

BTC-FUSION: Fusing Bitcoin Peer-to-Peer Network Metadata with On-Chain Analytics for Explainable Lead Generation and Risk-Controlled Attribution

Harshit Kumar Sengar

harshitkumarsengar_cse_25a02072@dtu.ac.in

Delhi Technological University

Yury Trochin

Delhi Technological University

Tanav Dwivedi

Delhi Technological University

Akshat Prasad

Delhi Technological University

Ashwani Kumar

Delhi Technological University

Tanvi Goel

Delhi Technological University

Research Article

Keywords: Bitcoin forensics, Network-layer deanonymisation, Anti-money laundering, Selective prediction, Risk control, Explainability

Posted Date: October 20th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11236577/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

BTC-FUSION: Fusing Bitcoin Peer-to-Peer Network Metadata with On-Chain Analytics for Explainable Lead Generation and Risk-Controlled Attribution

Harshit Kumar Sengar^{1*}, Yury Trochin¹, Tanav Dwivedi¹,
Akshat Prasad¹, Ashwani Kumar¹, Tanvi Goel¹

^{1*}Department of Computer Science and Engineering, Delhi Technological University, Shahbad Daulatpur, Main Bawana Road, Delhi, 110042, India.

*Corresponding author(s). E-mail(s):
harshitkumarsengar_cse_25a02072@dtu.ac.in;

Abstract

Investigating Bitcoin-enabled crime requires the analysis of public blockchain transactions and peer-to-peer (P2P) network metadata that records the IP addresses announcing transactions. Joining them can prioritise leads, but a relay or a shared exit is easily mistaken for the originator. BTC-FUSION is an offline system that clusters addresses with a CoinJoin-aware resolver, ranks entities with gradient-boosted trees, explains each lead, and names an IP address only when a risk-controlled answering policy bounds the error among its answers at 5% with 95% confidence. No public dataset pairs both layers with labels, so the network side is evaluated on a leakage-audited simulator, over five independently generated captures per profile, and the chain side on the real Elliptic and Elliptic++ datasets. At illicit prevalences of 0.27% to 0.54%, the ranking reaches an average precision of 0.1859 ± 0.0760 across captures, and every tuned design choice is made on validation data. The policy answers at least 91.4% of gradable entities with an error of at most 2.56% on all ten captures. A security analysis shows where it fails: actors on hosting infrastructure are answered wrongly, a frozen policy exceeds its error target when coverage falls, and forged first announcements redirect most attributions to an attacker. On Elliptic, the classifier collapses after a dark-market closure. The results support selective lead generation, not operational attribution accuracy.

Keywords: Bitcoin forensics, Network-layer deanonymisation, Anti-money laundering, Selective prediction, Risk control, Explainability

1 Introduction

Bitcoin’s ledger is public, but its addresses carry no identity. Investigators therefore combine ledger heuristics, above all the assumption that addresses spent together as inputs share an owner [1], with outside information that ties some addresses to real parties. A second, lower-level source of evidence exists: the peer-to-peer (P2P) network over which transactions are broadcast. Early measurement studies showed that a well-connected listener could sometimes link a transaction to the IP address that first relayed it [2, 3]. Stronger relay defences such as Dandelion++ [4] have been proposed, and recent versions of Bitcoin Core offer an optional private broadcast mode that sends transactions submitted through the `sendrawtransaction` interface only over Tor or I2P.

For an investigator, a wrong link is more harmful than no link, because it points to someone else. Measurements on real traffic show that the risk is not hypothetical: Neudecker and Hartenstein found that fewer than 8% of address clusters were conspicuously associated with a single IP address [5]. A system that joins the two layers should therefore be judged by how often it is wrong when it gives an answer, and by whether it is able to decline, rather than by accuracy alone.

This paper describes BTC-FUSION, a prototype that takes bulk records carrying both layers and produces a ranked queue of explained case files for a human analyst. The work originated in Problem Statement 26146 of Smart India Hackathon 2026, posed by India’s National Technical Research Organisation (NTRO), which asks for a system that correlates network-layer observations (IP address, port and timing) with blockchain-layer data (addresses, transaction identifiers and amounts) and produces prioritised, explainable investigative leads.

The central obstacle is data. We are not aware of a public dataset that pairs Bitcoin transactions with labelled network observations: Elliptic [6] and Elliptic++ [7] contain only chain-side information. We therefore evaluate the full system on a seeded simulator whose output is audited for label leakage, repeat every synthetic measurement on independently generated datasets, and validate the chain-side components separately on Elliptic and Elliptic++. The synthetic results are self-referential, because the same authors wrote the generator that produced them; where that can be bounded we bound it, and where it cannot we say so.

The main contributions are as follows.

- (1) We formulate entity-to-IP attribution as selective prediction, allowing the system to answer or decline while bounding the error rate among answered cases with a finite-sample guarantee.
- (2) We develop a nine-stage offline cross-layer pipeline for CoinJoin-aware entity resolution, ranking, selective attribution, and exact SHAP-based explanation, processing 2,470,860 records in 42.58 s on a laptop.

- (3) We conduct a replicated and leakage-audited evaluation using five independently generated captures per profile, evidence-layer ablations, and validation-only selection of all tuned design choices.
- (4) We evaluate the robustness of the proposed system against relay injection, withheld announcements, address rotation, reduced observation coverage, and mimicry of licit network behaviour.
- (5) We validate the chain-side classifier on Elliptic and Elliptic++ and show that a published Elliptic++ result is sensitive to positional rather than temporal data splitting.

Section 2 reviews related work, Sections 3 and 4 state the problem and the system, Section 5 the data and protocol, Sections 6 and 7 the results on synthetic and real data, and Section 8 the security analysis; Section 9 concludes.

2 Background and related work

Address clustering.

Meiklejohn et al. [1] formalised the common-input-ownership heuristic and a change-address heuristic and used them to trace flows to known services, and Harrigan and Fretter [8] traced the effectiveness of these heuristics to address reuse and high-centrality super-clusters. CoinJoin transactions break the common-input assumption on purpose, because many owners each contribute an input and take an identical output; the CoinJoin guard used here follows the transaction structure described by Stütz et al. [9].

Network-layer inference.

Koshy et al. [2] used anomalous relaying behaviour in months of collected traffic to map some Bitcoin addresses to the IP addresses of their likely owners, and Biryukov et al. [3] showed that an attacker with modest resources could link clients’ transactions to their IP addresses. Dandelion++ [4] proposed relaying with formal anonymity guarantees. Juhász et al. [10] built a Bayesian model that links addresses to originating IP addresses using more than a hundred modified clients as observers, and Zhang et al. [11] used traffic features with semi-supervised learning to decide whether a monitored node originated a transaction. Neudecker and Hartenstein [5], the study closest to our question, associated blockchain clusters with IP information from observed flooding and concluded that for most users network information cannot easily facilitate address clustering.

Learning to detect illicit activity.

Weber et al. [6] released the Elliptic dataset and found that a random forest outperformed graph convolutional networks; Elmougy and Liu [7] extended it to wallet addresses as Elliptic++. The evaluation practices behind these benchmarks have been questioned: Šafář et al. [12] showed that Elliptic’s features are undocumented and its common splits leak information, and Maganti [13] traced the apparent advantage of graph neural networks to exposure to test-period structure. SHAP values can be computed exactly for tree ensembles [14], and synthetic labels can encode shortcuts that a detector exploits, the leakage problem formalised by Kaufman et al. [15].

Risk control.

Learn then Test [16] treats the control of a risk such as the error rate among answers as a multiple-testing problem over a grid of thresholds. The answering policy in this paper is a simple instance of it, using Clopper-Pearson bounds [17] with a Bonferroni correction.

How the closest work relates to this paper.

The points below summarise the most closely related work and comment on its relevance to the proposed method.

- *Ledger heuristics and CoinJoin structure* [1, 9]. Comment: these establish the clustering heuristics, but a CoinJoin breaks the common-input assumption and no network data are used; adopted here for entity resolution, with a several-owners CoinJoin guard (Section 4.2).
- *Network observations and address associations* [2, 3, 5]. Comment: transactions can sometimes be linked to the IP addresses that relayed them, but on real traffic conspicuous single-IP associations are rare; this motivates candidate generation and a policy that is able to decline (Section 4.6).
- *Bayesian inference with thresholded acceptance* [10]. Comment: pairings are accepted above a posterior threshold using more than a hundred modified observer clients; here the acceptance threshold is instead fitted per policy key to a stated error target with a finite-sample bound (Section 4.6).
- *Transaction and address classification* [6, 7]. Comment: chain-side benchmarks whose reported performance depends on how the data are split; used here for real-data validation under temporal and address-disjoint splits, with a reproduction of a published result (Section 7).
- *Data leakage* [15]. Comment: synthetic data give complete labels but can encode shortcuts; here every capture is gated by a leakage audit before any result is accepted (Section 5.1).
- *Risk control for predictions* [16]. Comment: a general framework for controlling risk under stated sampling assumptions; applied here to entity-to-IP attribution with keys that are observable at scoring time (Section 4.6).

Position of this work.

BTC-FUSION is neither a new deanonymisation attack nor a new classifier. Illicit-activity classifiers return a score per transaction or address from ledger features alone; blockchain forensics groups addresses into entities but does not locate them on the network; and network-layer studies ask whether an observer can link transactions to IP addresses, usually reporting how often the links they make are right. BTC-FUSION takes passively collected announcement records as given and adds the decision layer that none of these provides: a policy that decides, entity by entity, whether any address may be named, with a stated bound on the error among the answers it gives, alongside a ranked and explained queue of chain-side leads. Its components are established techniques; the contribution lies in their combination, in calibrating the decision to answer, and in an evaluation that measures failures as carefully as successes.

3 Problem setting

Input.

Each record is one observed transaction announcement with 14 fields: the timestamp; the source and destination IP address and port; the transaction identifier (TXID); the input and output addresses and amounts; the fee; the script type; and, optionally, the country and autonomous system number (ASN). Records arrive as CSV, JSON Lines or XML.

Collector model.

The records are collected by an authorised operator who runs passive listening nodes and logs the announcements received from their peers. The fraction of the network’s peers that are monitored is the *observation coverage*. No active probing, traffic manipulation or control of Internet routing is assumed.

Output.

The output is a ranked queue of entity-level case files. Each case file carries a probability, the features that drove the score, typology evidence that cites concrete transactions, and either a named IP address with up to four alternatives or a stated reason for declining. The output is a lead for a human analyst. It is not evidence, and an attribution names an address, not a person: an address may be shared, reassigned or operated on someone else’s behalf.

Scope.

The detection scope was defined by the authors: eight actor archetypes, six laundering typologies (two held out of training), and entity-to-IP attribution. The system runs on one air-gapped workstation, and a build check fails if application code references a remote host.

4 System design

4.1 Overview

One orchestration function owns the nine-stage sequence of [Figure 1](#) and is shared by the command-line interface, the API service and the tests. Training and scoring share stages 1 to 5 exactly, so features cannot be computed differently at training and scoring time. The service, the models and a DuckDB database run in one process on one workstation. When a stage cannot run, the pipeline degrades instead of failing: with no model artefacts it falls back to unsupervised scoring, with no IP columns it runs in chain-only mode, and if attribution fails the queue is still produced with attribution marked as unavailable.

4.2 Ingest and entity resolution

Rows are mapped to a canonical schema, exact duplicates are removed, and invalid rows are quarantined with a reason rather than silently dropped. Source IP addresses

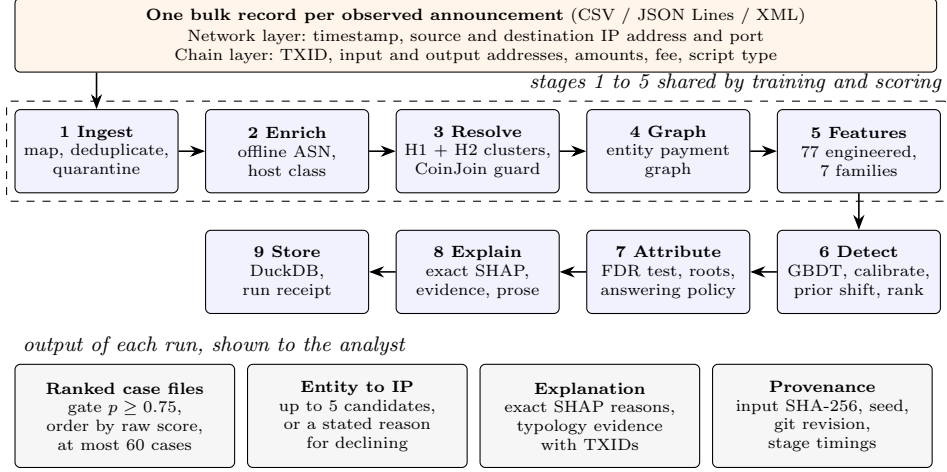


Fig. 1 Architecture of BTC-FUSION. Both layers arrive in the same record. The nine stages run in one Python process behind a local API service and a browser interface, with no network access. Stage 6 orders the queue by the classifier alone; stage 7 names an address only where the answering policy of [Section 4.6](#) allows it.

are enriched from a bundled offline copy of the DB-IP Lite database with the ASN, the country and a host class: residential, mobile, hosting, CDN, VPN or Tor. Announcements are collapsed to one row per transaction before any chain statistic is computed, so that repeated observations do not multiply chain features. Heuristic H1 (common-input ownership) links the input addresses of a transaction, and heuristic H2 (change) links a single qualifying output, one that first appears in the transaction and matches its script type, to the first input. Entities are the connected components of both edge sets.

CoinJoin guard.

H1 assumes that one party signed every input, which a CoinJoin is built to break. Transactions with a CoinJoin shape are therefore excluded from H1: following Stütz et al. [9], a block of at least three equal-value outputs that make up at least half of all outputs, with at least as many inputs as equal outputs. A single owner paying many equal amounts has the same shape, so the rule also requires evidence of several owners: two or more non-equal outputs, or one input worth at least the denomination for every equal output. On an audit capture the rule flags all 120 generated rounds with no false flags, whereas the version without the several-owners test flagged 340 transactions, 220 of which were custodial payout batches that H1 should have merged.

4.3 Features

Each active entity, one with at least one outgoing or two incoming payments, is described by 77 engineered features in seven families: network (19 features, such as IP, ASN and country diversity, the shares announced from Tor, VPN, hosting, residential and mobile addresses, port fingerprints and peers reached), amount and structure (18), temporal (13), graph topology (12), script type (5), counterparty (5) and observation

window (5). Every feature has a stated meaning, which is what makes the explanations checkable. An earlier structural embedding computed from random walks on the entity graph [18] was removed, because its axes are not fixed across captures: it helped on the capture it was fitted to but lowered AP on an unseen one (0.2028 against 0.2166 and 0.4879 against 0.4992), and a deployed model always scores an unseen capture.

4.4 Detection, calibration and ranking

The classifier is a gradient-boosted tree ensemble; all results use the histogram gradient boosting classifier of scikit-learn with at most 300 iterations, 63 leaves, a learning rate of 0.06, balanced class weights and early stopping on a 15% validation split. Isotonic regression [19] fitted on a separate calibration fold maps the raw score to a probability. Because the calibration fold and a scored capture need not share a prevalence, the probability shown to the analyst is moved to the capture’s own base rate, which is estimated from the scores alone by the expectation maximisation (EM) procedure of Saerens et al. [20]. Entities with a calibrated probability of at least 0.75 are flagged and ordered by the raw score, with value moved breaking exact ties. An Isolation Forest [21] and six rule-based typology matchers (peel chain, fan-out and fan-in, rapid layering, mixer signature, dormancy burst and cross-ASN structuring) attach novelty and evidence citing concrete transactions to each case, but they carry no weight in the ranking, for the reason measured in Section 6.3.

4.5 Attribution scoring

Attribution is run for a set E of entities, such as the alerted entities or a test fold. Let k_{ei} be the number of announcements of transactions sent by entity e that carry source IP address i , and let n_e , K_i and N be the row, column and grand totals. Every observed pair is scored with a hypergeometric tail probability p_{ei} and a positive pointwise mutual information magnitude q_{ei} :

$$p_{ei} = \Pr[X \geq k_{ei}], \quad X \sim \text{Hyp}(N, K_i, n_e), \quad (1)$$

$$q_{ei} = \max\left(0, \log \frac{k_{ei}N}{n_e K_i}\right). \quad (2)$$

Pairs are kept if they pass Benjamini-Hochberg control [22] at $\alpha = 0.01$. Each kept pair receives the confidence

$$c_{ei} = \min \left\{ 0.99, h_{ei}(1 - e^{-k_{ei}/4})(1 + 0.35 r_{ei}) \right. \\ \left. \times f_i(0.85 + 0.15 a_{ei}) \right\}. \quad (3)$$

The significance strength $h_{ei} = (\min\{1, -\log_{10} p_{ei}/12\} \cdot \min\{1, q_{ei}/4\})^{1/2}$ combines significance and effect size, because a busy relay can be highly significant yet weakly associated. The repeat weight $1 - e^{-k_{ei}/4}$ saturates, so a single sighting earns 22% of full weight. The root fraction r_{ei} uses the destination addresses of the announcements to

reconstruct each transaction’s propagation: an address that announces the transaction but is never announced to within it is a root candidate, the earliest candidate receives a weight of one over the number of candidates, and r_{ei} is the accumulated weight divided by k_{ei} , capped at 1. The infrastructure factor f_i multiplies a declared-class factor (residential 1.00, mobile 0.90, hosting 0.72, CDN 0.30, VPN 0.22, Tor 0.12, unknown 0.60) by an observed-sharing factor $\min\{1, \max\{0.05, 1/(1 + (m_i - 1)/8)\}\}$, where m_i is the number of distinct entities seen announcing from i . The time-zone agreement a_{ei} compares the UTC offset that best fits the entity’s hour-of-day activity with the offset of the address’s ASN; it is 1 within 90 minutes, falls linearly to 0 over the next six hours, and is 0.5 when the entity shows no clear daily rhythm. The five best candidates are returned.

4.6 Risk-controlled answering policy

Statistical significance alone is not a suitable criterion for naming an address. A relay really is associated with the entities whose transactions it forwards, so it also passes the test (Section 6.7). The decision to answer is therefore treated as a selective prediction problem [16].

Let $c_e = \max_i c_{ei}$ be the confidence of entity e ’s best candidate. A policy key j combines the declared class of that candidate with an actor profile observable at scoring time: *shared* when any announcement of the actor’s transactions came from a VPN, Tor or CDN exit, and *direct* otherwise. For the calibration cases D_j of key j and each threshold t on the grid $G = \{0, 0.01, 0.02, 0.03, 0.05, 0.08, 0.10, 0.15, 0.20, 0.30\}$, let $A_j(t) = \{e \in D_j : c_e \geq t\}$ be the answered set and $w_j(t)$ the number of wrong answers in it. Key j answers from the threshold

$$t_j = \min \{ t \in G : \text{UCB}(w_j(t), |A_j(t)|; \delta') \leq \varepsilon \}, \quad (4)$$

$$\delta' = \frac{\delta}{(|J| + 1) |G|}, \quad (5)$$

where J is the set of keys in the calibration fold, $\text{UCB}(w, n; \delta')$ is the one-sided Clopper-Pearson upper confidence bound [17] on an error rate after w errors in n answers, $\varepsilon = 0.05$ and $\delta = 0.05$; the extra term in the denominator covers a pooled threshold used for keys never seen in calibration. If the calibration cases of each key are exchangeable with the cases it will score, then, except with probability at most δ , every selected threshold keeps the true error rate among answers at or below ε , simultaneously for all keys. This is a Bonferroni instance of Learn then Test [16]. A key with fewer than 30 calibration cases, or with no grid point that meets the bound, never answers, and a VPN, Tor or CDN candidate is never named at all, because such an address belongs to a provider.

Sharing-aware key.

The key above cannot see whether the named address serves one actor or many, yet when an actor’s own address is never observed, a relay serving many entities is often its best remaining candidate. The sharing-aware variant therefore adds one observable attribute of the best candidate: *busy* when more than M distinct entities announce

from it, and *exclusive* otherwise. M is chosen from $\{4, 8, 16, 32, 64\}$ or no split, as the cut that yields the most correct answers on the second half of the calibration fold; a fixed cut of 16, specified before any result was seen, is also reported. The pinned artefacts use the class-and-profile key.

4.7 Explanation

For the cases shown, exact TreeSHAP values [14] are computed and the top eight features are turned into sentences. Where the engine declines to name an address, the case file gives the policy’s reason instead. Each run stores a receipt with row counts, per-stage timings, the SHA-256 of the input, the seed and the git revision.

5 Data and evaluation protocol

5.1 Synthetic data and its leakage audit

The generator draws actors from eight archetypes (exchange, merchant, retail, miner, mixer, darknet market, ransomware and extortion), each with its own activity rate, address count, diurnal rhythm and network infrastructure. The illicit label is carried by darknet, ransomware and extortion actors and by the mule wallets that their laundering campaigns create; victims who pay criminals, mixers and licit pass-through wallets are labelled licit. Campaigns follow one of six typologies: peel chain, fan-out and fan-in, rapid layering, mixer pass-through, dormancy burst and cross-ASN structuring. Criminal and licit payments are drawn from the same size distribution, fees depend only on transaction size and a daily congestion cycle, and licit pass-through chains of the same shape as laundering mules are generated deliberately, so that chain structure alone is not a shortcut. The simulated P2P network has 2,500 peers, of which a fixed 25% are monitored; each originator announces a transaction to eight peers and relaying continues for two further hops. In this simulator the originator’s own announcements make up half of the traffic generated for a transaction, whereas on the real network essentially every reachable node relays it, so attribution results here are optimistic by construction.

A generator can encode its labels in fields that a detector then exploits [15]. Before any result is accepted, the classifier is trained on a *strict* tier of four fields whose construction gives them no path to the label (the mean fee rate, two operating-system shares derived from ephemeral ports, and the share of round amounts), and the capture is rejected if the resulting average precision exceeds 1.30 times the prevalence. During development the audit failed three times, and each failure exposed a generator defect that was fixed before any reported result was produced: criminal proceeds drawn from fixed absolute amounts (a 1.7-fold shortcut), short-lived pass-through wallets that existed only as criminal mules (about 2.4-fold), and mule wallets drawn from a different operating-system distribution (about 1.2-fold), which the model had read through the ephemeral-port fingerprint. Table 1 reports the audit for the two pinned captures: the strict tier scores 1.166 and 1.240 times the prevalence, against the gate of 1.30, while the same classifier on all 77 features reaches 10.786 and 13.733 times. The informative tier of amount aggregates is reported but not gated, because amount

Table 1 Leakage audit. Lift is average precision divided by the test prevalence of the audit split (0.0827 on demo, 0.0653 on bulk). Only the strict tier, jointly, is gated, at 1.30.

Field set	Tier	demo (49,310 / 21,134)		bulk (207,247 / 88,821)	
		AP	Lift	AP	Lift
mean_fee_rate_sat_vb	strict	0.0945	1.143	0.0789	1.209
ephemeral_linux_frac	strict	0.0952	1.151	0.0731	1.120
ephemeral_windows_frac	strict	0.0923	1.116	0.0739	1.132
round_number_frac	strict	0.0827	1.000	0.0653	1.000
mean_fee_sat	informative	0.1019	1.232	0.0782	1.198
mean_value_out	informative	0.1137	1.375	0.0900	1.380
std_value_out	informative	0.0931	1.126	0.0727	1.115
total_out	informative	0.1127	1.362	0.0850	1.302
total_in	informative	0.1054	1.275	0.0835	1.279
Strict tier, jointly (gated)	strict	0.0965	1.166	0.0809	1.240
Informative tier, jointly	informative	0.1841	2.226	0.1450	2.222
Control: all 77 engineered features		0.8921	10.786	0.8961	13.733

structure legitimately carries signal. The audit tests only fields that the authors chose to call uninformative, so passing it is necessary but not sufficient.

5.2 Captures and replication

Two profiles are generated. The demo profile (4,200 actors, 564,838 announcement records) has a test prevalence of 3.25%; the bulk profile (16,000 actors, 2,470,860 records) keeps the number of laundering campaigns almost fixed, which lowers the test prevalence to a realistic 0.39%, and is treated as the primary result. The pinned captures use seed 20260826; [Table 2](#) lists them and the populations derived from them.

To measure variation between datasets rather than within one, four further captures of each profile were generated with an unchanged configuration (seeds 20260901, 20260902, 20260903, 20260904 for demo and 20260901, 20260902, 20260903, 20260905 for bulk). A capture is used only if it passes the leakage audit. The bulk capture generated from seed 20260904 failed the audit (strict-tier lift 1.316) and was replaced by the next seed that passed; its results are not used anywhere in this paper. The strict tier of the ten captures used scored between 1.166 and 1.271 times the prevalence, while all 77 features reached between 10.8 and 15.4 times. Regenerating the demo profile from its seed reproduced the pinned capture byte for byte.

5.3 Splits, metrics and selection

Entities are ordered by their first activity and cut 60/20/20 into training, calibration and test folds, after the entities of two held-out typologies (dormancy burst and cross-ASN structuring) are removed; guards reject any split that places an entity in two folds, starts the test fold before training ends, or leaks a held-out typology into training. Features are computed over the whole capture without labels, so the setting is transductive. Each resolved entity takes the label of the true actor that owns most of its addresses.

Table 2 Synthetic captures and the populations derived from them (model artefacts, manifests and generator configuration).

	demo	bulk (primary)
Generated actors / days / campaigns / licit pass-through chains	4,200 / 45 / 55 / 320	16,000 / 60 / 70 / 1,400
Announcement records	564,838	2,470,860
Capture size (bytes); SHA-256 prefix	242,225,989; 458a72b04177	1,021,960,616; 1064dbf99ebd
Addresses $\rightarrow$ resolved entities	437,566 $\rightarrow$ 199,060	1,816,507 $\rightarrow$ 836,843
H1 co-spend edges / H2 change edges	160,106 / 78,481	635,144 / 344,779
CoinJoin transactions left unmerged	120	453
Singleton entities (share of resolved)	165,439 (83.1%)	695,892 (83.2%)
Largest entity (addresses)	674	957
Mean cluster purity; clusters with purity below 0.6	0.9958; 994	0.9957; 4,248
Active (scored) entities (share of resolved)	70,444 (35.4%)	296,068 (35.4%)
Train fold: entities / illicit / prevalence	42,223 / 3,730 / 8.83%	177,568 / 15,720 / 8.85%
Calibration fold: entities / illicit / prevalence	14,074 / 1,602 / 11.38%	59,189 / 3,472 / 5.87%
Test fold: entities / illicit / prevalence	14,075 / 457 / 3.25%	59,191 / 233 / 0.39%
Held-out-typology entities (all illicit)	72	120

Ranking is measured by average precision (AP), whose chance level equals the prevalence [23], together with precision among the top 50 entities and AUROC; calibration by the expected calibration error (ECE) [24]. Attribution is measured by top-1 accuracy among answered entities, compared with the accuracy of guessing uniformly among the returned candidates, and by the error rate among answers. Each capture passes through the whole procedure (audit, features, splits, training, calibration, policy fitting and evaluation) with the model seed fixed; we report the mean, the sample standard deviation and a 95% t -interval over the five captures of each profile, and retrain the pinned models with five seeds to separate model variance from data variance. Every tuned design choice is made on the calibration fold alone: its first half, in time order, fits the calibrator or policy, and its second half scores the options. The alert threshold of 0.75 and the policy target were fixed in advance.

5.4 Real data

On Elliptic [6] (203,769 transactions, 9.76% of the labelled ones illicit), the classifier is trained on the labelled transactions of time steps 1 to 34 and tested on steps 35 to 49 at a threshold of 0.5, with five model seeds. Elliptic links transactions only within a time step, so this split leaves no graph edge between training and test, which we verified by counting. On Elliptic++ [7] (822,942 wallet addresses), our harness trains on steps 1 to 33 and tests on steps 34 to 49, keeping one row per address and dropping test addresses already seen in training. The published Elliptic++ actor-classification result is also reproduced from the released notebook and re-evaluated under the temporal split that the paper describes.

6 Results on synthetic captures

All figures in this section come from synthetic data. Pinned-capture figures are read from the model artefacts trained on 19 September 2026; replicate figures come from runs of 29 September 2026. The bulk profile is the primary result.

6.1 Detection and its variation across captures

Table 4 gives the main measurements. On the pinned bulk capture, the ranking reaches an AP of 0.2527 at a prevalence of 0.39%, 64 times the chance level, and 46 of the 50 highest-ranked entities are illicit. For an analyst this is the figure that matters: the top of the queue is almost entirely illicit although fewer than one entity in 250 is. Accuracy, by contrast, is uninformative at this base rate: the model’s accuracy (0.9947) is lower than that of predicting every entity licit (0.9961), which is why the evaluation uses AP. Table 3 gives the full detail for the two pinned captures. The interval for precision among the top 10 is wide ([0.72, 1.00]), so the top 50 is the more informative figure. At the operating threshold the picture is less favourable: only 0.2976 of the flagged bulk entities are illicit and 0.2618 of the illicit entities are flagged, because the threshold is applied to a probability calibrated at a higher prevalence (Section 6.5).

The replicates show how much of this depends on the particular dataset. On bulk, AP ranges from 0.1024 to 0.2760 across the five captures (mean 0.1859), whereas retraining the pinned model with five seeds gives only 0.2579 ± 0.0115 . The variation therefore comes from the data rather than from the model: with a few hundred test positives, which actors the laundering campaigns happen to recruit changes the result, and the test prevalence itself ranges from 0.27% to 0.54%. The pinned capture has the second highest AP of the five, so it should not be read as the expected value.

6.2 Where the signal comes from

Table 5 retrains the classifier on the 19 network-family features alone, on the 58 chain-side features alone, and on all 77. On demo the fused set is the best of the three on all five captures. At the realistic prevalence of the bulk profile the picture is not stable: the fused set has the highest mean AP (0.1859, against 0.0921 for chain and 0.0920 for network features), but it is the best of the three on only three of the five captures, and the chain-only model is the best on two. Chain-only AP ranges from 0.0066 on

Table 3 Entity-level detection on the test fold. “Deployed ordering” is the raw score with a value tie-break. Threshold metrics use the calibrated probability $p \geq 0.75$. Lift is AP divided by the prevalence.

	demo	bulk (primary)
Test entities / illicit / prevalence	14,075 / 457 / 3.25%	59,191 / 233 / 0.39%
AP, deployed ordering (lift)	0.4725 (15 $\times$)	0.2527 (64 $\times$)
AP, calibrated probability (lift)	0.4596 (14 $\times$)	0.2475 (63 $\times$)
AUROC, calibrated probability	0.8108	0.6715
P@10, deployed ordering [Wilson 95%]	1.00 [0.72, 1.00]	1.00 [0.72, 1.00]
P@25, deployed ordering [Wilson 95%]	1.00 [0.87, 1.00]	0.96 [0.80, 0.99]
P@50, deployed ordering [Wilson 95%]	1.00 [0.93, 1.00]	0.92 [0.81, 0.97]
Recall at precision ≥ 0.80	0.3698	0.2318
Flagged at $p \geq 0.75$ (TP / FP / FN)	187 (162 / 25 / 295)	205 (61 / 144 / 172)
Precision / recall at threshold	0.8663 / 0.3545	0.2976 / 0.2618
F1 / MCC at threshold	0.5031 / 0.5459	0.2785 / 0.2764
Accuracy vs. all-negative baseline	0.9773 vs. 0.9675	0.9947 vs. 0.9961
ECE: raw score, isotonic, after prior shift	0.1639, 0.0651, 0.0250	0.2528, 0.0506, 0.0323
Held-out typologies: recall / AP	0.2083 / 0.1380	0.1250 / 0.0560

the pinned capture to 0.1607, so the pinned capture is unusually favourable to the fused model. Two conclusions follow. The two layers are complementary on average, but which one carries the signal depends on the capture, so a single-capture ablation would have been misleading. And part of the network signal is built into the generator, which gives illicit archetypes more Tor, VPN and hosting infrastructure; this signal lies in choices that an adversary controls, which [Section 8](#) turns into an attack.

6.3 Design choices made on validation data

Three choices were re-made on validation data on every one of the ten captures, and in every case the choice agrees with the deployed design. First, the ranking uses the classifier alone. On their own, the rule-based typology matchers rank below chance (0.76 times the prevalence on bulk) and the Isolation Forest barely above it (1.78 times), and blending them into the score, as an earlier design did with weights of 0.85, 0.05 and 0.10, lowers the validation AP on the pinned bulk capture from 0.7177 to 0.4716; on the test fold it would have cut AP from 0.2475 to 0.0194 ([Table 6](#)). The evidence score is coarse, so even a small weight promotes a broad band of unremarkable entities. Second, the queue is ordered by the raw score rather than by the calibrated probability, because isotonic calibration maps many distinct scores to one value and loses the order within them. Third, the five observation-window features are kept (validation AP 0.7190 with them against 0.6941 without on bulk). Because the test fold played no part in these choices, the test results in [Table 4](#) are not biased by them.

6.4 Unseen typologies and short histories

Generalisation to laundering patterns absent from training is the clearest weakness of the ranking. On the pinned bulk capture the alert threshold recalls 15 of the 120 entities

Table 4 Main results on the test fold. “Pinned” is the capture generated from seed 20260826; the other columns give the mean $\pm$ sample standard deviation over five independently generated captures per profile and, for the primary bulk profile, a 95% t -interval for the mean (truncated to $[0, 1]$ for proportions). AP and precision use the deployed ordering; held-out recall is at $p \geq 0.75$.

Measurement	demo		bulk (primary)		
	Pinned	Mean $\pm$ s.d.	Pinned	Mean $\pm$ s.d.	95% CI
Test prevalence	3.25%	2.70% $\pm$ 0.35%	0.39%	0.38% $\pm$ 0.10%	[0.25%, 0.51%]
Average precision (AP)	0.4725	0.3617 $\pm$ 0.0960	0.2527	0.1859 $\pm$ 0.0760	[0.0915, 0.2803]
AP over chance (lift)	14.6	13.3 $\pm$ 2.7	64.2	51.9 $\pm$ 21.8	[24.9, 79.0]
Precision in the top 50	1.00	0.90 $\pm$ 0.15	0.92	0.62 $\pm$ 0.27	[0.28, 0.96]
AUROC	0.8108	0.7627 $\pm$ 0.0540	0.6715	0.7467 $\pm$ 0.0540	[0.6796, 0.8138]
ECE after prior shift	0.0250	0.0386 $\pm$ 0.0301	0.0323	0.0635 $\pm$ 0.0306	[0.0255, 0.1016]
Held-out typologies: recall	0.2083	0.1149 $\pm$ 0.0801	0.1250	0.1318 $\pm$ 0.0277	[0.0975, 0.1662]
Attribution: share answered	0.9140	0.9193 $\pm$ 0.0033	0.9285	0.9290 $\pm$ 0.0029	[0.9254, 0.9326]
Attribution: top-1 when answering	0.9747	0.9771 $\pm$ 0.0016	0.9764	0.9757 $\pm$ 0.0008	[0.9747, 0.9766]
Attribution: error among answers	0.0253	0.0229 $\pm$ 0.0016	0.0236	0.0243 $\pm$ 0.0008	[0.0234, 0.0253]
Wrong answers for hosting/CDN actors	217	185 $\pm$ 22	876	897 $\pm$ 37	[851, 942]

from the two held-out typologies (0.1250), against 0.2618 for in-distribution entities; across the five captures held-out recall is 0.1318 ± 0.0277 . Holding out each typology in turn on the pinned bulk capture, recall is highest for peel chain (0.2143) and zero for one typology. The unseen entities are usually ranked above the negatives, so the model sees something unusual, but rarely high enough to cross the alert threshold. Weighting the novelty score into the ranking does not solve this: a weight of 0.05 raised held-out recall only from 0.1250 to 0.1500 while cutting AP from 0.2475 to 0.1357. A second weakness concerns entities that the capture saw only briefly. Ranking quality rises with the time an entity was observable, and on bulk the quarter of test entities observed for the shortest time is ranked below chance (AUROC 0.366), consistent with features computed from truncated histories. A low score for such an entity should be read as unranked rather than cleared.

Table 5 Feature-family ablation: the same classifier trained on network features only, chain features only, or both. Mean $\pm$ standard deviation over five captures per profile (test fold).

Profile	Features	AP	P@50	Held-out recall
demo	Fused (all 77)	0.3617 $\pm$ 0.0960	0.90 $\pm$ 0.15	0.1149 $\pm$ 0.0801
	Chain only (58)	0.0696 $\pm$ 0.0435	0.16 $\pm$ 0.18	0.0057 $\pm$ 0.0078
	Network only (19)	0.2870 $\pm$ 0.0741	0.72 $\pm$ 0.16	0.2202 $\pm$ 0.0943
bulk	Fused (all 77)	0.1859 $\pm$ 0.0760	0.62 $\pm$ 0.27	0.1318 $\pm$ 0.0277
	Chain only (58)	0.0921 $\pm$ 0.0669	0.37 $\pm$ 0.28	0.0504 $\pm$ 0.0578
	Network only (19)	0.0920 $\pm$ 0.0403	0.31 $\pm$ 0.11	0.0625 $\pm$ 0.0175

Table 6 Fusion-weight sweep on the pinned captures. Validation AP scores the calibrated probability on the second half of the calibration fold, with the calibrator fitted on the first half; the choice is made on this column. Test-fold values are shown for transparency. The held-out columns score held-out-typology positives mixed with test negatives. The first row is the deployed configuration.

Weights			demo			bulk					
Sup.	Nov.	Evid.	Valid.	AP	Test AP	Valid.	AP	Test AP	Test P@50	Held-out AP	Held-out recall
1.00	0.00	0.00	0.6744	0.4596	0.7177	0.2475	0.92		0.0560	0.1250	
0.95	0.05	0.00	0.6640	0.4234	0.7143	0.1357	0.52		0.0310	0.1500	
0.90	0.10	0.00	0.6532	0.3956	0.6858	0.0644	0.14		0.0182	0.1583	
0.80	0.20	0.00	0.6450	0.3377	0.6644	0.0369	0.12		0.0253	0.1667	
0.95	0.00	0.05	0.6549	0.4164	0.5972	0.0431	0.20		0.0051	0.0583	
0.85	0.05	0.10	0.6268	0.3339	0.4716	0.0194	0.20		0.0035	0.0083	
0.65	0.25	0.10	0.5936	0.2206	0.4433	0.0110	0.04		0.0099	0.0917	

6.5 Calibration at deployment prevalence

The calibrator is fitted on a fold with 5.87% prevalence and applied to one with 0.39%, so the displayed probabilities are too high: in the bin with mean predicted probability 0.8436 only 3.7% of bulk test entities were illicit. Moving the probabilities to the capture’s own base rate with EM lowers the ECE from 0.0506 to 0.0323, but EM estimates that base rate at 0.0362 against a true 0.0039, because the score distribution it fits was produced at a much higher prevalence. With the true base rate the ECE would be 0.0020, so almost all of the remaining error lies in the prevalence estimate. Across the five bulk captures the ECE after correction is 0.0635 ± 0.0306 . The displayed probability is therefore best read as an ordinal indicator, not a frequency.

6.6 Explanations

An explanation is useful only if it reflects what drives the score and does not change arbitrarily. Both properties were measured on the deployed bulk model. Replacing each explained entity’s top SHAP features with the training median lowers its score far more than replacing random features (area over the perturbation curve 0.3199 against 0.0120), and retraining without the ten globally most important features lowers AP from 0.2532 to 0.0131, against 0.2418 when ten random features are removed. The

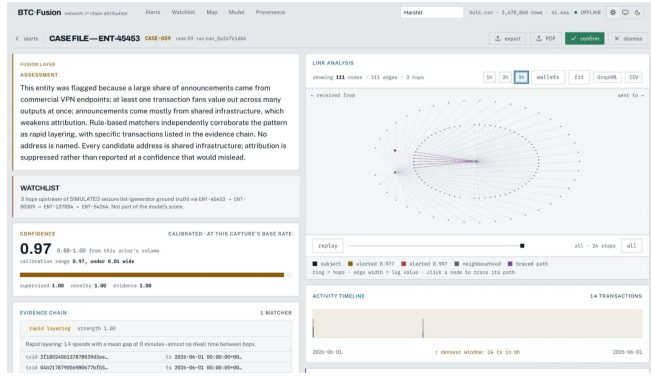


Fig. 2 A case file from the same run: the generated assessment, the watchlist proximity (shown but not part of the score), the calibrated confidence, the transactions that corroborate a rapid-layering pattern, the entity’s payment neighbourhood and its activity timeline.

explanations are therefore faithful to the model. They are also reasonably stable: the top eight features of five models retrained with different seeds share about 6.9 of eight features (Jaccard index 0.7507, against 0.0582 for random sets), and 5% input noise leaves a Jaccard index of 0.8086. They do not, however, name the planted laundering mechanism more often than chance would with a random typology: a mechanism feature appears among the top three in 32.0% of cases, against 38.8% when typologies are permuted across entities, because a few generic structural features, above all the maximum number of outputs, dominate every explanation. An explanation tells the analyst why the model ranked a lead high, not which pattern the offender used. [Figure 2](#) shows how an explanation reaches the analyst: the case file states in words why the entity was flagged, lists the transactions that corroborate a typology, gives the calibrated confidence with its range, and explains why no address is named.

6.7 Attribution

Link-level significance does not bound wrong answers.

Nearly every tested entity-IP pair passes false discovery rate control (97.3% on demo and 96.5% on bulk). On the demo capture, 78.5% of the pairs accepted by the Benjamini-Hochberg procedure are not the actor’s own address, and the stricter procedures of Benjamini and Yekutieli [25] and of Wang and Ramdas [26] still leave 45.1% and 23.7%. The reason is that the null hypothesis of the test, that an address and an entity are unrelated, is false for a relay, which really does carry the entity’s transactions. The guarantee must therefore apply to the decision to answer, which is what the policy of [Section 4.6](#) does ([Table 7](#)).

The policy keeps its bound where it applies.

[Table 8](#) reports the outcomes on the pinned test folds. On the pinned bulk capture the policy answers 92.9% of gradable entities, and its top-1 accuracy of 0.9764 is three times the 0.3136 expected from guessing among the same candidates. The error among answers, 2.36%, is below the 5% target, and it stays below 2.6% on all ten captures

Table 7 False attributions among accepted entity-IP pairs on the demo capture at $\alpha = 0.01$ (research FDR study). Power is the share of the 61,396 true pairs recovered.

Procedure	Pairs accepted	Power	False-attribution share
Benjamini-Hochberg [22]	285,981	0.9998	0.7854
Benjamini-Yekutieli [25]	96,497	0.8637	0.4505
e-BH, $\kappa = 0.5$ [26]	63,429	0.7885	0.2368

(Table 4). Top-3 accuracy exceeds top-1 accuracy by less than 0.001, so when the first candidate is wrong the true address is usually absent from the list rather than ranked second. Most abstentions are correct: of the 4,230 entities the policy declined on bulk, 3,735 had no observable address of their own, so no answer could have been right. Keying the policy is what makes this trade-off acceptable. On the same bulk scores, the single hand-set cutoff of 0.15 used in earlier versions answers very accurately (0.9905) but withholds 24,502 entities whose own address was observed, and one pooled threshold fitted on the calibration fold answers slightly more entities but gives 1,837 wrong answers instead of 1,295. The fitted keys that answer at all are those in which a direct actor’s best candidate is a residential or mobile address; every key whose candidate is a hosting address, or whose actor was seen through an exit, never answers.

The failure is concentrated on hosting infrastructure.

Table 9 breaks the outcomes down by the class of the actor’s own address. Residential and mobile actors are answered correctly in more than 99% of cases. Actors whose own address belongs to a hosting provider or a CDN are answered 876 times on the pinned bulk capture and are never right, and such actors account for 66% to 69% of all wrong answers on every bulk capture. They announce directly, so they present as direct actors, but their own address is usually not observed and the candidate the engine names is a relay belonging to someone else. The policy cannot see the class of the actor’s own address, only that of the candidate, so its bound, which is an average over each key, does not protect this subgroup. With a fixed cut of 16, the sharing-aware key removes 23% to 27% of all wrong answers on the bulk captures while lowering the answered share by at most 0.8 points, because it stops naming busy relays; it does not repair the hosting failure (876 wrong answers become 861 on the pinned capture). When the cut is chosen on validation data instead, the selection rule favours the most correct answers and keeps a policy identical to the deployed one on 5 of the five bulk captures.

The same behaviour is visible at the top of the live queue (Figure 3). In a scoring run of the bulk capture, the highest-ranked leads carry the status *suppressed*: the engine has ranked them as likely illicit but declines to name an address, because their candidates are shared infrastructure or belong to keys that never answer.

6.8 Runtime

On a laptop (Apple M4, 16 GB of memory, no GPU), scoring the bulk capture end to end took 42.58 s with a peak memory use of 4.75 GB. Figure 4 shows the analyst

Table 8 Attribution on the test fold at observation coverage 0.25. Answered entities are those for which the engine named an address; every other status, including a suppressed entity that still carries ranked candidates, is an abstention. Chance is the expected accuracy of uniform guessing among the returned candidates.

	demo	bulk
Entity-IP pairs tested $\rightarrow$ significant at FDR 0.01	66,414 $\rightarrow$ 64,647 (97.3%)	290,798 $\rightarrow$ 280,750 (96.5%)
Distinct IP addresses / pairs with root evidence	7,083 / 19,015	17,162 / 80,683
Gradable entities / answered (rate)	14,075 / 12,865 (0.9140)	59,191 / 54,961 (0.9285)
Abstentions, of which suppressed by the policy	1,210 / 1,024	4,230 / 3,604
Top-1 accuracy over answered entities	0.9747	0.9764
Error among answered entities	2.53%	2.36%
Top-1 over all gradable entities (abstention counted as a miss)	0.8909	0.9067
Correct / wrong / correctly declined / withheld	12,539 / 326 / 1,175 / 35	53,666 / 1,295 / 3,735 / 495
Top-3 accuracy / mean reciprocal rank	0.9755 / 0.9751	0.9768 / 0.9766
Mean candidates / single-candidate share	3.68 / 0.0508	3.73 / 0.0486
Chance among the same candidates (top-1 over chance)	0.3181 (3.06 $\times$)	0.3136 (3.11 $\times$)
AURC / excess AURC / oracle AURC	0.0084 / 0.0081 / 0.0003	0.0090 / 0.0087 / 0.0003

interface during such a run: each stage reports its own time, feature extraction is the slowest, and no network connection is used.

Every figure except wall-clock time is deterministic given the seed, and 214 tests pass at the revision used for the replication runs.

7 Results on real Bitcoin data

Everything in this section uses the public Elliptic and Elliptic++ datasets and no synthetic data. They carry no network layer, so only the chain-side classifier and the clustering heuristic can be tested on them.

Elliptic.

Over five model seeds, the classifier reaches an illicit-class F1 of 0.7585 ± 0.0097 and an AP of 0.8014 ± 0.0038 on the last 15 time steps. The published random forest of Weber et al. [6] reports an F1 of 0.788 with higher precision and lower recall, under a different feature protocol. Removing the time-step column barely changes the result (F1 0.7562),

Table 9 Attribution by the declared class of the actor’s own address. “Observable” counts entities at least one of whose own addresses appears in the capture.

Own class	bulk				demo			
	Gradable	Answered	Observable	Top-1	Gradable	Answered	Observable	Top-1
residential	28,471	27,209	27,052	0.9919	6,655	6,307	6,286	0.9914
mobile	27,982	26,868	26,686	0.9929	6,484	6,194	6,150	0.9916
hosting	1,685	686	446	0.0000	505	319	149	0.4514
CDN	361	190	0	0.0000	76	42	0	0.0000
VPN	665	8	0	0.0000	262	3	0	0.0000
Tor	27	0	0	n/a	93	0	0	n/a

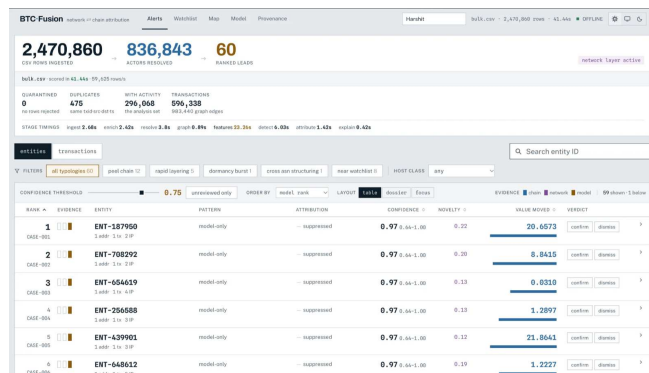


Fig. 3 The alert queue after scoring the bulk capture (screenshot of a run on 29 September 2026). Each lead shows its evidence layers, pattern, attribution status, calibrated confidence with its range, novelty and value moved. The top leads are ranked as likely illicit, but their attribution is suppressed rather than guessed.

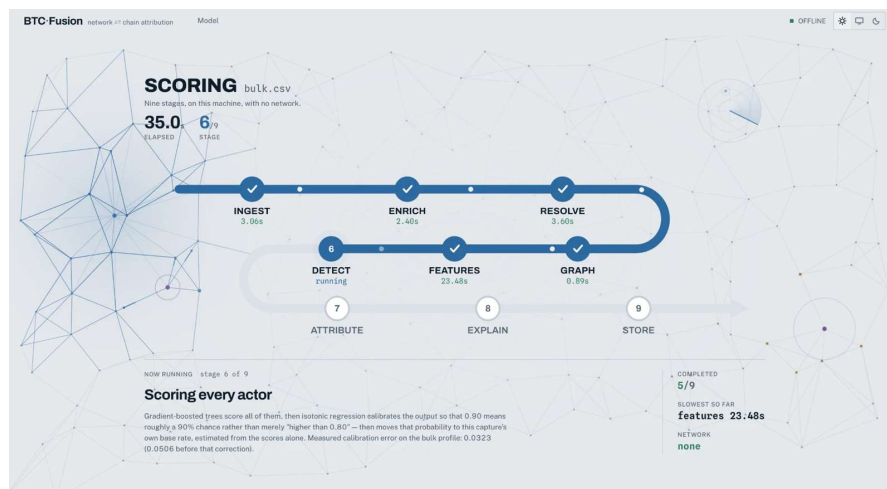


Fig. 4 The analyst interface while scoring the 2,470,860-record bulk capture on a laptop. Stages 1–5 are complete, the classifier stage is running, and feature extraction is the slowest stage.

Table 10 Chain-side classifiers on the Elliptic dataset, illicit class, test time steps 35 to 49. Rows of the standard harness give the mean $\pm$ standard deviation over five model seeds. Rows of the strict inductive protocol are single runs, except GraphSAGE, whose standard deviations over five seeds are 0.0509 (precision), 0.0112 (recall), 0.0295 (F1) and 0.0162 (AP). Published figures come from a different model and feature protocol and are context rather than a controlled comparison.

Protocol and model	Precision	Recall	F1	AP	MCC
<i>Standard harness: labelled transactions of steps 1 to 34 for training, 16,670 test transactions</i>					
Proposed GBDT, all 166 features	0.7837 $\pm$ 0.0198	0.7350 $\pm$ 0.0013	0.7585 $\pm$ 0.0097	0.8014 $\pm$ 0.0038	0.7427 $\pm$ 0.0109
Proposed GBDT, time-step column removed	0.7771 $\pm$ 0.0116	0.7365 $\pm$ 0.0024	0.7562 $\pm$ 0.0064	0.8017 $\pm$ 0.0031	0.7401 $\pm$ 0.0071
Proposed GBDT, 94 local features only	0.7135 $\pm$ 0.0202	0.7404 $\pm$ 0.0025	0.7265 $\pm$ 0.0098	0.7882 $\pm$ 0.0028	0.7073 $\pm$ 0.0106
Weber et al. [6], random forest, all features (published)	0.956	0.670	0.788	n/a	n/a
<i>Strict inductive: training subgraph of steps 1 to 34 only, time step dropped, test prevalence 6.5%</i>					
Random forest, 50 trees	0.9001	0.7239	0.8025	0.7845	
Proposed GBDT	0.7535	0.7368	0.7451	0.7954	
Two-hop propagated features + GBDT	0.7682	0.7313	0.7493	0.7958	
SGC, 2 hops	0.1704	0.7452	0.2773	0.2792	
GraphSAGE, mean over 5 seeds	0.5187	0.6081	0.5585	0.5066	

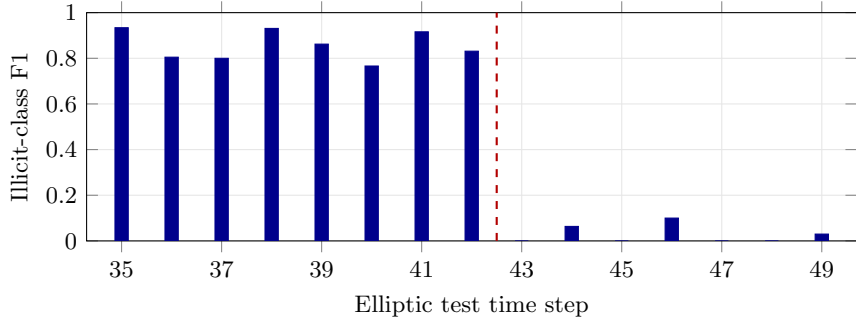


Fig. 5 Illicit-class F1 of the classifier (reference seed) on each Elliptic test time step. The dashed line separates steps 35 to 42 from steps 43 to 49. Illicit transactions per step, from step 35 to 49: 182, 33, 40, 111, 81, 112, 116, 239, 24, 24, 5, 2, 22, 36, 56.

so the score does not rest on the time index. Under a strict inductive protocol, in which the training graph contains steps 1 to 34 only, the tree models lead (random forest F1 0.8025, gradient boosting AP 0.7954) and the graph models trail (GraphSAGE F1 0.5585 over five seeds, SGC 0.2773), in line with [13]. Table 10 gives the full comparison.

The aggregate F1 hides a sharp change over time (Figure 5). Up to time step 42 the F1 on every step is at least 0.7662; from step 43 onwards it is at most 0.1000,

Table 11 Elliptic++ actor classification under different protocols (research reproduction ladder). “P2SH in test” is the share of pay-to-script-hash addresses among test addresses.

Protocol and model	Test rows	P2SH in test	Precision	Recall	F1	AP
As published [7], random forest	n/a	n/a	0.911	0.789	0.845	n/a
Published notebook: positional 70/30 cut, random forest (50 trees)	79,607	0.46	0.917	0.789	0.848	0.903
Same cut, proposed GBDT	79,607	0.46	0.650	0.928	0.764	0.926
Temporal split as stated (steps 1 to 34 / 35 to 49), random forest	92,451	0.14	0.333	0.210	0.257	0.231
Same temporal split, proposed GBDT	92,451	0.14	0.249	0.621	0.356	0.408
Our harness: address-disjoint, steps 1 to 33 / 34 to 49, GBDT	96,023	n/a	0.371	0.414	0.392	0.375

and the classifier recalls 4 of the 169 illicit transactions in those seven steps. Weber et al. [6] report the same collapse after time step 43 and attribute it to the shutdown of a dark market. A model trained on earlier illicit behaviour does not recognise the behaviour that replaces it; this is the real-data counterpart of the unseen-typology result of Section 6.4, and it is why a deployed model has to be retrained as criminal practice changes.

Elliptic++.

Elmougy and Liu report an actor-level F1 of 0.845 for a random forest under a temporal split into steps 1 to 34 and 35 to 49 [7]. Their released notebook, however, drops the time step and trains on the first 70% of rows in file order, and the file is sorted by address string. Running the notebook reproduces an F1 of 0.8484, but because address strings sort by encoding, 46% of its test addresses are pay-to-script-hash addresses, against 14% under the stated split. Under the temporal split as stated, the same random forest reaches an F1 of 0.2572 and our classifier 0.3557. The published figure therefore does not measure performance on later time periods. Table 11 shows each step of the reproduction.

Under our address-disjoint harness, the classifier reaches an AP of 0.3786 ± 0.0182 over five seeds at a prevalence of 5.56%, about 6.8 times chance. Actor classification on real data under temporal splits is weak for every model tested, including the published one.

Clustering on real data.

Running heuristic H1 on 202,804 Elliptic++ transactions, labels agree within clusters in 0.9979 of cases against 0.8209 when labels are shuffled within the same clusters. The largest real cluster, however, contains 14,885 addresses, against 674 in the demo

capture at a similar address count; this super-cluster effect, described by Harrigan and Fretter [8], is absent from our generator, so label purity on real data would be lower than on synthetic data.

8 Security and evasion analysis

BTC-FUSION is a forensic tool that offenders have reason to defeat, and an attribution tool that can harm a bystander if it is deceived. The adversary we consider controls one or more wallets and wants to avoid being ranked as illicit, to avoid being attributed to its own address, or to have another address named instead. It can choose its network path (residential, mobile, hosting, VPN, Tor or I2P), change addresses, join CoinJoin rounds, broadcast through a third party, and run P2P nodes that re-announce other users' transactions. It controls neither the collector nor the labels, and it does not know which peers are monitored.

Each measured attack modifies the pinned demo capture and reruns the whole pipeline with the model, calibrator and answering policies frozen. Attacks on attribution target 400 test entities drawn at random from the 12,539 that the deployed engine attributed correctly, so any change in outcome is caused by the attack. [Table 12](#) summarises the results.

Relay manipulation.

A node operator who receives a victim's transaction early can announce it before the victim's own announcements reach the collector. We model the strongest form of this attack: a fresh residential address, in the target's time zone wherever the address plan allows, announces each of the target's transactions one second before the first genuine announcement, to one, two or four of the monitored peers that received it. The engine then names the attacker's address for 74.0%, 96.8% and all of the targets. This is the most serious weakness found: the attribution score rests on first announcement and co-occurrence, and an active adversary can forge both. The sharing-aware key does not help, because each forged address serves a single target. If the forging addresses belonged to a residential proxy service, the named address would belong to an unwitting third party. Cross-checking several independent collectors, which a forged announcement would have to reach first, is a natural countermeasure but was not evaluated; until such checks exist, an attribution must be treated as a lead that an adversary may have planted.

Private and custodial broadcasting.

Private broadcast in Bitcoin Core sends transactions submitted through `sendrawtransaction` only over Tor or I2P, custodial wallets broadcast from the custodian's infrastructure, and light clients hand transactions to a server; in each case the actor's own address never announces the transaction. Removing every announcement made from the targets' own addresses leaves no correct answer. The deployed key then names another address, in practice a relay, for 16.8% of the targets, whereas the sharing-aware key names no address at all and declines. This is the case the refined key was designed for, and as private broadcast spreads it will become the common one.

Table 12 Measured attacks and shifts against the frozen system (pinned demo capture unless stated). Attribution attacks use 400 entities that were attributed correctly before the attack. The sharing-aware key uses the cut chosen on validation data.

Attack or shift	Outcome measured	Deployed key	Sharing-aware key
Relay injection, 1 forged announcement per transaction	attacker’s address named	74.0%	74.0%
Relay injection, 2 forged announcements per transaction	attacker’s address named	96.8%	96.8%
Relay injection, 4 forged announcements per transaction	attacker’s address named	100.0%	100.0%
Own announcements withheld (private or custodial broadcast)	another address named	16.8%	0.0%
	declined	75.0%	91.8%
Address rotation, new address per day	actor’s own address named	100.0%	100.0%
Address rotation, new address per transaction	actor’s own address named	100.0%	100.0%
No attack (coverage 0.25, as calibrated)	error among answers	2.5%	1.9%
Loss of coverage to 0.15	error among answers	4.4%	10.3%
Loss of coverage to 0.10	error among answers	14.0%	22.6%
Loss of coverage to 0.05	error among answers	34.4%	35.1%
Network mimicry by illicit actors (demo)	detection AP (before 0.4725)	0.0810	n/a
Network mimicry by illicit actors (bulk)	detection AP (before 0.2527)	0.0279	n/a

Address rotation.

Moving every announcement from a target’s own addresses to a fresh address in the same network block, once per day or once per transaction, did not change the outcome: the engine named one of the actor’s own addresses for every target, because each fresh address still carries the actor’s first announcements and serves no other entity. Rotation lowers the value of an answer rather than its correctness, since an address used for one transaction identifies a session rather than a subscriber. The simulator’s diffusion is kinder to attribution than the real network, so this is an upper bound.

Loss of coverage.

Discarding every announcement received by a random subset of the monitored peers emulates a collector that sees less of the network. Detection degrades gradually, but the frozen answering policy does not: its error among answers stays within the 5% target down to 15% coverage (4.36%), then rises to 14.04% at 10% and 34.37% at 5%. With fewer monitored peers, fewer of an actor’s own announcements are seen and relays become the best remaining candidates, while thresholds fitted at the original coverage let them through. When the policy is instead refitted at each coverage level, top-1 accuracy among answers stays between 0.974 and 0.983 from 5% to full coverage, and lower visibility reduces the number of answers rather than their correctness: the share of entities answered correctly falls from 0.854 at full coverage to 0.215 at 5%. A policy must therefore be recalibrated whenever the collector’s coverage changes, and a fall in the number of monitored peers should suspend attribution until it has been.

Exits, hosting and CoinJoin.

Announcing through a VPN, Tor or CDN exit is the most effective evasion of attribution, and the policy concedes it by design: an exit address belongs to a provider and is never named. I2P is not a separate class in the enrichment database, so a transaction relayed into the clearnet by a node connected to both networks could present that node as the originator; this case was not simulated, but the withholding experiment above is its closest analogue. Running a wallet on hosting infrastructure defeats the policy, as [Section 6.7](#) shows. CoinJoin participation is handled by the guard of [Section 4.2](#); rounds in which every participant brings several small inputs and takes no change would pass it, but merged entities matter little for attribution, because up to 5% of injected false merges changed top-1 accuracy among answers by less than 0.004.

Evading detection.

Where network features carry the synthetic signal, an offender can remove it by using the same infrastructure as ordinary users. Replacing every illicit test entity’s 19 network features with those of a randomly chosen licit entity, while leaving its chain features unchanged, cuts AP from 0.4725 to 0.0810 on demo and from 0.2527 to 0.0279 on bulk. The substitution is an idealised upper bound, since it assumes that the adversary can match every network statistic at once, but it shows that the synthetic-trained detector depends on cues an offender controls. On real data the chain-side model uses no network features at all ([Section 7](#)), which is why a deployment should rest on both layers rather than on network cues alone.

9 Conclusion

BTC-FUSION integrates Bitcoin P2P announcements with on-chain analytics to rank and explain attribution leads while abstaining when the fitted error bound does not support an answer. Across five independently generated captures, it achieves an AP of 0.1859 ± 0.0760 at a prevalence below 0.6%, with the error among answered cases remaining below 2.56%. In contrast, link-level significance produces many false links, demonstrating the importance of controlling error at the final attribution decision.

The evaluation also reveals important limitations. Performance varies across captures and evidence layers, while unseen laundering patterns and briefly observed entities are difficult to rank. Attribution degrades for actors using hosting infrastructure, under reduced observation coverage, forged first announcements, and when offenders mimic ordinary network profiles. On real Elliptic data, tree models outperform the tested graph baselines, although performance declines on illicit activity following a dark-market closure and remains sensitive to data-splitting choices. Network-layer validation is currently simulation-based, and the generator parameters represent engineering choices rather than measured real-world distributions. Consequently, the synthetic results characterize method behavior rather than operational accuracy. Future work should conduct lawful multi-vantage-point Bitcoin network measurements using researcher-controlled transactions and evaluate robustness to relay manipulation and other network countermeasures. The current prototype supports such real-network validation but does not yet establish operational attribution accuracy.

Acknowledgements. The work began as the authors’ submission to Smart India Hackathon 2026 for Problem Statement 26146, posed by the National Technical Research Organisation (NTRO), Government of India. The views expressed are those of the authors and do not necessarily reflect those of NTRO or of the Smart India Hackathon. IP-to-ASN enrichment uses the DB-IP Lite database (db-ip.com), licensed under CC BY 4.0. The authors thank the creators of the Elliptic and Elliptic++ datasets for making them available.

Declarations

- **Funding.** The authors received no funding for this work.
- **Competing interests.** The authors declare that they have no competing interests.
- **Ethics approval and consent to participate.** Not applicable. The study involves no human participants and no personal data; it uses a synthetic simulator and two public, de-identified datasets.
- **Consent for publication.** Not applicable.
- **Data availability.** The Elliptic and Elliptic++ datasets are third-party datasets that are not redistributed here; they are available from their original publishers [6, 7]. The synthetic captures are not distributed because of their size, but they can be regenerated byte for byte from their seeds.
- **Materials availability.** Not applicable.
- **Code availability.** The source code, generator configuration, random seeds, pinned model artefacts and the research artefacts behind the results are held in a private project repository. They are available from the corresponding author on reasonable request.

References

- [1] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, “A fistful of bitcoins: Characterizing payments among men with

- no names,” in *Proceedings of the 2013 Internet Measurement Conference (IMC '13)*, pp. 127-140, 2013, doi:[10.1145/2504730.2504747](https://doi.org/10.1145/2504730.2504747).
- [2] P. Koshy, D. Koshy, and P. McDaniel, “An analysis of anonymity in Bitcoin using P2P network traffic,” in *Financial Cryptography and Data Security (FC 2014)*, LNCS, Springer, pp. 469-485, 2014, doi:[10.1007/978-3-662-45472-5_30](https://doi.org/10.1007/978-3-662-45472-5_30).
 - [3] A. Biryukov, D. Khovratovich, and I. Pustogarov, “Deanonymisation of clients in Bitcoin P2P network,” in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*, pp. 15-29, 2014, doi:[10.1145/2660267.2660379](https://doi.org/10.1145/2660267.2660379).
 - [4] G. Fanti, S. Bojja Venkatakrishnan, S. Bakshi, B. Denby, S. Bhargava, A. Miller, and P. Viswanath, “Dandelion++: Lightweight cryptocurrency networking with formal anonymity guarantees,” *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 2, pp. 1-35, 2018, doi:[10.1145/3224424](https://doi.org/10.1145/3224424).
 - [5] T. Neudecker and H. Hartenstein, “Could network information facilitate address clustering in Bitcoin?” in *Financial Cryptography and Data Security: FC 2017 International Workshops (BITCOIN)*, LNCS, Springer, pp. 155-169, 2017, doi:[10.1007/978-3-319-70278-0_9](https://doi.org/10.1007/978-3-319-70278-0_9).
 - [6] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, “Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics,” presented at the KDD '19 Workshop on Anomaly Detection in Finance, 2019, arXiv:1908.02591, doi:[10.48550/arXiv.1908.02591](https://doi.org/10.48550/arXiv.1908.02591).
 - [7] Y. Elmougy and L. Liu, “Demystifying fraudulent transactions and illicit nodes in the Bitcoin network for financial forensics,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*, pp. 3979-3990, 2023, doi:[10.1145/3580305.3599803](https://doi.org/10.1145/3580305.3599803).
 - [8] M. Harrigan and C. Fretter, “The unreasonable effectiveness of address clustering,” in *Proceedings of the 2016 International IEEE Conferences on UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld*, pp. 368-373, 2016, doi:[10.1109/UIC-ATC-ScalCom-CBDCCom-IoP-SmartWorld.2016.0071](https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCCom-IoP-SmartWorld.2016.0071).
 - [9] R. Stütz, J. Stockinger, P. Moreno-Sanchez, B. Haslhofer, and M. Maffei, “Adoption and actual privacy of decentralized CoinJoin implementations in Bitcoin,” in *Proceedings of the 4th ACM Conference on Advances in Financial Technologies (AFT '22)*, pp. 254-267, 2022, doi:[10.1145/3558535.3559782](https://doi.org/10.1145/3558535.3559782).
 - [10] P. L. Juhász, J. Stéger, D. Kondor, and G. Vattay, “A Bayesian approach to identify Bitcoin users,” *PLoS ONE*, vol. 13, no. 12, art. e0207000, 2018, doi:[10.1371/journal.pone.0207000](https://doi.org/10.1371/journal.pone.0207000).

- [11] S. Zhang, B. Han, C. Tian, R. Shi, L. Lan, and Q. Wang, “Deanonymizing Bitcoin transactions via network traffic analysis with semi-supervised learning,” arXiv:2603.17261 (preprint), 2026, doi:10.48550/arXiv.2603.17261.
- [12] M. Šafář, J. Pluskal, V. Veselý, and O. Ryšavý, “The enemy of reproducibility is opacity: What’s inside the Elliptic bitcoin dataset (and why it is wrong),” *Forensic Science International: Digital Investigation*, vol. 57 (DFRWS USA 2026), art. 302124, 2026, doi:10.1016/j.fsidi.2026.302124.
- [13] S. Maganti, “When graph structure becomes a liability: A critical re-evaluation of graph neural networks for Bitcoin fraud detection under temporal distribution shift,” arXiv:2604.19514 (preprint), 2026, doi:10.48550/arXiv.2604.19514.
- [14] S. M. Lundberg, G. Erion, H. Chen, et al., “From local explanations to global understanding with explainable AI for trees,” *Nature Machine Intelligence*, vol. 2, no. 1, pp. 56-67, 2020, doi:10.1038/s42256-019-0138-9.
- [15] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, “Leakage in data mining: Formulation, detection, and avoidance,” *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, art. 15, pp. 1-21, 2012, doi:10.1145/2382577.2382579.
- [16] A. N. Angelopoulos, S. Bates, E. J. Candès, M. I. Jordan, and L. Lei, “Learn then test: Calibrating predictive algorithms to achieve risk control,” *The Annals of Applied Statistics*, vol. 19, no. 2, pp. 1641-1662, 2025, doi:10.1214/24-AOAS1998.
- [17] C. J. Clopper and E. S. Pearson, “The use of confidence or fiducial limits illustrated in the case of the binomial,” *Biometrika*, vol. 26, no. 4, pp. 404-413, 1934, doi:10.1093/biomet/26.4.404.
- [18] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pp. 855-864, 2016, doi:10.1145/2939672.2939754.
- [19] B. Zadrozny and C. Elkan, “Transforming classifier scores into accurate multiclass probability estimates,” in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pp. 694-699, 2002, doi:10.1145/775047.775151.
- [20] M. Saelens, P. Latinne, and C. Decaestecker, “Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure,” *Neural Computation*, vol. 14, no. 1, pp. 21-41, 2002, doi:10.1162/089976602753284446.
- [21] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, pp. 413-422, 2008, doi:10.1109/ICDM.2008.17.

- [22] Y. Benjamini and Y. Hochberg, “Controlling the false discovery rate: A practical and powerful approach to multiple testing,” *Journal of the Royal Statistical Society: Series B*, vol. 57, no. 1, pp. 289-300, 1995, [doi:10.1111/j.2517-6161.1995.tb02031.x](https://doi.org/10.1111/j.2517-6161.1995.tb02031.x).
- [23] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” *PLoS ONE*, vol. 10, no. 3, art. e0118432, 2015, [doi:10.1371/journal.pone.0118432](https://doi.org/10.1371/journal.pone.0118432).
- [24] M. Pakdaman Naeini, G. F. Cooper, and M. Hauskrecht, “Obtaining well calibrated probabilities using Bayesian binning,” in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-15)*, pp. 2901-2907, 2015, [doi:10.1609/aaai.v29i1.9602](https://doi.org/10.1609/aaai.v29i1.9602).
- [25] Y. Benjamini and D. Yekutieli, “The control of the false discovery rate in multiple testing under dependency,” *The Annals of Statistics*, vol. 29, no. 4, pp. 1165-1188, 2001, [doi:10.1214/aos/1013699998](https://doi.org/10.1214/aos/1013699998).
- [26] R. Wang and A. Ramdas, “False discovery rate control with e-values,” *Journal of the Royal Statistical Society: Series B*, vol. 84, no. 3, pp. 822-852, 2022, [doi:10.1111/rssb.12489](https://doi.org/10.1111/rssb.12489).