interactive graph of the network. This allows analysts to search and filter, investigate entities, view metrics and highlighted patterns, and manually add/remove entities.

4.5. TOOLS

The backend is built using Python 3.12. The regex extraction is handled by the built in re module. FastAPI 0.141.1 is used to provide a REST API. The server is run using Uvicorn 0.52.4 on port 8000. Pydantic 2.13.4 is used to create and validate node, edge, request models. The HTTP client library is HTTPX 0.28.1. python-dotenv 1.2.3 is used to pull in variables from a .env file.

Networkx 3.6.1 is used to analyse the graph. This is used to build the graph, calculate betweenness centrality, density, connected components and cycles.

Storage uses SQLite, which is part of Python. This stores the nodes and edges locally. The Supabase client 2.31.0 optionally synchronizes the same data to a cloud PostgreSQL database.

The frontend was developed using React 18.3.1. The development server and build tool is Vite 5.4.21, with the app running on port 5173. Styling was done using Tailwind CSS 3.4.19. The interactive graph was produced using react-force-graph-2d 1.29.1. Icons were provided by lucide-react 0.363.0.

5. ALGORITHM

Rule-based entity extraction. Rule-based entity extraction. For each category, apply its regular expressions and keyword rules to the whole text, discard matches on the exclusion list or below a minimum length, normalize the matched text into an identifier, and add the entity as a node unless a node with that identifier already exists.

Rule-based relationship inference. The text is split into sentences (Section 8), and edges are created by six rules. (1) A person and a phone number that appear in the same sentence are linked with CALLS. (2) A person and an organization in the same sentence are linked with COMMANDS if the person is the first person extracted, and with OPERATES_OUT_OF otherwise. (3) Every organization is linked to every extracted account with FUNDS, without requiring textual evidence. (4) A vehicle and a location in the same sentence, or a person and a vehicle in the same sentence, are linked with OPERATES_OUT_OF. (5) Persons are chained in order of first appearance: the first links to the second with COMMANDS, and each later person links to the next with ASSOCIATE_OF. (6) If no edge has been created and there are at least two entities, consecutive entities are linked with ASSOCIATE_OF Within a sentence, mentions are matched by case-insensitive text containment, using a person's first or last name token, the organization, vehicle and location names, and any digit group of four or more characters from a phone number. Short names can therefore match unrelated text, and a phone number can be linked to a person through a shared digit group.

Graph measures. For a graph with n nodes and m directed edges, density is d = m / (n(n−1)). Betweenness centrality

of node v is g(v) = Σ σst(v) / σst over node pairs s ≠ v ≠ t, where σst is the number of shortest paths from s to t and σst(v) is the number of those passing through v [7]. NetworkX computes it with Brandes' algorithm [8].

Cycle detection. Directed simple cycles in the graph are enumerated and reported as circular-flow patterns when they contain two to six nodes.

6. CONCLUSION

This paper presented, a rule-based prototype that converts crime-related text into a network of persons, organizations, locations, phone numbers, vehicles and financial accounts, analyzes it with standard graph measures and simple pattern rules, and presents it through a web interface. Its main value at present is as a transparent and extensible baseline: the rules behind every output are explicit and open to inspection. Its main limitations are the heuristic nature of its extraction and relationship rules, the absence of quantitative evaluation, and the absence of any predictive or spatial component. Future work will build an annotated evaluation set, measure extraction and relationship quality, refine or replace weak heuristics, compare against learning-based extraction, and explore integration with spatial and temporal analysis, with attention to the ethical issues of bias and misuse.

7. ACKNOWLEDGEMENT

Every project requires the guidance and support of the experts for its completion. Therefore, we would like to take a moment to express our gratitude to all the individuals who played a crucial role in shaping and realizing this project.

Our heartfelt appreciation goes out to our project mentor, Prof. V. S, Gahire from the Department of Computer Science and Engineering at International Center of Excellence in Engineering & Management, CSN Maharashtra. Her guidance, unwavering support, and dedication throughout the project have been invaluable.

We'd also like to extend our thanks to our project coordinator, Prof. Sarita Bhopalkar, for her assistance in selecting this project and providing us with essential insights on how to present it effectively.

Our sincere recognition goes to all our professors at International Center of Excellence in Engineering & Management,, who provided us with valuable guidance and tips during the project's design phase. Their contributions have been so significant that it's challenging to individually acknowledge each one.

We are grateful for the support, from our Head of Department, Prof. S. A. Chechare. Her assistance was instrumental in various aspects of our project, and we genuinely appreciate her contributions through various channels.

I express my heartfelt gratitude for the warm and caring environment provided. This nurturing atmosphere and the excellent working conditions have been truly encouraging and appreciated.

8. REFERENCES

[1] J. J. Xu and H. Chen, "Criminal network analysis and visualization," Commun. ACM, vol. 48, no. 6, pp. 100–107, 2005, doi: 10.1145/1064830.1064834.

[2] J. J. Xu and H. Chen, "CrimeNet Explorer: A framework for criminal network knowledge discovery," ACM Trans. Inf. Syst., vol. 23, no. 2, pp. 201–226, 2005, doi: 10.1145/1059981.1059984.

[3] D. Weisburd, "The law of crime concentration and the criminology of place," Criminology, vol. 53, no. 2, pp. 133–157, 2015, doi: 10.1111/1745-9125.12070.

[4] H. Wang, D. Kifer, C. Graif and Z. Li, "Crime rate inference with big data," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2016, pp. 635–644, doi: 10.1145/2939672.2939736.

[5] P. Chen, H. Yuan and X. Shu, "Forecasting crime using the ARIMA model," in Proc. 5th Int. Conf. Fuzzy Systems and Knowledge Discovery (FSKD), 2008, pp. 627–630, doi: 10.1109/FSKD.2008.222.

[6] K. Lum and W. Isaac, "To predict and serve?," Significance, vol. 13, no. 5, pp. 14–19, 2016, doi: 10.1111/j.1740-9713.2016.00960.x.

[7] L. C. Freeman, "A set of measures of centrality based on betweenness," Sociometry, vol. 40, no. 1, pp. 35–41, 1977, doi: 10.2307/3033543.

[8] U. Brandes, "A faster algorithm for betweenness centrality," J. Math. Sociol., vol. 25, no. 2, pp. 163–177, 2001, doi: 10.1080/0022250X.2001.9990249.

[9] A. A. Hagberg, D. A. Schult and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in Proc. 7th Python in Science Conf. (SciPy 2008), 2008, pp. 11–15.

[10] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.