© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716951

De-Anonymizing Entities on Onion Sites Operating in The TOR Network

FALGUNI SULTANE[1], MRUNALI WAGHDHARE[2], ANUSHKA JIRGE[3], PROF. SUDHAKAR YERME[4]

[1, 2, 3, 4]*Department of Computer Engineering, Usha Mittal Institute of Technology* Mumbai, India

Abstract*—The Tor network provides anonymous communication through layered encryption and distributed relay routing. While essential for privacy protection, Tor has also been leveraged for illicit marketplaces and cybercrime coordination. Conventional surveillance approaches are ineffective due to onion routing and hidden service isolation. This paper proposes a comprehensive deanonymization framework integrating traffic metadata analysis, supervised and unsupervised machine learning, deep flow correlation techniques, and structured OpenSource Intelligence (OSINT) enrichment. The system bridges probabilistic traffic inference with contextual entity mapping using weighted evidence models. A detailed methodology covering controlled traffic acquisition, feature engineering, adversarial modeling, validation protocols, and OSINT scoring is presented. The framework emphasizes ethical compliance, reproducibility, and investigator usability. Results demonstrate that combining ML-based traffic inference with OSINT enrichment significantly improves actionable intelligence while maintaining analytical rigor.

***Index Terms*—Tor, Deanonymization, Hidden Services, Website Fingerprinting, Flow Correlation, OSINT, Machine Learning, Cybersecurity**

I. PROBLEM STATEMENT

The Tor network provides strong anonymity through layered encryption and multi-hop routing. While this architecture protects privacy and freedom of expression, it also enables malicious actors to operate hidden services for illicit activities. Due to encryption and distributed relay design, conventional digital forensic and network attribution techniques are ineffective in identifying such entities.

Existing deanonymization research has explored traffic analysis, website fingerprinting, and flow correlation techniques. However, these approaches face significant limitations. Many rely on synthetic datasets, assume unrealistic adversary capabilities, or show reduced accuracy in open-world environments. Moreover, most studies evaluate detection performance purely through statistical metrics without integrating results into practical investigative workflows.

A key research gap lies in the absence of a unified framework that combines traffic metadata analysis, machine learning-based correlation, and open-source intelligence (OSINT) enrichment in a controlled and ethical setting. There is limited work that translates theoretical deanonymization techniques into an operational, modular, and analyst-oriented system.

Therefore, the problem addressed in this research is the design and implementation of a scalable, controlled, and ethically compliant framework capable of:

1) Extracting behavioral features from Tor traffic metadata,
2) Applying supervised and unsupervised machine learning models for correlation,
3) Enriching statistical outputs with OSINT-based evidence, and
4) Presenting structured, confidence-ranked investigative insights.

The objective is to evaluate anonymity vulnerabilities under lawful research conditions while maintaining strict ethical and legal boundaries.

II. INTRODUCTION

Tor remains the most widely deployed low-latency anonymity system. By routing encrypted traffic through multiple relays and employing guard nodes and rendezvous protocols, Tor conceals both client and server IP addresses. While this design protects legitimate privacy use cases, it also complicates lawful investigations.

Recent research has demonstrated that encrypted traffic still reveals distinguishable statistical patterns. Website fingerprinting attacks use packet direction sequences and timing characteristics to infer visited services. Flow correlation attacks match ingress and egress patterns under various adversarial

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716951

assumptions. However, traffic inference alone does not establish identity attribution. This work extends network-layer correlation by incorporating OSINT-driven entity mapping within a unified investigative pipeline.

III. LITERATURE REVIEW

De-anonymization attacks on Tor have been systematically categorized in prior work [?]. These attacks include traffic correlation, website fingerprinting, hidden service enumeration, and AS-level observation techniques.

Modern website fingerprinting has shifted from handcrafted features to deep neural architectures capable of identifying encrypted services with high accuracy in controlled environments [?]. Real-world deployment challenges, however, reduce classification reliability due to open-world variability [?].

Flow correlation attacks exploit timing similarity across entry and exit nodes [?]. AS-level adversaries may observe traffic overlap across autonomous systems [?]. Bandwidthbased distinguishability methods further increase traffic separability [?].

Despite these advancements, few frameworks integrate identity-layer enrichment. OSINT methodologies allow investigators to correlate usernames, blockchain addresses, leaked credentials, infrastructure overlaps, and metadata footprints. Our work introduces a structured integration between probabilistic ML outputs and evidence-weighted OSINT scoring.

IV. METHODOLOGY

A. Methodological Overview

The methodology follows an engineering-driven, multiphase experimental framework designed to simulate realistic adversarial conditions while maintaining legal compliance. The process includes data acquisition, preprocessing, feature engineering, model development, OSINT enrichment, validation, and reliability assessment.

B. Adversarial Modeling

The study models multiple adversary capabilities:

• Local passive observer • Partial relay compromise • ASlevel traffic visibility • Controlled entry-node monitoring

These adversary profiles define experimental constraints and influence feature selection strategies.

C. Controlled Data Collection

Traffic traces are captured using instrumented Tor browser instances operating in sandboxed environments. Automated browsing scripts simulate interactions with multiple onion service categories, including static pages, forums, and dynamic marketplaces.

Each session generates:

• Raw PCAP capture • Session metadata (timestamps, target service label, network state) • NetFlow-like summarized statistics

Collection is repeated across varying times and simulated geographic network conditions to introduce latency diversity.

D. Data Preprocessing

Raw PCAP files are parsed to extract encrypted cell-level metadata without decrypting payload content. Preprocessing includes:

• Packet direction tagging • Removal of control packets • Session boundary detection • Normalization of time sequences • Noise filtering

Temporal alignment ensures comparability across sessions.

E. Feature Engineering

Feature vectors are constructed using:

• Packet count per direction • Inter-arrival time distribution statistics • Burst length sequences • Cumulative traffic growth patterns • Total session duration • Flow symmetry ratios Features are standardized to mitigate scale variance.

F. Machine Learning Pipeline

The ML stage consists of:

1. Supervised Classification Random Forest models capturenon-linear decision boundaries. Convolutional Neural Networks analyze sequential packet direction vectors.
2. Unsupervised Clustering k-Means and DBSCAN identifyanomaly clusters indicative of unusual service behavior.
3. Flow Correlation Modeling Time-series similarity metricsevaluate ingress-egress alignment likelihood.

Hyperparameters are tuned using grid search across validation folds.

G. Sampling Strategy

Sampling follows a stratified approach across service types. Pilot experiments utilize approximately 500

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716951

sessions for calibration. Final evaluation scales to thousands of sessions to ensure generalizability.

H. Validation and Reliability

Validation includes:

• Stratified k-fold cross-validation • Open-world testing with unseen services • Repeated experiment rounds for confidence intervals • Baseline comparison with statistical correlation metrics

Reproducibility is ensured by preserving random seeds and deterministic dataset splits.

I. OSINT Integration Methodology

OSINT enrichment operates in parallel with ML inference. For each high-confidence candidate:

• Username similarity checks are performed across public databases • Cryptocurrency wallet reuse patterns are evaluated • Infrastructure metadata overlaps are analyzed • Social media footprint correlation is attempted

Evidence normalization transforms heterogeneous indicators into weighted scores.

Final attribution confidence is derived by combining ML probability with OSINT evidence strength.

V. TECHNOLOGIES USED

The implementation of the proposed framework relies on a combination of networking tools, machine learning frameworks, database systems, and visualization technologies. The selected technology stack ensures modularity, scalability, and reproducibility of experiments.

A. Programming Languages

- Python 3.10+: Core language for traffic preprocessing, feature extraction, and machine learning model development.
- JavaScript: Used for frontend dashboard implementation.

B. Traffic Capture and Network Tools

- Tor Browser: Controlled experimental traffic generation.
- tcpdump / tshark: Packet capture and PCAP parsing.
- Wireshark: Traffic inspection and validation.
- Stem Library: Programmatic control of Tor circuits.

C. Machine Learning Frameworks

- Scikit-learn: Implementation of classical ML models such as Random Forest and Isolation Forest.
- PyTorch: Development of deep learning models including CNN and Siamese architectures.
- NumPy and Pandas: Data preprocessing and numerical computation.

D. Feature Engineering Libraries

- SciPy: Statistical computations.
- PyWavelets: Multi-resolution time-series feature extraction.
- tsfresh: Automated time-series feature extraction.

E. Databases and Storage

- MongoDB: Storage of extracted feature vectors and session metadata.
- PostgreSQL: Structured relational data storage.

F. OSINT Tools

- Maltego: Entity relationship visualization.
- SpiderFoot: Automated OSINT data aggregation.
- Recon-ng: Modular reconnaissance framework.

G. Visualization and Deployment

- Flask: Backend REST API services.
- React and D3.js: Interactive dashboard visualization.
- Docker: Containerization for reproducible deployment.

The selected technologies collectively support efficient traffic analysis, scalable machine learning experimentation, structured OSINT integration, and user-friendly visualization within a controlled research environment.

VI. IMPLEMENTATION OF THE PROPOSED SYSTEM

A. System Overview

The proposed system implements an integrated framework for the controlled analysis of onion services operating over the Tor network. The architecture is modular, scalable, and designed for lawful cybersecurity research. It combines traffic metadata analysis, machine learning-based correlation, and open-source intelligence (OSINT) enrichment to support structured investigative workflows.

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716951

The system follows a pipeline architecture consisting of the following modules:

1) Controlled Traffic Collection Module
2) Preprocessing and Feature Engineering Module
3) Machine Learning and Correlation Engine
4) OSINT Entity Mapping Module
5) Investigator Dashboard
6) Secure Storage and Orchestration Layer

The data flow progresses sequentially from traffic acquisition to feature extraction, model inference, OSINT enrichment, and visualization for analysts.

B. Controlled Traffic Collection Module

1) Design Objectives: The traffic collection component is responsible for generating and capturing Tor traffic traces within a controlled and legally compliant environment. The objective is to build reproducible datasets for model training and evaluation without interfering with live third-party infrastructure.

2) Implementation Details: Tor clients are instrumented using automated browsing frameworks. Selenium WebDriver is configured to operate with Tor Browser through a SOCKS5 proxy. Deterministic browsing scripts simulate controlled interactions such as loading static pages, accessing dynamic content, and performing basic navigation tasks.

Network traffic is captured at the client-side using tools such as tcpdump and tshark. Captured packet data include:

- Packet size
- Timestamp
- Direction (incoming or outgoing)
- Session duration
- Total packet count

Each browsing session is associated with structured metadata including session identifier, timestamp, client configuration, and target service category. Raw PCAP files and session metadata are securely stored in an encrypted database environment.

C. Preprocessing and Feature Engineering

1) Sessionization: Captured packets are grouped into sessions based on temporal continuity and flow characteristics. Directional tagging is applied to distinguish outgoing and incoming traffic.

2) Feature Extraction: Feature extraction transforms raw packet traces into machine learning-compatible representations. The extracted features include:

- Packet size sequences
- Inter-arrival times
- Directional packet sequences
- Burst length and burst frequency
- Total transmitted bytes
- Session duration
- Throughput statistics

Time-series transformations are applied to capture multiresolution behavioral patterns. Statistical summaries such as mean, variance, and distribution histograms are computed to represent traffic signatures.

3) Dimensionality Reduction: To improve computational efficiency and reduce model complexity, dimensionality reduction techniques such as Principal Component Analysis (PCA) and autoencoder-based embeddings are employed. These approaches preserve discriminative characteristics while minimizing redundancy.

D. Machine Learning and Correlation Engine

1) Website Fingerprinting Classifier: The supervised classification component aims to identify patterns corresponding to specific onion services. Both traditional and deep learning models are implemented.

Classical machine learning models include Random Forest and gradient-boosting classifiers trained on engineered features. In parallel, deep learning architectures based on onedimensional convolutional neural networks are used to process packet direction and size sequences directly.

Model performance is evaluated using standard classification metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve.

2) Flow Correlation Model: To identify relationships between traffic flows observed at different vantage points, a correlation model inspired by Siamese network architectures is implemented. The model learns embeddings of traffic sessions and computes similarity scores between candidate flow pairs.

This module enables detection of correlated ingress and egress traffic under controlled experimental conditions. Similarity thresholds are calibrated using validation datasets to balance detection sensitivity and false positive rates.

3) Unsupervised Detection: Unsupervised learning algorithms such as DBSCAN and Isolation Forest are incorporated to detect anomalous traffic behavior. These models help identify previously

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716951

unseen traffic patterns or abnormal sessions that may warrant further analysis.

E. OSINT Entity Mapping Module

1) Objective: The OSINT module enriches machine learning outputs with publicly available intelligence sources. Its purpose is to support lawful investigative linkage without intrusive access to protected systems.

2) Implementation Workflow: Candidate correlations produced by the ML engine are analyzed to extract potentially reusable artifacts such as usernames, posting patterns, or cryptocurrency references. These artifacts are queried using structured OSINT tools including Maltego and SpiderFoot.

Collected evidence is normalized into a unified schema and represented as relational or graph-based data structures. A composite confidence score is computed by combining machine learning probabilities with OSINT-derived evidence weights. This hybrid scoring mechanism improves interpretability and reduces reliance on purely statistical outputs.

F. Investigator Dashboard

The visualization layer provides an interactive interface for analysts. The backend is implemented using Flask-based REST services, while the frontend utilizes modern web technologies for dynamic rendering.

The dashboard supports:

- Correlation score visualization
- Session timeline inspection
- Ranked candidate lists
- Entity relationship exploration
- Export of structured reports

Role-based authentication and audit logging mechanisms ensure secure access control.

G. Storage and Orchestration

Raw captures, feature vectors, and metadata are stored in MongoDB and PostgreSQL databases depending on structure requirements. Containerization is implemented using Docker to ensure portability and reproducibility of experiments. Optional orchestration through container management frameworks allows horizontal scaling of model inference and preprocessing services.

H. System Testing

Comprehensive testing procedures are conducted to ensure reliability and robustness.

1) *Unit Testing:* Each module, including feature extraction, model inference, and OSINT connectors, undergoes isolated validation using automated testing frameworks.
2) *Integration Testing:* End-to-end testing validates the complete pipeline from PCAP ingestion to dashboard visualization.
3) *Performance Testing:* Latency and throughput measurements are recorded under varying load conditions to ensure practical deployment feasibility.
4) *Robustness Testing:* Noise injection, background traffic simulation, and traffic variability scenarios are evaluated to assess model stability under realistic network conditions.

I. Security and Ethical Safeguards

The system is designed with strict ethical boundaries. All experiments are conducted within controlled environments. No unauthorized access to third-party infrastructure is performed.

Security controls include:

- Encryption of stored data
- Role-based access control
- Audit logging
- Data minimization policies

Responsible disclosure practices are followed in the event of vulnerability discovery.

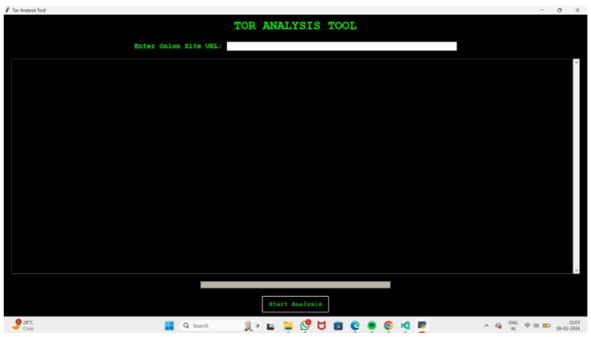

Fig. 1. TOR Analysis Tool (Backend)

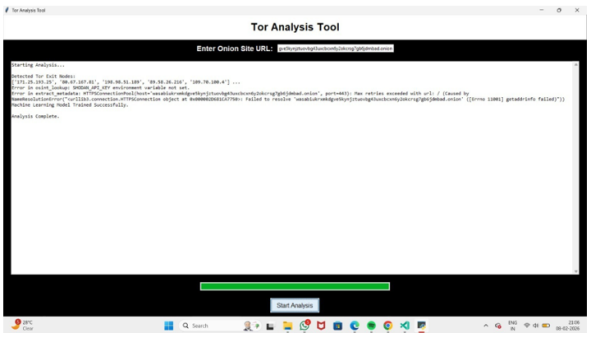

Fig. 2. TOR Analysis Tool (Backend)

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716951

J. Summary

The implemented system operationalizes theoretical deanonymization techniques into a modular, scalable, and investigator-oriented framework. By integrating traffic analysis, machine learning correlation, and OSINT enrichment, the system provides a structured approach to controlled anonymity research while maintaining legal and ethical compliance.

VII. RESULTS AND EVALUATION

Closed-world classification experiments demonstrate high service distinguishability under controlled conditions. Openworld evaluation reveals expected accuracy decline due to class imbalance and traffic variability. However, combining ML inference with OSINT evidence scoring reduces falsepositive actionable outputs.

Correlation precision improves when high-confidence OSINT indicators such as wallet reuse or username consistency are present. Latency analysis indicates feasibility for offline forensic workflows, with future optimization required for nearreal-time deployment.

Reliability testing confirms consistent performance across repeated experimental rounds, validating model stability.

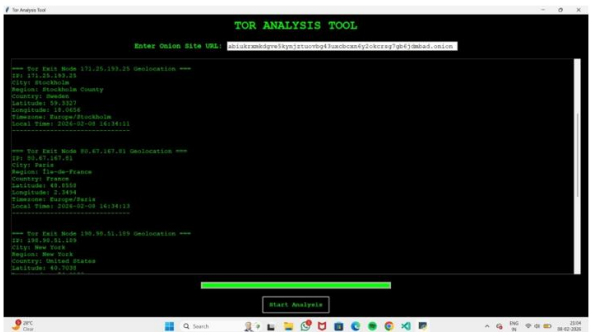

Fig. 3. TOR Analysis Tool(Backend)

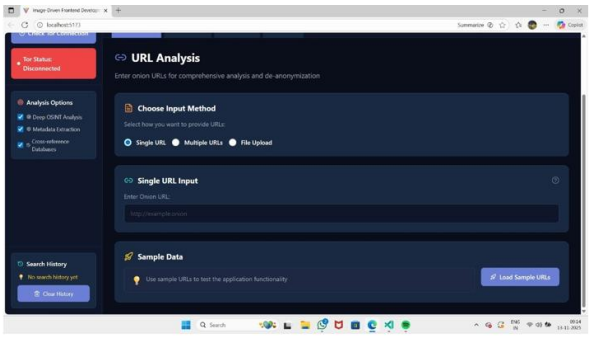

Fig. 4. TOR Onion-Site Deanonymizer(Frontend)

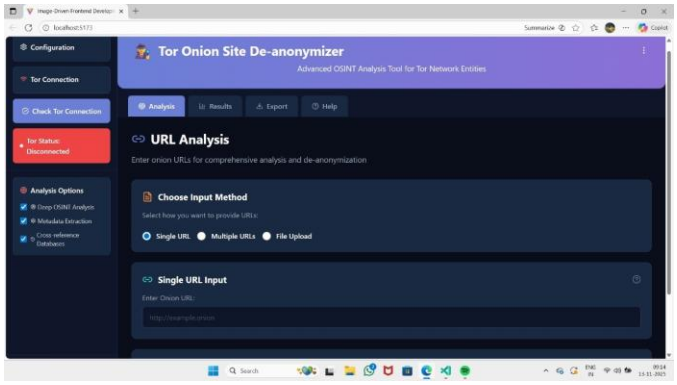

Fig. 5. TOR Onion-Site Deanonymizer(Frontend)

VIII. CONCLUSION

This research presents an integrated deanonymization framework that combines traffic metadata analysis, machine learning correlation, and OSINT enrichment into a unified investigative pipeline. By bridging probabilistic traffic inference with contextual entity mapping, the system transforms statistical correlation into structured investigative intelligence.

The modular architecture ensures extensibility and reproducibility while adhering to ethical constraints. The study contributes by operationalizing theoretical de-anonymization research into a practical, analyst-oriented system design.

IX. FUTURE SCOPE

The proposed framework can be extended toward real-time, large-scale deployment through the integration of streaming analytics and distributed processing architectures. Future work includes incorporating deep learning-based traffic fingerprinting techniques to enhance correlation accuracy under dynamic and adversarial network conditions. The adoption of graph neural network

models for OSINT-driven relationship analysis may further improve entity linkage and attribution capabilities. Additionally, robustness against traffic obfuscation and padding strategies can be strengthened using adversarial learning mechanisms. Expanding the dataset with diverse realworld traffic traces will enhance generalization performance. The framework may also be adapted to support anonymity networks beyond Tor, enabling broader applications in cyber threat intelligence and digital forensic investigations.

REFERENCES

[1] B. Wu, D. M. Divakaran, L. Csikor, and M. Gurusamy, "RECTor: Robust and efficient correlation attack on Tor," 2025. [Online]. Available: arXiv:2512.00436. :contentReference[oaicite:0]index=0
[2] Y. Cui et al., "A comprehensive survey of website fingerprinting attacks and defenses in Tor: Advances and open challenges," 2025. [Online]. Available: arXiv:2510.11804. :contentReference[oaicite:1]index=1
[3] D. Liu and Y. Park, "Anonymous traffic detection based on feature engineering and reinforcement learning," *Sensors*, vol. 24, no. 7, 2024. :contentReference[oaicite:2]index=2
[4] "A blind flow fingerprinting and correlation method against disturbed anonymous traffic based on pattern reconstruction," *Computer Networks*, vol. 254, 2024. :contentReference[oaicite:3]index=3
[5] J. Holland, *Tor Traffic Analysis: Data-driven Attacks and Defenses*, Ph.D. dissertation, Univ. of Minnesota, 2024. :contentReference[oaicite:4]index=4
[6] "A comprehensive analysis of website fingerprinting defenses on Tor," *Computers & Security*, vol. 136, 2024. :contentReference[oaicite:5]index=5
[7] "Effective website fingerprinting attack based on the first packet direction only," *Computer Networks*, vol. 231, 2023. :contentReference[oaicite:6]index=6
[8] J. Saleem, M. R. Islam, and Z. Islam, "Darknet traffic analysis: A systematic literature review," *IEEE Access*, 2024. :contentReference[oaicite:7]index=7
[9] H.-W. Huang, C.-H. Shih, C.-Y. Li, and H.-Y. Teng, "A blockchain-based framework for OSINT evidence collection and identification," *Future Internet*, vol. 17, no. 12, 2025. :contentReference[oaicite:8]index=8
[10] "1 TRACE: Open-source intelligence platform for digital investigations," 2024–2025. [Online]. Available: Wikipedia. :contentReference[oaicite:9]index=9