IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

AI Powered Criminal Network Analysis System

Abhishek Balasaheb Gaikwad[1], Rohit Rahul Chayal[2], Rohit Rajesh Ghallal[3], Pravej Raju Attar[4], Dr. H. B. Jadhav[5], Dr. H. B. Jadhav[6], Dr. H. B. Jadhav[7]
Student, Department of Computer Engineering[1234]
Faculty, Department of Computer Engineering[5-7]
Adsul's Technical Campus, Chas, Ahilyanagar, Maharashtra, India

Abstract: *Criminal investigations increasingly involve large volumes of linked data such as call records, financial transactions, case reports and social media activity, yet investigators still analyse these relationships largely by hand. This paper presents the design of an AI powered criminal network analysis system built in Python. The system ingests structured and unstructured records, extracts people, organisations, phones, locations and accounts using natural language processing (spaCy), builds a relationship graph (NetworkX), measures the importance of each actor with centrality metrics, detects groups with community detection, suggests hidden or missing links with machine learning, and presents the result as an interactive graph with an investigator-readable report. The system is organised as a linear five-stage pipeline that is modular and can be run on modest hardware. The paper describes the architecture, technology stack, processing flow, applications, advantages and limitations, and discusses ethical and privacy safeguards, a planned evaluation methodology and future work including graph neural networks, multilingual (Hindi and Marathi) text processing and temporal network analysis. No measured performance results are claimed in this paper.*

Keywords: Criminal Network Analysis, Social Network Analysis, Artificial Intelligence, Graph Mining, Link Prediction, Community Detection, Natural Language Processing, Python.

I. INTRODUCTION

Crime is rarely the work of isolated individuals. Drug trafficking, financial fraud, cyber-crime and organised theft are typically carried out by groups whose members communicate, share money and resources, and play different roles such as leader, broker or courier [1]. Understanding who is connected to whom, and which people hold the group together, is therefore central to modern investigation. Social network analysis (SNA) represents such groups as a graph in which people are nodes and their relationships are edges, and it has long been applied to criminal intelligence [2], [3].

In practice, however, investigators receive data in many forms: call detail records, bank statements, first information reports, witness statements and online posts. Linking these by hand is slow and error-prone, and important connections may be missed as the volume grows. Artificial intelligence can help by automatically extracting entities, building the network, ranking key actors and suggesting links that deserve human attention.

This paper presents an AI powered criminal network analysis system implemented in Python. It combines three well-established technologies, namely natural language processing, graph analytics and machine learning, into one lightweight decision-support tool. The system is intended to assist, and never to replace, the investigator: every output is a lead to be verified, not a finding of guilt.

The remainder of the paper is organised as follows. Section II reviews related work. Section III states the problem and objectives. Section IV describes the architecture and methodology. Section V lists the technology stack, Section VI presents the features, and Section VII discusses applications. Sections VIII and IX discuss advantages, limitations, ethics, evaluation and future scope, and Section X concludes the paper.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40830

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

II. LITERATURE REVIEW

The application of network analysis to criminal intelligence was examined early by Sparrow, who argued that centrality measures can reveal key players in criminal groups [2]. Xu and Chen described a framework for criminal network analysis and visualisation covering data extraction, subgroup detection, centrality analysis and visualisation [3], and Morselli studied how the structure of criminal networks relates to their efficiency and their vulnerability to disruption [4]. Krebs's well-known mapping of a terrorist cell showed how a small set of well-connected individuals can be identified from open-source data [5].

Centrality measures such as degree, closeness and betweenness [6] quantify how connected, how close and how "in between" a node is, while PageRank [7] ranks nodes by the importance of their neighbours. For finding groups, Girvan and Newman proposed community detection by edge betweenness [8], and Blondel et al. introduced the fast Louvain method based on modularity optimisation [9]. Standard text on networks and social network analysis is given in [10] and [11].

More recent work applies machine learning to graphs. Graph convolutional networks [12] and GraphSAGE [13] learn node representations that support node classification and link prediction, and Weber et al. applied graph convolutional networks to illicit transaction detection in a Bitcoin transaction graph [14]. For the language side, spaCy [15] provides tokenisation, part-of-speech tagging and named-entity recognition, and transformer models such as BERT [16] improve entity extraction on informal text. On the software side, NetworkX offers a mature Python library for building and analysing graphs [17].

A second body of work warns about the risks of data-driven policing. Studies of predictive policing show that models trained on historical police data can reproduce and amplify existing bias [18], and audits of commercial AI systems report unequal accuracy across demographic groups [19]. These observations motivate a transparent, explainable and human-supervised system such as the one proposed here.

III. PROBLEM STATEMENT AND OBJECTIVES

A. Problem Statement

Investigators must combine data from many sources to discover how suspects are connected. Manual link analysis does not scale to large case files, depends on individual experience, and can overlook indirect connections such as a broker who links two otherwise separate groups. Existing commercial intelligence platforms are expensive and closed, which limits their use in teaching, research and small agencies, while naive automated tools may produce unexplained or biased results that investigators cannot trust.

B. Objectives

- To extract entities and relationships automatically from structured and unstructured case data using NLP.
- To build a criminal network graph and identify key actors using centrality measures.
- To detect groups and sub-groups within the network using community detection.
- To suggest probable hidden links using link prediction, with an explanation for every suggestion.
- To present the network and results through an interactive visual dashboard and an exportable report.
- To embed privacy, access control and human-oversight safeguards in the design.

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Overall Architecture

The proposed system follows a linear pipeline in which each stage hands its output to the next, from raw records to an investigator report, as shown in Fig. 1. The five stages are data ingestion, entity and relationship extraction, graph construction, network analytics and visualisation with reporting. This modular design allows any stage to be replaced, for example swapping the entity recogniser, without affecting the others.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

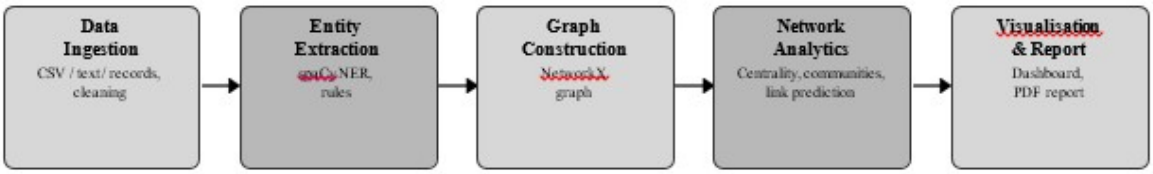


Fig. 1. Architecture of the AI powered criminal network analysis system

B. Process Flow

The operation of the system can be summarised in five steps.

• Ingest: records such as call logs, transaction tables and case narratives are loaded, cleaned and stored with their source.

• Extract: names, phone numbers, accounts, vehicles, organisations and places are identified, and a relationship is recorded whenever two entities co-occur in a call, a transaction or a sentence.

• Build: entities become nodes and relationships become weighted, typed edges in a graph; duplicate entities are merged.

• Analyse: centrality, community detection and link prediction are computed on the graph.

• Present: the graph and ranked results are shown on a dashboard, and a report listing each finding with its supporting evidence is generated.

C. Data Ingestion and Preprocessing

Structured sources (call detail records, bank transactions) are read with pandas and mapped to a common schema of source, target, type, time and weight. Unstructured sources (reports and statements) are converted to plain text, normalised to lower-case where appropriate and split into sentences. Each record keeps a reference to its origin so that every edge in the graph can be traced back to evidence. Records that cannot be parsed are logged and skipped instead of failing silently.

D. Entity and Relationship Extraction

Named-entity recognition with spaCy [15] identifies persons, organisations and places in text, while regular expressions capture phone numbers, account numbers and vehicle registrations. Name variants such as “R. Sharma” and “Rahul Sharma” are matched by string similarity and shared attributes before nodes are merged; uncertain matches are flagged for the investigator instead of being merged automatically. A relationship is created when two entities appear together in a call, a transaction or a sentence, and the relation type (called, paid, met, co-accused, family) is taken from the data field or from simple keyword rules. Because the approach is rule and model based, new entity types and relation rules can be added by registering a pattern and its handler.

E. Graph Construction and Network Analytics

The network is stored as a weighted graph G = (V, E), where an edge weight reflects the number or value of interactions. The following analyses are applied.

• Centrality. Degree centrality counts direct contacts; betweenness centrality measures how often a node lies on the shortest paths between others and highlights brokers; closeness shows how quickly a node can reach the rest of the network; PageRank ranks nodes by the importance of their neighbours [6], [7]. Together they give a ranked list of key actors.

• Community detection. The Louvain method [9] groups nodes so that modularity Q is maximised, revealing cells or sub-groups within a larger network.

• Link prediction. For node pairs that are not connected, features such as common neighbours, the Jaccard coefficient and the Adamic–Adar index are computed, and a supervised classifier (for example logistic regression or random forest) ranks the pairs most likely to be linked. Every suggestion is shown with the features that produced it, so that the investigator can see why it was raised.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

• Path and neighbourhood queries. The investigator can ask for the shortest connection between two persons or for all contacts within two steps of a suspect.

F. Visualisation and Reporting

The graph is displayed as an interactive node-link diagram in which node size shows importance and colour shows community. The investigator can filter by relationship type and time, click a node to see its evidence, and export a report that lists key actors, groups and suggested links together with the source records. Predicted links are drawn differently from confirmed ones so that the two are never confused.

V. TECHNOLOGY STACK

The system is implemented with freely available libraries, summarised in Table I.

TABLE I: TECHNOLOGIES USED IN THE PROPOSED SYSTEM

Component	Technology	Purpose
Core language	Python 3	Main programming language
Data handling	pandas	Loading and cleaning records
NLP	spaCy / NLTK	Entity and keyword extraction
Graph analysis	NetworkX	Graph building, centrality, communities
Machine learning	scikit-learn	Link prediction classifier
Storage	SQLite / PostgreSQL	Records and evidence store
Interface	Streamlit or Flask with PyVis / Plotly	Interactive graph dashboard

VI. FEATURES AND SUPPORTED ANALYSES

The system provides the capabilities listed in Table II. Each feature is available from the dashboard and each result is linked to its supporting records.

TABLE II: KEY FEATURES AND EXAMPLE QUERIES

Feature	Description	Example query
Entity extraction	Finds persons, phones, accounts, places in text and tables	“Load case file 12”
Network visualisation	Interactive graph with filters	“Show network for case 12”
Key-actor ranking	Ranks nodes by centrality	“Top 5 brokers”
Group detection	Detects communities in the graph	“List sub-groups”
Link suggestion	Ranks probable hidden links with reasons	“Suggest missing links”
Path search	Shortest connection between two persons	“Link between A and B”
Report export	Findings with evidence	“Export report”

VII. APPLICATIONS

• Law-enforcement support: organising case data and highlighting leads for human investigators.
• Financial crime and fraud analysis: tracing links between accounts, shell entities and individuals in money-laundering cases.
• Cyber-crime investigation: connecting accounts, devices and online groups.
• Training and research: a transparent, open platform for students and researchers to study criminal network analysis on synthetic or public data.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

VIII. ADVANTAGES AND LIMITATIONS

A. Advantages

- Free and open source, with no vendor lock-in.
- Automates the slow work of linking records and reveals indirect connections.
- Every result is explainable and traceable to source records.
- Modular, and runs on an ordinary laptop.

B. Limitations

- Results depend on the completeness and correctness of the input data; missing or false records distort the network.
- Entity resolution is imperfect, and wrongly merged or split identities change the graph.
- Rule-based extraction handles only known patterns, and accuracy on informal or code-mixed text is limited.
- Centrality and link prediction indicate structural importance, not guilt.
- No measured accuracy is reported in this paper; quantitative evaluation is planned as part of further work.

C. Comparison with Commercial Intelligence Platforms

Table III summarises, in qualitative terms, how the proposed system differs from commercial link-analysis platforms. The comparison is based on the publicly documented nature of such products, not on measured benchmarks, and is intended only to clarify the design space.

TABLE III: COMPARISON OF THE PROPOSED SYSTEM WITH COMMERCIAL PLATFORMS

Aspect	Commercial platforms	Proposed system
Source code	Typically closed	Open and fully modifiable
Cost	Licence usually required	Built on free libraries
Customisation	Limited to vendor features	Direct access to extraction rules and analytics
Transparency	Varies by product	Every result traced to evidence
Scale	Designed for very large deployments	Suited to case-level and academic use

The principal trade-off is scale versus transparency. Commercial platforms are built for very large data volumes and integrated data sources, whereas the proposed system gives up that scale in return for full control and explainability, which is valuable for teaching, experimentation and for deployments in which data must stay on the local machine.

D. Processing Walk-through

To make the flow concrete, Table IV traces four representative inputs through the pipeline. The traces illustrate the design and are not measured results; all names are fictitious.

TABLE IV: EXAMPLE TRACES THROUGH THE PIPELINE

Input record	Extracted entities / relation	Graph action	Output
Call log: A calls B, 14 times	A, B; called (weight 14)	Add weighted edge A–B	Edge shown in graph
Bank entry: A pays account of C	A, C; paid	Add typed edge A–C	Payment link listed
Statement: “B met C near the market”	B, C; met; place	Add edge B–C, place node	Evidence linked to edge
Query: top brokers	Whole graph	Compute betweenness	Ranked list of actors

In every case the same sequence applies: entities are extracted and resolved first, relationships are attached with their evidence, the graph is updated, and the analytics are recomputed. If a record yields no entities, it is logged and the system continues with the next record.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

E. Implementation Challenges and Design Considerations

• Entity resolution: the same person may appear under spelling variants, nicknames or several phone numbers. Matching by similarity plus shared attributes, with uncertain cases sent to the investigator, balances automation and correctness.
• Noisy and incomplete data: records contain typing errors and gaps. Validation at ingestion and confidence scores on edges limit the effect of errors.
• Scalability: centrality measures such as betweenness are expensive on large graphs. Sampling-based approximations or restricting analysis to a case subgraph keeps response time acceptable.
• Class imbalance in link prediction: true hidden links are rare compared with all unconnected pairs, so negative sampling and appropriate metrics (precision at k, area under the ROC curve) are required.
• Dense hubs: a shared public phone number or a bank can appear as a false hub. Down-weighting very high-degree generic nodes and allowing the investigator to exclude them resolves most cases.
• Dynamic data: relationships change over time. Timestamps on edges allow time-window filtering and a later move to temporal analysis.

F. Extensibility and Maintainability

Because each analysis is an ordinary Python function, a new capability such as a different centrality or a new relation rule is added by writing one function and registering it in a dictionary, which avoids long chains of conditional statements. The same separation makes testing easy: extraction rules can be checked on sample text, analytics can be run on small synthetic graphs with known answers, and the interface can be replaced by stubs in automated tests. Logging every extraction, merge decision and analysis run supplies the data needed for the evaluation described below.

G. Ethical, Legal and Privacy Considerations

A system that analyses people's connections carries serious risks and must be designed with safeguards from the start. Being connected to a criminal is not evidence of crime: a relative, a shopkeeper or a taxi driver may appear as a central node without any wrongdoing. Outputs are therefore presented as leads that need human verification and never as accusations or automated decisions.

Models trained on historical records can inherit the biases of how those records were collected [18], and AI systems can perform unequally across groups [19]. The design therefore calls for transparent rule-based and statistical methods whose results can be explained, regular review of outputs for bias, and no use of protected attributes as model features. Access should be restricted by role and logged, data should be stored securely and kept only as long as needed, and processing must respect applicable law, including India's Digital Personal Data Protection Act, 2023 [20], and the established procedures for evidence handling. Development and teaching should use synthetic or lawfully released data, since real case data requires proper authorisation.

H. Planned Evaluation Methodology

The evaluation planned for the next stage of the work is summarised in Table V. Experiments will use synthetic networks with known ground truth and, where permitted, publicly available network datasets, with noise added to simulate missing and false records.

TABLE V: METRICS PLANNED FOR EVALUATING THE SYSTEM

Metric	Definition	How it is measured
Entity precision / recall / F1	Correct entities found versus labelled entities	Annotated sample texts
Key-actor accuracy	Overlap of top-k ranked nodes with known leaders and brokers	Synthetic ground truth
Community quality	Modularity Q and agreement with known groups (NMI)	Graphs with planted groups

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

Link-prediction quality	Area under ROC curve and precision at k on held-out edges	Hidden-edge test split
Response time	Time from request to displayed result	Timestamps in the program log
Usability	Perceived ease of use	System Usability Scale [21]

Here, precision is the fraction of extracted items that are correct, recall is the fraction of true items that are found, and F1 is their harmonic mean. Reporting these figures for graphs of increasing size will show where the current pipeline is weakest and will give a baseline against which the planned extensions can be compared.

IX. FUTURE SCOPE

• Graph neural networks: use models such as GCN and GraphSAGE for node classification and stronger link prediction [12], [13].
• Multilingual support: process Hindi and Marathi reports, including code-mixed text.
• Temporal analysis: study how networks evolve and detect changes in structure over time.
• Cross-source data fusion: integrate call, financial and online data with consistent identity resolution.
These directions are developed below. Fig. 2 shows how the planned components would extend the present pipeline while keeping its modular, stage-by-stage structure.

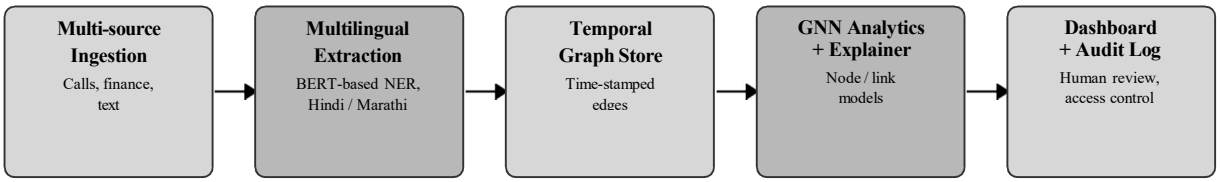


Fig. 2. Proposed extended architecture with multilingual extraction, temporal graph and explainable GNN analytics

A. Graph Neural Networks and Explainability

Graph neural networks learn node features from both attributes and structure and often outperform hand-built features for link prediction [12], [13]. Because they are harder to interpret, they should be paired with explanation techniques and kept optional, with the simpler transparent measures retained as a fallback and reference.

B. Multilingual and Code-Mixed Text

Many reports and statements in Maharashtra are written in Marathi or Hindi, or mix them with English. Supporting them requires entity-recognition models for those languages, handling of Devanagari script and of Latin transliteration, and name matching that tolerates spelling variants. Multilingual transformer models such as BERT-based models [16] are a candidate, and their accuracy must be verified on representative text before use.

C. Temporal Network Analysis

Storing a timestamp on every edge allows the network to be viewed in time windows, showing when a group formed, when a new broker appeared, or how activity changed before an incident. Temporal analysis requires careful treatment of incomplete data, because absence of records in a period does not mean absence of contact.

D. Data Fusion and Integration

Combining call, financial and online data gives a fuller picture but raises identity-resolution and legal questions. Each source should keep its own provenance and legal basis, and integration should proceed only where lawful and authorised.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

E. Generative Language Models

A language model could summarise a case file or draft a report from the graph. This would save time, but generated text can be wrong or invented and may expose sensitive data to an external service. It should therefore remain optional, be clearly labelled, run locally where possible, and never replace verification against the source records.

TABLE VI: PROPOSED ROADMAP OF ENHANCEMENTS

Phase	Enhancement	Technology	Expected benefit
1	Quantitative evaluation	Synthetic graphs, metrics in Table V	Measured baseline
2	Multilingual extraction	Hindi and Marathi NER models	Wider data coverage
3	GNN link prediction	GCN / GraphSAGE	Better hidden-link ranking
4	Temporal analysis	Time-stamped graph store	Evolution of networks

X. CONCLUSION

This paper presented the design of an AI powered criminal network analysis system built in Python. By chaining data ingestion, natural language extraction, graph construction, network analytics and interactive visualisation, the system helps investigators see who is connected to whom, which actors hold a network together, and which links deserve attention. The project shows that a useful and explainable analysis tool is within reach using open and free libraries, and it lays the groundwork for more capable, accountable and fair data-supported investigation.

The main contribution of the work is a clear, modular reference design that places human oversight, explainability and privacy at its centre. Because each stage has a narrow interface, students and developers can replace the entity recogniser, add an analytic or change the interface without touching the rest of the system. The discussion of limitations, ethics and evaluation is intended to make clear what the present system can and cannot do, and to give later work a fair basis for comparison.

Future work will concentrate on measuring extraction and prediction accuracy on synthetic and public data, on supporting Indian languages, and on graph neural network and temporal extensions, always under human supervision.

ACKNOWLEDGMENT

The authors thank the Department of Computer Engineering, Adsul's Technical Campus, Ahilyanagar, for providing the facilities and guidance that made this project possible.

REFERENCES

[1]. C. Morselli, Inside Criminal Networks. New York, NY, USA: Springer, 2009.
[2]. M. K. Sparrow, "The application of network analysis to criminal intelligence: An assessment of the prospects," Social Networks, vol. 13, no. 3, pp. 251–274, 1991.
[3]. J. Xu and H. Chen, "Criminal network analysis and visualization," Commun. ACM, vol. 48, no. 6, pp. 100–107, 2005.
[4]. C. Morselli, C. Giguère, and K. Petit, "The efficiency/security trade-off in criminal networks," Social Networks, vol. 29, no. 1, pp. 143–153, 2007.
[5]. V. E. Krebs, "Mapping networks of terrorist cells," Connections, vol. 24, no. 3, pp. 43–52, 2002.
[6]. L. C. Freeman, "Centrality in social networks: Conceptual clarification," Social Networks, vol. 1, no. 3, pp. 215–239, 1978.
[7]. L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, 1999.
[8]. M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," Proc. Natl. Acad. Sci. U.S.A., vol. 99, no. 12, pp. 7821–7826, 2002.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

[9]. V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," J. Stat. Mech., 2008, P10008.
[10]. M. E. J. Newman, Networks: An Introduction. Oxford, U.K.: Oxford University Press, 2010.
[11]. S. Wasserman and K. Faust, Social Network Analysis: Methods and Applications. Cambridge, U.K.: Cambridge University Press, 1994.
[12]. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. ICLR, 2017.
[13]. W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Proc. NeurIPS, 2017.
[14]. M. Weber et al., "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," arXiv:1908.02591, 2019.
[15]. M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017.
[16]. 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.
[17]. 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, pp. 11–15.
[18]. K. Lum and W. Isaac, "To predict and serve?," Significance, vol. 13, no. 5, pp. 14–19, 2016.
[19]. J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in Proc. Conf. Fairness, Accountability and Transparency (PMLR 81), 2018, pp. 77–91.
[20]. Government of India, "The Digital Personal Data Protection Act, 2023."
[21]. J. Brooke, "SUS: A quick and dirty usability scale," in Usability Evaluation in Industry, London, U.K.: Taylor & Francis, 1996, pp. 189–194.