Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

FraudBench-NG: A NIBSS-Calibrated Synthetic Dataset and Generator for Nigerian Inter-Bank Fraud Detection

Otu Jacob
Wrexham University
Colin Kuka
Colin.Kuka@wrexham.ac.uk

Wrexham University
Adam Knowles
Wrexham University
Cyrus Orji
Imo State Polytechnic Omuma

Research Article

Keywords:

Posted Date: November 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11431606/v1

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

Additional Declarations: No competing interests reported.

FraudBench-NG: A NIBSS-Calibrated Synthetic Dataset and Generator for Nigerian Inter-Bank Fraud Detection

Otu Jacob[1], Colin Kuka[1*], Adam Knowles[1], Cyrus Ebere Orji[2]

[1]School of Cyber and Computing, Wrexham University, Wrexham, United Kingdom.
[2]Department of Computer Science, Imo State Polytechnic Omuma, Imo State, Nigeria.

*Corresponding author(s). E-mail(s): Colin.Kuka@wrexham.ac.uk;
Contributing authors: otujacob4@gmail.com;
Adam.Knowles@wrexham.ac.uk; cyrus.orji@imopoly.edu.ng;

Abstract

Research on inter-bank fraud detection in Nigeria is limited by the lack of shareable transaction data, and synthetic benchmarks can mislead if they leak information or hide their assumptions. FraudBench-NG is a reproducible framework for studying fraud detection under explicitly documented modelling assumptions. It is a parametric simulation calibrated to 2023 NIBSS aggregates, with a released dataset of 2.1 million transactions from 25,000 customers and 0.1% fraud. Fraud is generated as incidents, matching NIBSS counts and loss per case; every parameter is marked as NIBSS or assumed; and every behavioural feature uses only prior transactions, verified by row-by-row recomputation and a label-permutation test. Logistic regression, random forest and XGBoost reach a PR-AUC of 0.016–0.036 (no-skill 0.0007), with wide customer-level bootstrap intervals, so the models cannot be ranked. At a 1% false-positive rate they raise about 70 false alerts per fraud row caught and flag only about 19% of incidents at their first transaction. Results depend strongly on the assumptions: a model trained with takeover bursts transfers only partly to data with a different burst probability, and performance differs between burst settings mainly because of how an incident's loss is split across its transactions. Fraud prevalence is a design choice that cannot be compared with the NIBSS case-to-transaction ratio, and no real Nigerian transactions were available for validation.

Keywords: Synthetic data, fraud detection, Nigerian inter-bank payments, dataset generation, NIBSS, data leakage

1 Introduction

Digital payments in Nigeria have grown rapidly, and so has fraud. The Nigeria Inter-Bank Settlement System (NIBSS) recorded 95,620 fraud cases and NGN 17.7 billion in losses across inter-bank channels in 2023, over 90% of which by count involved mobile, web and point-of-sale channels [1]. Later NIBSS figures give 70,111 cases and NGN 52.26 billion for 2024 (NGN 31.1 billion of it a single incident) and 67,518 cases and NGN 25.85 billion for 2025 [2, 3], but, to our knowledge, without the breakdown by channel, month and technique that a transaction-level generator needs (Section 3.1). Machine-learning fraud detection for this setting needs transaction-level data, but bank transaction records cannot be shared for confidentiality and regulatory reasons, and the aggregate statistics NIBSS publishes contain no individual transactions. Researchers therefore work with either small private samples that cannot be released, or with datasets from other countries and payment systems.

FraudBench-NG provides a reproducible framework for investigating financial fraud detection under explicitly documented modelling assumptions. Beyond reproducing selected NIBSS aggregates, the benchmark enables researchers to examine how synthetic fraud-generation mechanisms influence machine-learning performance, temporal feature construction and model robustness. Our central message is that the assumptions behind a synthetic dataset matter when evaluating AI models, and the paper therefore shows how benchmark results change when those assumptions change.

1.1 Leakage in synthetic benchmarks.

A synthetic benchmark can look rigorous and still hand the model the answer. A feature computed over a customer's whole year gives a January transaction knowledge of December; a window that contains the transaction being judged lets an unusual amount hide inside its own average; a score derived from the generator's own fraud weights lets a model learn the generator. Such flaws do not announce themselves in headline metrics: they inflate them. FraudBench-NG was therefore designed so that fraud status and amounts are fixed first, every behavioural feature is computed from strictly prior transactions, and every assumed quantity is declared and kept out of the default model features. Each of these properties is tested directly (Section 4).

1.2 Research questions.

We structure the work around three questions.

RQ1 Can a synthetic Nigerian banking fraud benchmark reproduce selected published NIBSS aggregate properties while maintaining internal and temporal consistency?
RQ2 Does enforcing causal feature generation avoid temporal leakage and redundant features, as verified by direct tests on the released data?

RQ3 How sensitive are benchmark performance and model explanations to the generator's assumptions (random seed, takeover-burst probability, loss dispersion, merchant and channel mix), do models trained under one assumption generalise to data generated under another, and what does the resulting detection task mean for a bank?

1.3 Contributions.

This paper makes five contributions.

1. FraudBench-NG, a parametric synthetic benchmark and transparent generator, specified mathematically (Section 3). Fraud status and final amounts are generated before behavioural features; every feature is computed from strictly prior transactions; fraud is generated as incidents with one loss per incident, split over its transactions (matching how NIBSS reports cases); merchants are channel-compatible; the generator's own merchant assumption is kept out of the default model features; and every other parameter is declared as an assumption (Section 3 and Supplementary Appendix A). Fraud channel mix, monthly profile, technique mix and mean loss per case follow the 2023 NIBSS Annual Fraud Landscape (RQ1). We claim that FraudBench-NG is transparent, auditable and assumption-aware, not that it is realistic in every detail.
2. Customer-level behaviour (spend level, activity, preferred channel and bank, diurnal and weekly profile, bursts), with strictly causal behavioural features verified against brute-force recomputation and a label-permutation test (RQ1, RQ2).
3. A systematic evaluation (Section 4) against NIBSS statistics and statistical sanity tests, a comparison of logistic regression, random forest and XGBoost with customer-clustered bootstrap intervals, and a SHAP analysis of what the models use (RQ1–RQ3).
4. An investigation of how the generator's assumptions influence fraud-detection results: a ten-seed sensitivity analysis, results for first fraud events and follow-ons separately, an ablation that explains why performance is not monotone in the takeover-burst probability, a test of whether a model trained under one assumption generalises to data generated under another, and the alert burden and detection timing that the results imply for a bank (RQ3).
5. An open generator and dataset, with fraud prevalence set to 0.1% as a stated design choice (not comparable with the NIBSS case-to-transaction ratio), a register that gives the source or justification of every design choice, and a list of what is not modelled (Section 6).

The remainder of the paper is organised as follows. Section 2 reviews existing public fraud datasets and the gap this work addresses, Section 3 specifies the FraudBench-NG generator, Section 4 presents the results for the three research questions, Section 5 discusses them, Section 6 lists limitations, and Section 7 concludes. Data and code availability are given in the Declarations. The supplementary appendices hold supporting material in the order in which the main text first refers to it: the register of design choices (Supplementary Appendix A), what the generator does not model (Supplementary Appendix B), the variable reference (Supplementary Appendix C), the NIBSS calibration data (Supplementary Appendix D), the assumed activity weights

and merchant factors (Supplementary Appendix E), and additional validation tests and figures (Supplementary Appendix F).

2 Related work

2.1 Public fraud datasets

Publicly available fraud datasets fall into two groups: real data that has been anonymised, and simulated data.

The most widely used real dataset is the credit-card dataset of the Université Libre de Bruxelles (ULB) [4, 5]. It holds 284,807 European card transactions made over two days, of which 492 (0.172%) are fraudulent. For confidentiality, all but two of its input variables are principal components of the original features; only the elapsed time and the amount are interpretable. It therefore supports benchmarking of classifiers but cannot be used to study behavioural features, channels or customer segments, and it spans too short a period for studies of concept drift or seasonality. The IEEE-CIS competition dataset [6] contains about 590,000 real e-commerce transactions with roughly 3.5% fraud and hundreds of masked features; its prevalence is far above that of any national payment system, and its feature semantics are hidden. Neither dataset describes Nigerian payments.

Simulated datasets remove the confidentiality barrier. PaySim [7] is an agent-based simulator of mobile-money transactions, calibrated on logs from a mobile-money service in an African country; its public dataset has 6.36 million transactions over one simulated month, with fraud (about 0.13%) restricted to two transaction types. BankSim [8] simulates bank payments for fraud research, calibrated on aggregated card-payment data from a Spanish bank. These simulators are established tools and FraudBench-NG belongs to the same family; they differ from our setting in the following respects. They are calibrated, if at all, on a single source in a different country or payment system; they do not model Nigerian channels (mobile, web, POS, internet banking, e-commerce, ATM) or bank customers; their documentation seldom states which parameters are measured and which are assumed; and their evaluation is mostly limited to training classifiers on the result, which cannot show that the data resemble real transactions. A further common weakness of simulated data is that a feature derived from the label, or computed over a customer's whole history including the future, lets a classifier reach near-perfect scores that disappear in deployment. Feature-engineering studies show that how aggregates are constructed (window length, which transactions are included) changes detection performance substantially [9], so a dataset used to compare models must define its aggregates causally and unambiguously.

2.2 Gap addressed by this work

We know of no public dataset that documents (i) transaction-level records for a Nigerian inter-bank setting, (ii) fraud whose channel, monthly, technique and amount profile follow a published national source, (iii) customer-level behaviour with heterogeneity and burstiness, and (iv) causal, auditable feature definitions. A Kaggle listing describes 1 million "NIBSS-calibrated" Nigerian banking transactions [10]; its page

states no generation method, fraud rate or licence, so we could not assess it or compare with it. Without a documented benchmark, claims about fraud-detection methods for Nigeria can neither be reproduced nor compared. We have not had access to real NIBSS transaction records, so we make no claim that a model trained on our data would work on real transactions. The aim is a transparent benchmark whose properties are verified against published statistics and against tests of internal validity, and whose simplifications are stated openly (Section 6).

Table 1 compares FraudBench-NG with established fraud benchmarks in terms of calibration source, temporal behaviour, fraud incident modelling, feature causality, parameter transparency and sensitivity analysis. This comparison highlights the methodological contributions of FraudBench-NG and the limitations it shares with other synthetic datasets: it is simulated, calibrated only to published aggregates, and not validated against real Nigerian transactions.

Table 1 Comparison of FraudBench-NG with established public fraud datasets (ULB [4], IEEE-CIS [6], PaySim [7], BankSim [8]). "n/a": not applicable to a real dataset. "Not reported": we found no statement in the published descriptions, which does not mean the property is absent.

	ULB	IEEE-CIS	PaySim	BankSim	FraudBench-NG
Origin	Real, anonymised	Real, masked	Simulated	Simulated	Simulated
Payment setting	European cards	E-commerce cards	Mobile money	Bank card payments	Nigerian inter-bank, six channels
Calibration source	n/a	n/a	One mobile-money service log	Aggregated Spanish bank card data	NIBSS 2023 aggregates
Interpretable features	Time, amount only	Hidden	Yes	Yes	Yes, with variable reference
Fraud as multi-transaction incidents	Not available	Not available	Not reported	Not reported	Yes, one loss per incident
Features from prior transactions only, recomputation tested	n/a	n/a	Not reported	Not reported	Yes
Parameters labelled measured or assumed	n/a	n/a	Not reported	Not reported	Yes, register
Sensitivity analysis of assumptions	n/a	n/a	Not reported	Not reported	Yes
Real Nigerian data used	No	No	No	No	No

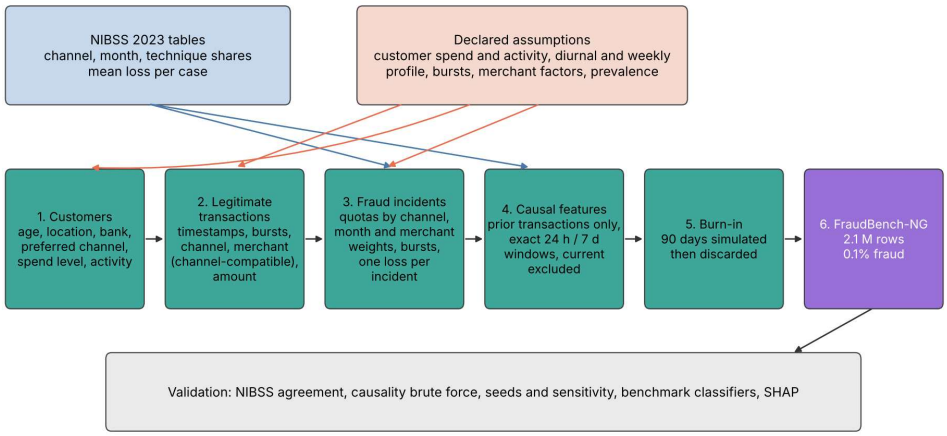


Fig. 1 Architecture of the FraudBench-NG generator. Fraud status and final amounts are fixed first; behavioural features are then computed once, from strictly prior transactions only.

3 The FraudBench-NG generator

FraudBench-NG is produced by a customer-level generative process (Fig. 1). Customers are created first with fixed attributes and behavioural parameters; transactions are drawn conditionally on those parameters; fraud events are injected so that their channel, month, technique and loss profile follow the 2023 NIBSS Annual Fraud Landscape [1]; and every behavioural feature is then computed from the final amounts using only information available at the time of the transaction. It is a parametric simulation calibrated to published aggregates, in the same family as PaySim and BankSim; learned generators such as GANs or copulas would need real transaction data to train on. Three principles guide the construction: *causality* (no feature uses future information), *internal consistency* (every derived feature is computed once, from the same final values) and *transparent provenance* (every parameter is either taken from a published NIBSS figure or declared as an assumption; Table 2). In `generate_dataset.py` each parameter is labelled `[NIBSS]` or `[ASSUMED]`. Supplementary Appendix A (Table A1) gives the source or justification of every parameter, Supplementary Appendix B lists what the generator does not model, and Supplementary Appendix C (Table C3) defines every variable in the dataset. We do not claim that the data are realistic in every detail; the claim is that FraudBench-NG is *transparent*, *auditable* (every feature can be recomputed from prior transactions) and *assumption-aware* (the assumptions that drive difficulty are named and varied).

3.1 Calibration data and its limits

The calibration uses the NIBSS Annual Fraud Landscape for 2023 [1], the most recent NIBSS publication we could find that tabulates fraud by channel, month and technique; later releases give only aggregates (Supplementary Appendix D, Table D4). Recalibrating on 2024 or 2025 would require profiles that have not been published,

Table 2 Provenance of the generator's parameters. Parameters marked *assumed* have no published source; they are modelling choices that users may change.

Parameter	Source	Used for
Fraud cases (incidents) by channel	NIBSS [1]	Channel mix of fraud incidents
Fraud count by month	NIBSS [1]	Monthly profile of fraud cases
Fraud count by technique	NIBSS [1]	Technique mix (six classes)
Mean loss per case by channel	NIBSS [1]	Mean loss of one incident, split over its transactions
Overall prevalence (0.1%)	Design choice	Class balance, enriched relative to NIBSS
Customer spend, activity, burstiness	Assumed	Customer heterogeneity
Diurnal and weekly profile	Assumed	Timing of transactions
Takeover-burst probability $\phi = 0.4$	Assumed	Follow-on transactions per incident
Split of incident loss, channel–merchant compatibility	Assumed	Amounts of rows within an incident; merchant of each row
Merchant fraud factors	Assumed	Merchant-category effect

and the 2024 loss total is dominated by a single incident of NGN 31.1 billion [3]. Case counts fell by 27% (2024) and 29% (2025) relative to 2023, and NIBSS reports that Lagos accounts for 63.43% of fraud volume in 2025 [3], so the 2023 channel mix (mobile 49.8%, web 22.9%) and the absence of any location effect may no longer reflect current conditions. NIBSS tables also count only *reported* fraud. All calibration constants sit in one block of `generate_dataset.py`, so later tables can be substituted.

The tables of the 2023 report do not agree with one another (Supplementary Appendix D, Table D5): the channel rows sum to 95,525 cases against a stated 95,620 (which the monthly and technique tables reproduce), and the channel losses sum to NGN 16,636.1m against NGN 17,666.8m stated, a gap of 5.8%. The generator never needs the tables to agree, because each is used only through quantities normalised within it: channel shares are $s_c = n_c/95{,}525$, monthly and technique shares are normalised by 95,620, and the mean loss per case of a channel is that channel's loss divided by its cases. The unallocated loss means that fraud amounts are about 6% too low if it belongs to the six channels in proportion, and since NIBSS reports *actual* loss (attempted fraud was 11% higher), loss per case understates attempted amounts.

3.2 Customers

N_c customers are generated. Customer i receives an age group g_i, a registered location ℓ_i, a primary bank β_i and a preferred channel π_i, fixed over the year and drawn independently of one another: age (18–25 18%, 26–35 30%, 36–45 24%, 46–55 15%, 56–65 8%, 65+ 5%), location (Lagos 28%, Abuja 20%, Oyo 14%, Ogun 12%, Rivers 11%, other 15%), bank (uniform over ten anonymised banks) and channel from the population mix $\mathbf{p}_c$ (ATM 5%, ECOM 11%, IB 11%, Mobile 42%, POS 16%, Web

15%). These distributions represent a hypothetical young, urban, mobile-first banking population rather than an empirically validated sample of Nigerian banking customers. We found no published customer-level distribution of age or location to match them to. National statistics support only the general direction: the EFInA Access to Finance Survey 2023 reports 52% of adults as banked, mobile-phone adoption of 93%, and exclusion of 37% of rural against 17% of urban adults [11], and Lagos holds 6.2% of Nigerians [12] against 28% of our customers. The values are therefore configurable assumptions. Age group, location and bank are *inert*: they do not influence any other variable, so their influence on model performance cannot be tested, and their presence should not be mistaken for demographic structure. The one population assumption that matters, the legitimate channel mix, is varied in Section 4.3.

Each customer also has a spending level and dispersion,

$$\mu_i \sim \mathcal{N}(10.8,\, 0.6^2), \qquad \sigma_i \sim \text{Uniform}(0.4,\, 0.9), \tag{1}$$

which parametrise the log-normal distribution of that customer's legitimate amounts (Eq. 6); μ_i and σ_i are the mean and standard deviation of the *logarithm* of the amount. The number of transactions of customer i is

$$n_i = \min\big(\max(\lfloor X_i \rfloor,\, 5),\, 1000\big), \qquad X_i \sim \text{LogNormal}(4.2,\, 0.7^2), \tag{2}$$

where $\text{LogNormal}(a, b^2)$ is the distribution of e^Y with $Y \sim \mathcal{N}(a, b^2)$, giving the long-tailed customer activity typical of positive, right-skewed quantities [13]. The median customer makes about 67 transactions a year (mean about 85) with a typical amount of about NGN 49,000; counts are scaled by 455/365 over the 455 simulated days. These values are design choices, not estimates, because NIBSS publishes no distribution of legitimate amounts or customer activity. Every customer is active throughout; there is no churn.

3.3 Transaction timestamps

A transaction's base time is drawn from a day-of-year and an hour-of-day distribution,

$$P(d) \propto w^{\text{dow}}_{\text{dow}(d)}, \quad P(h) \propto w^{\text{hour}}_{h}, \quad t = 86{,}400\, d + 3{,}600\, h + U, \quad U \sim \text{Uniform}(0, 3600), \tag{3}$$

where $d \in \{0, \ldots, 364\}$ indexes the day of 2023, $\text{dow}(d)$ is its weekday, h the hour, w^{dow} and w^{hour} relative weights (Table E7, Supplementary Appendix E), U an offset within the hour and t the time in seconds from 1 January 2023. The weights follow the generic pattern of human activity (low at night, morning and evening peaks, lighter weekends); NIBSS publishes no such statistics, so they are assumptions applied to every customer, with no month-end, salary-day, holiday or trend effects. Hour, day-of-week and month are also provided as sine and cosine pairs, $\sin(2\pi x/P)$ and $\cos(2\pi x/P)$ with $P \in \{24, 7, 12\}$ [9], and a peak-hour indicator marks 09:00–18:00.

Human event times are bursty rather than Poisson [14]. Each non-first transaction of a customer therefore follows the previous one with probability $\rho = 0.25$ after an

exponential delay of mean 15 minutes,

$$t_{i,k} = t_{i,k-1} + \Delta_k, \quad \Delta_k \sim \text{Exp}(\text{mean } 900\,\text{s}) \quad \text{with probability } \rho, \tag{4}$$

where $t_{i,k}$ is the time of customer i's k-th transaction and Δ_k the delay; otherwise the transaction keeps its base time, and a customer's transactions are then sorted by time. The values were chosen so that the coefficient of variation of inter-arrival times clearly exceeds the Poisson value of one (1.78 in the output); the true burstiness of Nigerian inter-bank payments is unknown to us.

3.4 Channel, bank and merchant

Bank is fixed per customer. Channel is sticky but not absolute: with probability 0.70 the customer uses the preferred channel π_i, otherwise a channel is drawn from $\mathbf{p}_c$,

$$c_{i,k} = \begin{cases} \pi_i & \text{with probability } 0.70, \\ \tilde{c} \sim \text{Categorical}(\mathcal{C}, \mathbf{p}_c) & \text{with probability } 0.30, \end{cases} \tag{5}$$

where $c_{i,k}$ is the channel of customer i's k-th transaction and $\mathcal{C}$ the set of six channels. Merchant category is drawn *uniformly among the categories compatible with the channel*, independently of customer, amount and time. The compatibility table (Table E9, Supplementary Appendix E) is a simple judgement that removes impossible combinations (for example ATM withdrawals in other channels, or transfers at card terminals); we have no source for how payments divide across the thirteen categories, which are a design choice. The consequences are listed in Section 6.

3.5 Transaction amount

A legitimate amount is log-normal around the customer's own spending level,

$$A_{i,k} \sim \text{LogNormal}\big(\mu_i + \delta_{c_{i,k}},\ \sigma_i^2\big), \tag{6}$$

where $A_{i,k}$ is the amount of customer i's k-th transaction, $c_{i,k}$ its channel, μ_i and σ_i the customer's spending parameters (Eq. 1) and δ_c an assumed channel shift on the log scale (ATM -0.30, ECOM -0.10, POS -0.10, Mobile 0, Web $+0.10$, IB $+0.50$). The loss of a fraud *incident* does not depend on the victim's spending; NIBSS reports loss and counts per case, so the unit of calibration is the incident. The incident loss L_j is drawn from a channel-specific log-normal whose mean is the NIBSS mean loss per case $\bar{A}_c^{\text{NIBSS}}$ (Table D6, Supplementary Appendix D),

$$L_j \sim \text{LogNormal}\Big(\ln \bar{A}_{c_j}^{\text{NIBSS}} - \tfrac{s^2}{2},\ s^2\Big), \qquad s = 0.9, \tag{7}$$

where c_j is the channel of incident j and s the standard deviation of the log loss; the term $s^2/2$ makes the mean of the draw equal $\bar{A}_c^{\text{NIBSS}}$, because $\mathbb{E}[X] = e^{a+b^2/2}$ for $X \sim \text{LogNormal}(a, b^2)$ [13]. The dispersion $s = 0.9$ is assumed because NIBSS

reports only means; it was chosen so that fraud amounts overlap the legitimate ones, and sensitivity runs use $s = 0.5$ and 1.3. Because each incident's loss is split over its transactions, the amount of a single fraud row averages about the NIBSS loss per case divided by 1.63, and amount alone separates the classes only weakly (single-feature ROC-AUC 0.48, Kolmogorov–Smirnov distance 0.11).

3.6 Fraud label generation

The target prevalence is $\pi_F = 0.1\%$ of all transactions, so the number of fraud rows is $F = \text{round}(\pi_F N)$. This target provides enough positive examples for machine-learning experiments. NIBSS reported 95,620 fraud cases against approximately 11.27 billion transactions in 2023 [1], roughly 0.00085 reported cases per 100 transactions. A case may contain several fraudulent transactions, so this ratio is not a transaction-level prevalence, and our 0.1% is a deliberately enriched design choice, not a measured multiple of the real rate.

Fraud is generated at the level of *incidents*. With continuation probability ϕ (below), the number of incidents is $J = \text{round}(F(1-\phi))$, since an incident has on average $1/(1-\phi)$ transactions. For each channel c a quota $J_c = \text{round}(J\, s_c)$ of incidents is allocated, where s_c is the NIBSS share of cases in channel c, and candidate *initiating* transactions are sampled without replacement with probability proportional to

$$\omega_k = f_M(\text{month}(t_k)) \cdot f_{\text{cat}}(\kappa_k), \qquad f_M(m) = \frac{q_m}{\text{days}_m/365}, \tag{8}$$

where ω_k is the sampling weight of candidate k, κ_k its merchant category, q_m the NIBSS share of 2023 fraud cases in month m, days_m the number of days in month m (so f_M is the monthly intensity per day) and f_{cat} an assumed merchant-category factor (Section 3.8). Fraud probability does not depend on bank, location, age, spending level or prior behaviour, so a customer's chance of being chosen is proportional to their transactions in the channel; about 4.8% of customers have at least one fraud row, and the data contain no demographic or geographic fraud signal, unlike reported patterns (customers aged 40 and above [2]; Lagos 63.43% of volume [3]).

3.6.1 Takeover bursts.

Account-takeover fraud typically produces several fraudulent transactions in quick succession. After every fraudulent transaction a further one follows with probability ϕ, so the number E of follow-ons after the initiating transaction is geometric,

$$P(E = e) = (1-\phi)\,\phi^{\,e}, \qquad \mathbb{E}[E] = \frac{\phi}{1-\phi} \approx 0.67 \quad (\phi = 0.4), \tag{9}$$

so that 60% of incidents have one transaction, 24% two, 9.6% three and 6.4% four or more (1.63 on average in the released data; follow-ons are 39% of fraud rows). Each follow-on occurs after the previous event with an exponential delay of mean 300 seconds, belongs to the same customer, channel and technique as the initiator, and

draws its merchant uniformly among the compatible categories *without* the weights f_{cat}.

The value $\phi = 0.4$ is an illustrative baseline, not an empirically estimated takeover probability: no NIBSS statistic reports transactions per case, and we do not consider it representative of Nigerian account-takeover fraud. With $\phi = 0$ there are no bursts and the velocity features carry no fraud signal, while at $\phi = 0.7$ they dominate (Section 4.3). Because difficulty depends largely on this assumption, we report results at $\phi = 0$, 0.4 and 0.7 and separately for first events and follow-ons (Section 4.4).

3.6.2 One loss per incident.

An incident's total loss L_j is drawn once (Eq. 7) and split across its m_j transactions in proportions $(g_1, \ldots, g_{m_j}) \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$ with $\alpha = 2$ (assumed; moderately uneven shares), so the row amounts $L_j g_r$ (floored at NGN 1) sum to the incident loss and the mean loss per incident, not per row, reproduces the NIBSS figure by channel (Section 4.1). Section 4.3 shows that this splitting rule, not the bursts themselves, largely determines why performance differs between $\phi = 0$ and $\phi = 0.4$.

3.7 Causal behavioural features

Let $\mathcal{H}_{i,k} = \{(t_{i,j}, A_{i,j}, c_{i,j})\}_{j<k}$ be the *prior* history of customer i: all transactions strictly before transaction k. The current transaction is excluded from every baseline and window; otherwise an unusually large (for example fraudulent) amount would inflate its own reference mean, and a 24-hour count that includes the current transaction would be at least one in every row, so that a velocity score built from it would merely repeat the amount ratio. Every feature below is therefore a function of $\mathcal{H}_{i,k}$ only (Fig. 2); the current amount appears only as the numerator of $R_{i,k}$. With $k-1$ prior transactions, the running mean and standard deviation of past amounts are

$$\bar{A}_{i,k} = \frac{1}{k-1} \sum_{j<k} A_{i,j}, \tag{10}$$

$$s_{i,k} = \sqrt{\tfrac{1}{k-1} \textstyle\sum_{j<k} (A_{i,j} - \bar{A}_{i,k})^2}, \tag{11}$$

computed by cumulative sums (equivalently Welford's recurrence [15]). Let $\mathcal{W}_{24h}(k) = \{j < k : t_{i,k} - t_{i,j} \le 86{,}400\,\text{s}\}$ and $\mathcal{W}_{7d}(k) = \{j < k : t_{i,k} - t_{i,j} \le 604{,}800\,\text{s}\}$ be the trailing windows, measured in exact seconds. The windowed aggregates are

$$n^{24h}_{i,k} = |\mathcal{W}_{24h}(k)|, \qquad \Sigma^{24h}_{i,k} = \sum_{j \in \mathcal{W}_{24h}(k)} A_{i,j}, \tag{12}$$

$$\bar{A}^{7d}_{i,k} = \frac{1}{|\mathcal{W}_{7d}(k)|} \sum_{j \in \mathcal{W}_{7d}(k)} A_{i,j}, \qquad s^{7d}_{i,k} = \text{std}\{A_{i,j}\}_{j \in \mathcal{W}_{7d}(k)}. \tag{13}$$

If a window is empty, $\bar{A}^{7d}_{i,k}$ is replaced by $\bar{A}_{i,k}$ and $s^{7d}_{i,k}$ by zero. The relative amount and the velocity score are

$$R_{i,k} = \frac{A_{i,k}}{\bar{A}_{i,k}}, \tag{14}$$

$$V_{i,k} = \frac{n^{24h}_{i,k}\,\Sigma^{24h}_{i,k}}{\bar{A}_{i,k}}. \tag{15}$$

$R_{i,k}$ states how large the current amount is relative to the customer's history; $V_{i,k}$ is the *prior* 24-hour spend scaled by the average amount and multiplied by the number of prior transactions in that window, an engineered heuristic that does not depend on the current amount. For a customer's first transaction $R_{i,k} = 1$, $V_{i,k} = 0$ and $\bar{A}_{i,k} = A_{i,k}$. Finally,

$$D^{\mathrm{ch}}_{i,k} = |\{c_{i,j}\}_{j<k}|, \tag{16}$$

$$O_{i,k} = \frac{1}{k-1}\sum_{j<k} \mathbb{1}[c_{i,j} \in \{\text{Mobile}, \text{Web}, \text{ECOM}\}] \tag{17}$$

(with $\mathbb{1}[\cdot]$ equal to 1 if its condition holds and 0 otherwise) count the distinct channels used before and the share of online-channel transactions before, and the prior transaction count is $k-1$. No composite risk score is published, because a linear combination of existing features would duplicate information and inflate its apparent importance in attribution analyses.

3.7.1 Burn-in.

If every customer's history began on 1 January, early rows would have almost no history and late rows a long one, confounding fraud with history length in a chronological split. The generator therefore simulates a 90-day burn-in, computes all features over the full 455 days, injects fraud only in 2023 and discards the burn-in rows.

3.8 Merchant risk score and fraud technique

The merchant factors f_{cat} weight which transactions start a fraud incident (Eq. 8). We found no published data on fraud by merchant category for Nigeria, so *the factors are the authors' ordinal judgements*, ranging from 0.3 to 1.4 (1.0 is no tilt) according to how quickly stolen funds can be converted, whether the payment is instant and irreversible, whether the payee checks identity and can reverse payments, and whether a physical item or cash is involved (Table E8, Supplementary Appendix E). The published merchant risk score is a monotone rescaling of the same factor to $[0.1, 0.9]$,

$$\mathrm{risk}(m) = 0.1 + 0.8\,\frac{f_{\mathrm{cat}}(m) - \min_{m'} f_{\mathrm{cat}}(m')}{\max_{m'} f_{\mathrm{cat}}(m') - \min_{m'} f_{\mathrm{cat}}(m')}, \tag{18}$$

where m is a merchant category and the minimum and maximum run over all categories m'. The score is therefore the generator's own assumption written down as a

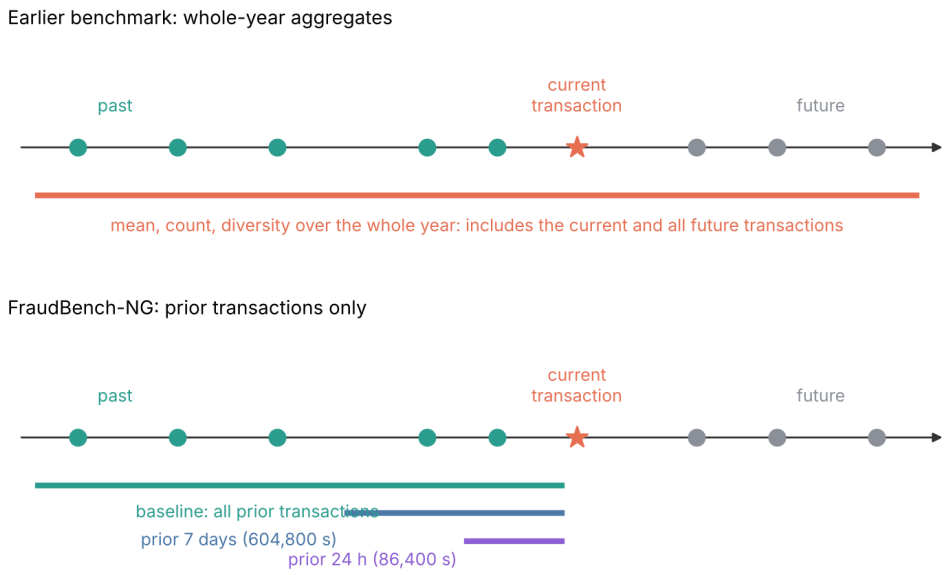


Fig. 2 Causal-feature construction: every feature of transaction k is a function of that customer's prior history $\mathcal{H}_{i,k}$ only; the current amount enters solely as the numerator of $R_{i,k}$.

feature: a model that uses it has learned the assumption, not an empirical risk. Because follow-on rows draw their merchant without the weights and channel compatibility changes the number of transactions per category, the realised tilt is weaker than the factors suggest (fraud rates per 1,000 transactions from 0.50 for medical to 1.35 for electronics, a ratio of 2.7 rather than 4.7; ROC-AUC of the score 0.58, Fig. F1 (Supplementary Appendix F)). **The score and the merchant category are excluded from the default model features** (Section 4.4); Section 4.3 tests how much results depend on the factors.

Fraud technique is drawn from the 2023 NIBSS shares (social engineering 65.8%, robbery 10.6%, card theft 7.1%, PIN compromise 5.5%, phishing 4.7%, other 6.3%) [1]. NIBSS tabulates 14 techniques; we keep the five largest and pool the other nine into "other" (5,985 cases) for simplicity, which loses information: internal collusion and server hacking are rare but account for NGN 1.0 bn and NGN 2.4 bn of 2023 losses, and the generator does not reproduce their cost. Technique is not conditioned on channel, because the NIBSS tables do not report their joint distribution.

4 Results

This section answers the three research questions in turn. Table 3 summarises the released dataset. Section 4.1 addresses RQ1 (agreement with NIBSS aggregates and internal and temporal consistency), Section 4.2 addresses RQ2 (whether causal generation avoids temporal leakage and redundant features), and Sections 4.3–4.6 address RQ3 (stability and transfer across assumptions, downstream difficulty, what the results

Table 3 Summary of the FraudBench-NG release

Property	Value
Transactions	2,102,151, 1 January–31 December 2023 (90-day burn-in discarded)
Customers	25,000; median 66 transactions each (mean 84, range 3–1,010)
Fraud rows	2,059 (0.098%), a design choice; not comparable with the NIBSS case-to-transaction ratio (about 0.00085%)
Fraud incidents	1,261 incidents (mean 1.63 rows each; 39% of fraud rows are follow-ons); 4.8% of customers have at least one fraud row
Columns	33: 13 raw (identifiers, time, amount, channel, bank, merchant, location, age, label, technique, incident identifier, follow-on indicator) and 20 engineered features
Categories	6 channels, 13 merchant categories, 10 banks, 6 fraud techniques
Chronological split	Train to 7 August (60%), validation to 18 October (20%), test 18 October–31 December (420,431 rows, 277 fraud: 172 first events and 105 follow-ons in 172 incidents)
Format	Parquet (about 160 MB); generator reproduces it from a seed

mean for a bank, and explainability). Additional tests and figures are in Supplementary Appendix F.

4.1 RQ1: agreement with NIBSS and internal consistency

4.1.1 NIBSS aggregates.

Figure 3 compares the synthetic fraud *incidents* with the NIBSS 2023 tables used for calibration; NIBSS counts and losses are per case, so the comparison is made per incident (Section 3.6). The generated incidents reproduce the target NIBSS channel proportions, with a maximum deviation of 0.04 percentage points. Since the channel quotas are explicitly imposed during generation, this agreement verifies the implementation of the calibration procedure rather than providing independent evidence of realism. The same distinction applies to the monthly profile of incident starts (maximum deviation 1.0 percentage point) and the technique mix (maximum deviation 1.4 points), which are generated from NIBSS-derived probabilities. The mean loss per incident is within 8% of the NIBSS mean loss per case for IB, Mobile, POS and Web (IB NGN 768k against 761k, Mobile 119.8k against 119.8k, POS 265.7k against 251.5k, Web 95.1k against 101.6k); the differences are larger for ATM (+19%, 10 incidents) and ECOM (+16%, 32 incidents), where the sample mean of a log-normal is noisy. The mean amount of a single fraud *row* is lower (about the incident loss divided by 1.63), because the loss is split across the incident's transactions. We have no independent source of Nigerian transaction data against which to test realism. Fraud-row prevalence is the one headline quantity that is deliberately *not* matched to NIBSS: our 0.1% is a design choice for supervised learning, and it cannot be compared precisely with the NIBSS case-to-transaction ratio because a case can contain several transactions (Section 3.6). For context, the ULB card dataset has 0.172% fraud and PaySim about 0.13% (Fig. F2 (Supplementary Appendix F)).

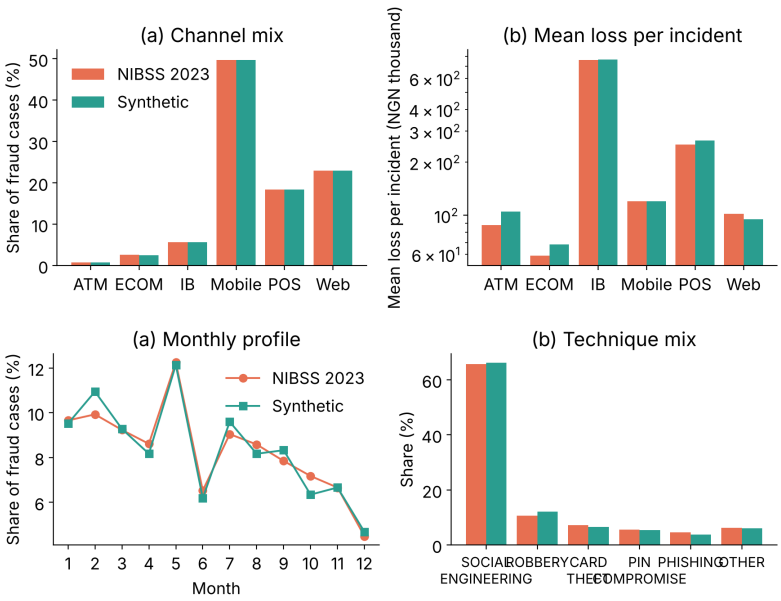


Fig. 3 FraudBench-NG versus NIBSS 2023. Top row: channel mix of fraud incidents and mean loss per incident by channel (log scale). Bottom row: monthly share of fraud cases and share by fraud technique.

4.1.2 Internal and temporal consistency.

Every behavioural feature was recomputed row by row from the strictly prior transactions (burn-in included) of 40 random customers, with a maximum relative error of 9.3×10^{-6} (rounding); the velocity score is not a copy of the amount ratio (Spearman correlation 0.001); only 0.002% of rows have no prior transaction; and a label-permutation test gives ROC-AUC 0.499 (chance) for $R_{i,k}$. The full list is in Table F10 (Supplementary Appendix F). The realism checks (Figs. F3 and F4, Supplementary Appendix F) show a diurnal cycle (6.7% of transactions between 00:00 and 05:59 against 25% under a uniform clock), customer heterogeneity (customer identity explains $\eta^2 = 0.46$ of the variance of log amounts), burstiness (inter-arrival CV = 1.78 against 1 for a Poisson process) and heavy-tailed amounts. These patterns are assumed into the generator and shown to be present in its output; they are not shown to match Nigerian data. Temporal consistency has one residual weakness: the burn-in reduces but does not remove the growth of customer history, so the median number of prior transactions is 56 in the training period and 125 in the test period (Section 6).

4.1.3 Answer to RQ1.

FraudBench-NG reproduces the NIBSS 2023 channel, monthly, technique and mean-loss aggregates to within the tolerances above and passes the internal and temporal tests, with three qualifications: agreement is by construction; the NIBSS tables are themselves mutually inconsistent (Supplementary Appendix D); and prevalence,

customer behaviour and merchant effects are assumptions, not reproductions. The calibration targets are *incident* counts and losses, matching how NIBSS reports them.

4.2 RQ2: causal features and absence of leakage

We tested the three ways in which a synthetic benchmark typically leaks. *Future or current-transaction information in a feature:* the brute-force recomputation of Section 4.1 reproduces every behavioural feature from the strictly prior transactions alone (maximum relative error 9.3×10^{-6}), and no customer carries an aggregate that is constant across all of that customer's rows (a whole-year aggregate would be); the label-permutation test gives a ROC-AUC of 0.499 for the relative amount $R_{i,k}$. *Redundant features:* the velocity score equals the relative amount in only 0.03% of rows and the Spearman correlation between them is 0.00, no composite score is published, and customer identity explains 46% of the variance of log amounts, so the amount is not simply re-measured by a second feature. *The generator's own assumption as a feature:* the merchant risk score is a rescaling of the factors used to select fraud, so merchant category and score are excluded from the default model features and tested separately (Section 4.3).

One further check separates designed signal from leakage. A random forest on the chronological test period reaches a PR-AUC of 0.016 (24 times its base rate of 0.0007) and 0.006 (9 times) when all customer-history features are removed. The history features therefore carry most of the signal, but, because their causality is verified directly, this is the designed takeover-burst effect and not leakage; the ablation does not say which history feature matters, because the 24-hour and 7-day windows are in the same group. Further tests, including sparse windows and the share of rows without any prior transaction, are in Supplementary Appendix F, Table F10.

4.2.1 Answer to RQ2.

Yes for the leakage routes tested: whole-year or future aggregates, the current transaction inside baselines and windows, duplicated or composite scores, and the generator's own merchant assumption as a model input. We cannot show that no other leakage exists; the evidence is the brute-force causality test, the label-permutation test and the redundancy checks, and a low benchmark score alone would not be evidence.

4.3 RQ3: stability across seeds and assumptions

4.3.1 Seeds.

Across ten independent seeds (8,000 customers each) the realised prevalence is 0.100%; the maximum deviation of the incident channel mix from NIBSS is 0.11 ± 0.01 percentage points; the inter-arrival CV is 1.78 ± 0.01; the follow-on share of fraud rows is 0.39 ± 0.02; and the single-feature ROC-AUC of the velocity score is 0.63 ± 0.02 and of the amount 0.48 ± 0.02 (Table F11 (Supplementary Appendix F)). The mean loss of internet-banking incidents varies widely between seeds (NGN 714k±181k, against 761k in NIBSS) because it is the mean of about 35 heavy-tailed draws. The structure of the data does not depend on a particular random draw. Model metrics are much

noisier: with about 90 test fraud rows per seed, the random-forest PR-AUC varies by about a third of its mean between seeds.

4.3.2 Merchant assumption.

The merchant factors are the one assumed quantity that was published as a feature (merchant risk score, Section 3.8), and a model that uses it is fed the generator's own assumption. We therefore *exclude both the score and the merchant category from the default model features*. Adding them back (`compare.py --with-merchant`) changes the test PR-AUC of logistic regression, random forest and XGBoost from 0.036, 0.016 and 0.017 to 0.036, 0.016 and 0.012, and recall at 1% FPR from 0.217, 0.231 and 0.195 to 0.217, 0.235 and 0.206 (single run, intervals overlap fully). We also regenerated the data with the assumed factors, with *flat* factors (all 1) and with *reversed* factors, ten seeds each (Table F12 (Supplementary Appendix F)). The manipulation works (the share of fraud in the three assumed-riskiest categories moves from 30.2% to 23.4% (flat) to 15.8% (reversed)), but the random-forest PR-AUC is 0.023 ± 0.017 (assumed), 0.032 ± 0.026 (flat) and 0.015 ± 0.012 (reversed) across seeds; the differences are smaller than the seed-to-seed spread, and adding the merchant columns to the features changes the PR-AUC by at most 0.002 in every setting. We cannot detect an effect of this size with ten seeds, and the default model results do not use the merchant columns. Equally, the data cannot be used to learn about merchant risk, because that effect is assumed.

4.3.3 Customer-population assumptions.

Age group, location and bank are inert in the generator (Section 3.2), so varying their shares cannot change any model result; the one population assumption that can matter is the legitimate channel mix. We regenerated the data with the assumed mix (mobile 42%), with a mobile-heavy mix (mobile 60%, ATM 3%, e-commerce and internet banking 7% each, POS 12%, web 11%) and with the NIBSS fraud channel mix used as the legitimate mix, which removes any difference between the channel mix of legitimate and fraud rows, ten seeds each. The random-forest PR-AUC is 0.023 ± 0.017, 0.024 ± 0.020 and 0.020 ± 0.024, and recall at a 1% false-positive rate 21%, 21% and 20%, so overall performance is not detectably sensitive to the mix. The share of the forest's feature importance that falls on the channel columns does change: it is 7.0 ± 0.3% with the assumed mix and 4.4 ± 0.2% when the legitimate mix equals the fraud mix, which supports the reading of Section 4.6 that the channel features of the SHAP analysis come from combining NIBSS fraud marginals with assumed legitimate ones.

4.3.4 Fraud-process assumptions.

We varied the takeover continuation probability ϕ (0, 0.4, 0.7) and the incident-loss dispersion s (0.5, 0.9, 1.3), ten seeds each (Table F13 (Supplementary Appendix F)). These matter far more than the merchant factors. The takeover parameter ϕ drives the velocity signal and the overall score: with no follow-on transactions the velocity score's ROC-AUC is 0.50 and the random-forest PR-AUC is 0.041 ± 0.019; at the assumed

$\phi = 0.4$ it is 0.023 ± 0.017 (pooled over seeds 0.014); and at $\phi = 0.7$ the velocity ROC-AUC is 0.76 and PR-AUC rises to 0.141 ± 0.082 (pooled 0.096; recall at 1% FPR 0.42). The gain comes from follow-ons: their PR-AUC is 0.190 ± 0.106 at $\phi = 0.7$, while the PR-AUC for *first* events is 0.005 ± 0.004, lower than at $\phi = 0$ (0.041 ± 0.019, where every fraud row is a first event). The overall level of the benchmark thus largely reflects the assumed takeover burst. The dispersion s changes how unevenly incident losses are split into row amounts: PR-AUC is 0.020 ± 0.013 at $s = 0.5$ and 0.068 ± 0.036 at $s = 1.3$. Run-to-run spread is large (a single run has about 90 test fraud rows), so we also report PR-AUC pooled over the ten seeds; none of the pairwise differences among $\phi = 0$, 0.4 and the s settings is established, but the large effect of $\phi = 0.7$ is.

4.3.5 Why is performance higher at $\phi = 0$ than at $\phi = 0.4$?

Bursts add a velocity signal, so one might expect $\phi = 0.4$ to be easier than $\phi = 0$, yet the PR-AUC lift over the base rate is 59 ± 27 at $\phi = 0$ and 36 ± 25 at $\phi = 0.4$ (mean $\pm$ s.d. over ten seeds). This difference lies within the seed-to-seed spread, so it is not established by the PR-AUC alone, but the mechanism behind it is clear and we tested it (Figure 4). Three explanations are possible. *(i) Splitting of the incident loss.* At $\phi = 0$ every fraud row carries the whole incident loss, which is large relative to the customer's usual amount. At $\phi > 0$ the loss is divided among the incident's rows (Section 3.6), so the first row is smaller: the mean amount of a first fraud event is NGN 171k at $\phi = 0$, 129k at 0.4 and 88k at 0.7, and the ROC-AUC of the relative amount $R_{i,k}$ for first events falls from 0.67 ± 0.02 to 0.55 ± 0.02 and 0.41 ± 0.02, i.e. below chance at $\phi = 0.7$, where a first row is on average smaller than a typical legitimate transaction. *(ii) Fewer independent incidents.* At a fixed fraud-row prevalence the number of incidents is $F(1 - \phi)$, so $\phi = 0.4$ has 60% as many incidents as $\phi = 0$ (401 against 668 per 8,000 customers). We therefore raised the prevalence at $\phi = 0.4$ until the number of incidents matched that at $\phi = 0$; the lift is then 32 ± 11 and the ROC-AUC of $R_{i,k}$ is still 0.55, so incident count does not explain the gap. *(iii) The burst itself.* We gave every row of a $\phi = 0.4$ incident the full incident loss instead of a share. The ROC-AUC of $R_{i,k}$ returns to 0.67 ± 0.02 and the lift to 64 ± 36 (first events 46 ± 41), equal to or above that at $\phi = 0$ within noise. The difference between $\phi = 0$ and $\phi = 0.4$ is therefore produced by our assumption that an incident's loss is split among its transactions, not by the bursts or by the number of incidents. This full-loss variant is an ablation only: it would multiply the total loss per case by about 1.7 and so break the NIBSS loss calibration.

4.3.6 Transfer across takeover assumptions.

A benchmark is more useful if a model trained under one assumption still works when the assumption changes. We trained the random forest at each of $\phi = 0$, 0.4 and 0.7 and tested it on data generated at each, using the same ten seeds (Table 4). A model trained at the baseline $\phi = 0.4$ keeps part of its performance: recall at a 1% false-positive rate is 18% at $\phi = 0$ and 36% at $\phi = 0.7$, against 24% and 42% for models trained at those values (74% and 86% of the matched recall), but its PR-AUC is 0.015 at $\phi = 0$ (a third of the matched 0.041) and 0.058 at $\phi = 0.7$ (0.141 matched). The transfer is asymmetric. A model trained without bursts ($\phi = 0$) fails on bursty data

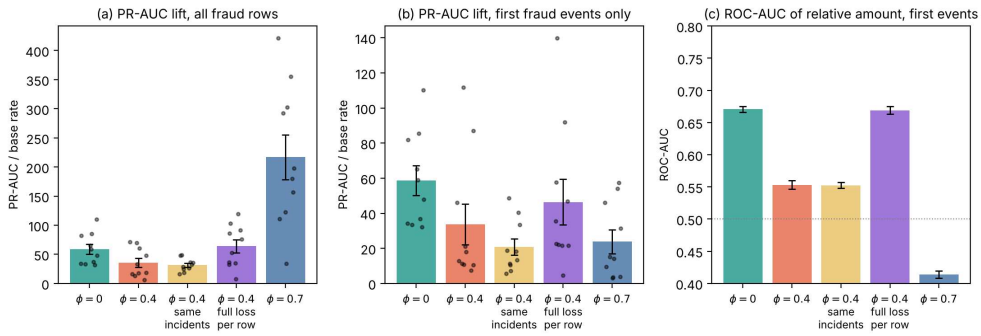


Fig. 4 Why performance depends on the takeover probability ϕ: random forest, 10 seeds of 8,000 customers. (a) PR-AUC lift (PR-AUC divided by the base rate) for all fraud rows and (b) for first fraud events only; dots are seeds, bars show the mean and standard error. (c) ROC-AUC of the relative amount for first fraud events. "Same no. of incidents": prevalence raised so that $\phi = 0.4$ has as many incidents as $\phi = 0$. "Full loss per row": every row of an incident carries the full incident loss (ablation only).

Table 4 Cross-assumption transfer of the random forest. Rows: takeover probability ϕ of the training data; columns: ϕ of the test data. Each cell gives the PR-AUC (mean ± s.d. over ten seeds of 8,000 customers) and, in brackets, the recall at a 1% false-positive rate (mean). Diagonal cells are models tested on their own assumption. The no-skill PR-AUC is 0.00064–0.00071.

Trained at	Tested at $\phi = 0$	Tested at $\phi = 0.4$	Tested at $\phi = 0.7$
$\phi = 0$	$\mathbf{0.041 \pm 0.019}$ (24%)	0.016 ± 0.015 (10%)	0.005 ± 0.007 (5%)
$\phi = 0.4$	0.015 ± 0.009 (18%)	$\mathbf{0.023 \pm 0.017}$ (21%)	0.058 ± 0.039 (36%)
$\phi = 0.7$	0.002 ± 0.001 (8%)	0.027 ± 0.020 (21%)	$\mathbf{0.141 \pm 0.082}$ (42%)

(recall 5%, PR-AUC 0.005 at $\phi = 0.7$), because it never learned to use the velocity features. A model trained at $\phi = 0.7$ is close to chance on burst-free data (recall 8%, PR-AUC 0.002 against a no-skill 0.0007), because it relies on velocity features that carry no signal there. Reported performance is therefore conditional on both the training and the test assumption. As elsewhere, these are properties of the benchmark and its assumptions; they do not show how a model would transfer between kinds of real fraud, and with about 900 fraud rows per setting over ten seeds the intervals are wide (Table 4).

4.3.7 Answer to RQ3 (stability).

Distributional properties are stable across ten seeds. Benchmark metrics are stable in order of magnitude but not in their second digit, and the three model families cannot be ranked with confidence (Table 5). Absolute performance is conditional on unmeasured assumptions: strongly on the takeover continuation probability and on how an incident's loss is split across its transactions, moderately on the loss dispersion, and not detectably on the merchant assumption or the legitimate channel mix. A

Table 5 Benchmark results on the FraudBench-NG test period (95% customer-clustered bootstrap intervals in brackets). The no-skill PR-AUC equals the test prevalence, 0.00066.

Model	PR-AUC	Recall @1% FPR	ROC-AUC	Recall @0.1% FPR	Prec. top 1%	F1
Logistic regression	0.036 [0.018, 0.068]	0.217 [0.165, 0.270]	0.774	0.076	0.014	0.086
Random forest	0.016 [0.009, 0.029]	0.231 [0.165, 0.289]	0.748	0.087	0.015	0.075
XGBoost	0.017 [0.008, 0.039]	0.195 [0.145, 0.254]	0.732	0.083	0.013	0.061

model trained under one takeover assumption transfers only partly to another, and asymmetrically (Table 4).

4.4 RQ3: downstream performance

We compare logistic regression, random forest and XGBoost on a chronological split (Table 5). Models are trained on all training fraud and a random sample of 300,000 legitimate transactions, with class weighting (logistic regression, random forest) or scale_pos_weight = 50 (XGBoost). **The default feature set excludes** identifiers, labels, incident identifier, follow-on indicator, merchant category and merchant risk score: the last two are the generator's own assumption and would let a model learn that assumption. Because fraud is rare, the operating threshold is chosen on the validation set and the main metrics are threshold-free: the precision–recall area under the curve (PR-AUC; preferred over ROC-AUC for rare classes [16, 17]), recall at fixed false-positive rate (FPR), precision among the top 1% of scores, and F1. Several fraud rows belong to one incident and one customer, so a row-level bootstrap would give intervals that are too narrow. Intervals are therefore 95% percentile *customer-clustered* bootstrap intervals (200 replicates with multinomial customer weights [18]); for recall, which is fraud-specific, we also report incident-clustered intervals.

Model settings are given in Table 6. The feature set has 48 columns after one-hot encoding: 20 numeric behavioural, amount and time features and the one-hot columns of channel, bank, location and age group. Bank, location and age group are inert by construction, so they can only add noise.

All three models detect fraud above chance (PR-AUC 0.016–0.036 against a no-skill 0.00066, i.e. 24–55 times the base rate; ROC-AUC 0.73–0.77), but the task is hard: at a 1% false-positive rate 20–23% of fraud is caught, and about 1.4% of the top 1% of scores are fraud. The intervals are wide and overlap completely, so *the three models cannot be ranked*; incident-clustered recall intervals (for example $[0.17, 0.30]$ for the random forest) are similar. The low absolute level is expected: with no leaking feature, the only signal is the documented takeover-burst mechanism and the NIBSS marginals. Figure 5 shows why the task is hard: legitimate and fraudulent amounts overlap heavily (Kolmogorov–Smirnov distance 0.11), and the behavioural features separate the classes only weakly. No single feature exceeds a ROC-AUC of 0.64 (Fig. F1

Table 6 Settings of the three benchmark models (scikit-learn 1.9.1, XGBoost 3.4.1). All models are trained on the same sample: every training-period fraud row (1,374) and 300,000 randomly drawn legitimate training rows, random seed 0. Parameters not listed are library defaults. The decision threshold for precision and F1 is chosen on the validation period to maximise F1; PR-AUC, ROC-AUC and recall at fixed false-positive rates do not depend on it.

Model	Settings
Logistic regression	L2 penalty, inverse regularisation strength $C = 1.0$, class weight "balanced", maximum 2,000 iterations, features standardised (zero mean, unit variance, fitted on the training sample)
Random forest	300 trees, minimum 5 samples per leaf, class weight "balanced_subsample", other settings default (square-root feature subsampling, no depth limit), random state 0
XGBoost	300 trees, maximum depth 6, learning rate 0.05, row subsample 0.8, column subsample 0.8, positive-class weight 50, evaluation metric area under the precision–recall curve, random state 0

(Supplementary Appendix F)); precision–recall curves and recall at fixed false-positive rates are in Fig. F5 (Supplementary Appendix F). The informative features are those tied to mechanisms that the generator contains (takeover bursts), so the signal comes from documented assumptions. We do not claim these numbers transfer to real data.

4.4.1 First fraud events and follow-ons.

The test fraud rows are of two kinds: 172 *first events* (the first transaction of an incident) and 105 *follow-ons*, whose velocity features are shaped by the takeover burst. We therefore score the models on each subset against all legitimate test rows (Table 7). Base rates differ (0.00041 and 0.00025), so we compare lifts over the base rate. First events are the realistic detection problem, because they come before any fraud history exists in the account; their PR-AUC lift is 32–111 times the base rate, ROC-AUC 0.72–0.74 and recall at 1% FPR 18–19%. Follow-ons are somewhat easier in ROC-AUC (0.71–0.84) and recall (22–30% at 1% FPR), but their PR-AUC lift is lower, 20–24 times the base rate; with 172 and 105 positives, none of these differences between subsets or models is established. **Overall benchmark difficulty largely reflects the takeover-burst assumption**: the share of follow-ons and their separability are set by ϕ (Section 4.3), so users who want a task dominated by first events should report that subset.

Table 7 Test performance by event type: first fraud events ($n = 172$, base rate 0.00041) and follow-ons ($n = 105$, base rate 0.00025). Lift is PR-AUC divided by the subset's base rate.

Subset	Model	PR-AUC (lift)	ROC-AUC	Recall@1%FPR
First events	Logistic regression	0.046 (111)	0.73	0.19
	Random forest	0.013 (32)	0.72	0.19
	XGBoost	0.014 (34)	0.74	0.18
Follow-ons	Logistic regression	0.005 (20)	0.84	0.26
	Random forest	0.005 (21)	0.80	0.30
	XGBoost	0.006 (24)	0.71	0.22

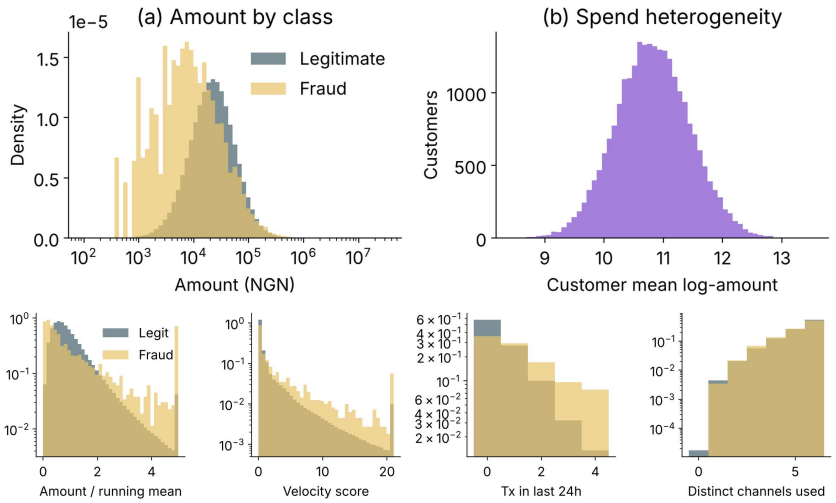


Fig. 5 Distribution overlap in FraudBench-NG. Top: legitimate versus fraudulent amounts (density, log axis) and customers' mean log amount. Bottom: four behavioural features by class (log density).

4.5 RQ3: what the results mean for a bank

In the test period (420,154 legitimate transactions) the 1% operating point raises 4,202 false alerts to catch 54–64 of the 277 fraud rows, that is 66–78 false alerts for each fraud row caught and a precision of 1.3–1.5% (Table 8). At a 0.1% false-positive rate the burden falls to 18–20 false alerts per fraud row caught, but only 8–11% of incidents are flagged at their first transaction and 9–12% at any transaction. The prevalence of 0.1% is itself enriched (Section 3.6), so at a lower prevalence the number of false alerts per fraud caught would be higher still.

The timing of detection matters as much as its rate. Here an incident is the unit a bank would want to stop, and its first transaction is the earliest chance to do so. At the 1% operating point only 18–19% of the 172 test incidents are flagged at their first transaction, a further 6–9% only at a later transaction of the same incident, and 72–76% are never flagged. Among the 64 incidents with more than one transaction, 17–19% are flagged at the first transaction and 34–44% at some transaction, so the follow-on bursts give the models a second chance but not a reliable one. Even if every alert blocked the account at once, only 7–8% of the fraud amount occurs after an incident's first alert, because most of the loss lies in incidents that are never flagged. These figures describe the benchmark and its assumed takeover bursts, not real Nigerian fraud, and with 172 incidents they are imprecise (Table 8 reports single-run proportions without intervals).

4.6 RQ3: explainability

We computed TreeSHAP values for the XGBoost model of Table 5 itself (settings of Table 6, the same training sample of 1,374 fraud and 300,000 legitimate rows, merchant columns excluded; test PR-AUC 0.017), on a sample made of all test fraud

Table 8 Alert burden and detection timing on the FraudBench-NG test period at a 1% false-positive rate (threshold set on the legitimate test rows; 420,154 legitimate rows, 277 fraud rows in 172 incidents; 64 incidents have more than one transaction). Incidents are flagged at their first transaction, only at a later transaction, or never.

	Logistic regression	Random forest	XGBoost
Fraud rows flagged (of 277)	60	64	54
False alerts per fraud row caught	70	66	78
Precision of alerts	1.4%	1.5%	1.3%
Incidents flagged at their first transaction	19.2%	18.6%	18.0%
Incidents flagged only at a later transaction	7.0%	9.3%	6.4%
Incidents never flagged	73.8%	72.1%	75.6%
Multi-transaction incidents flagged at first transaction (64)	17.2%	18.8%	17.2%
Multi-transaction incidents flagged at any transaction (64)	35.9%	43.8%	34.4%
Share of fraud amount after the first alert	8.0%	7.8%	6.9%

and 4,000 legitimate test rows, then summed mean absolute SHAP values within four feature groups. Customer history carries 57% of attribution, with the relative amount $R_{i,k}$, the number of prior transactions and the running mean amount the top three features (Figure 6); channel, bank, location and age together carry 26%, time 14% and the amount itself only 4%. Because merchant columns are excluded, the generator's merchant assumption no longer appears among the top features. *Why e-commerce and the month appear.* Two of the top features reflect how the data were built, not fraud behaviour. The fraud channel mix follows NIBSS (e-commerce 3% and internet banking 5% of fraud), whereas the legitimate channel mix is our assumption (11% each), so fraud rates per 1,000 transactions are low in e-commerce (0.28), internet banking (0.47) and ATM (0.20) and high on mobile, POS and web (1.15–1.39); the e-commerce indicator therefore marks *lower* risk. Likewise the monthly fraud profile is imposed from NIBSS while legitimate activity is flat over the year, so the fraud rate per 1,000 transactions runs from 0.48 in December to 1.47 in February, and the month encoding carries signal by construction. Mean absolute SHAP values show the size of a feature's influence, not its direction. Both effects arise from combining NIBSS fraud marginals with assumed legitimate ones, and they do not show that channel or calendar month predicts fraud in Nigeria. Two cautions apply. First, prior transaction count and the running mean and standard deviation rank high, and these cumulative features grow over the year (Section 6), so part of their attribution may reflect residual history drift, although prior transaction count has no predictive value alone; users should repeat the check by dropping them. Second, bank, location and age are inert by design, so their group share is mostly channel (e-commerce is the top channel feature; see above) plus noise from many one-hot columns. The attributions describe the documented mechanisms of the generator; they do not show that the explanations match real fraud.

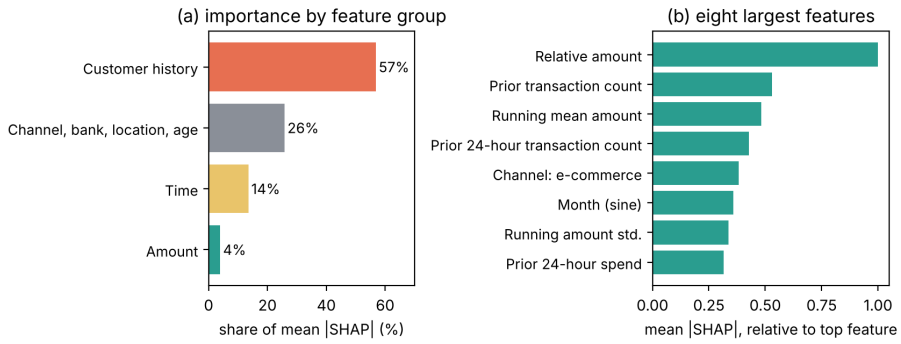


Fig. 6 TreeSHAP attribution of XGBoost trained on FraudBench-NG: share of attribution by feature group, and the eight largest features.

5 Discussion

5.0.1 Answers to the research questions.

RQ1: FraudBench-NG reproduces the selected NIBSS 2023 aggregates and is internally and temporally consistent, but agreement is by construction and prevalence, customer behaviour and merchant effects are assumed. *RQ2:* causal generation avoids the leakage routes tested (future or current-transaction information, redundant scores, the generator's own merchant assumption as an input), verified by brute-force recomputation and a label-permutation test; absence of other leakage is not proved. *RQ3:* the structure of the data is stable across seeds and, within the tested range, across the merchant assumption, while absolute benchmark performance depends on the assumed takeover continuation probability, the way incident losses are split across transactions and the fraud-amount dispersion; models trained under one burst probability transfer only partly to another; benchmark metrics are low (PR-AUC 0.016–0.036), the models cannot be ranked, at a 1% false-positive rate about 70 false alerts are raised per fraud row caught and only about 19% of incidents are flagged at their first transaction, and explanations follow the documented mechanisms.

5.0.2 What the dataset is for.

It is a benchmark with documented, adjustable assumptions, suitable for comparing models under identical conditions, studying feature construction, and testing human-in-the-loop workflows at a prevalence of 0.1%. It is not suitable for estimating the performance a model would have on real Nigerian transactions, for fairness analysis across age, location or bank, or for learning which merchants or customers are risky.

6 Limitations

1. *Calibration data.* The profiles are from 2023; NIBSS has since reported 70,111 (2024) and 67,518 (2025) cases but no channel, month or technique tables (Section 3.1). The 2023 tables are mutually inconsistent (about 6% of reported loss

is unattributed), cover reported fraud only, and give actual rather than attempted losses. Only marginal distributions are known, so every joint distribution (channel by month, technique by channel, anything by age, location or bank) is assumed independent.

2. *Prevalence is enriched by design.* The 0.1% fraud-row prevalence is a design choice and cannot be stated as a multiple of the NIBSS case-to-transaction ratio (Section 3.6); precision-based metrics would be lower at a lower prevalence.
3. *Fraud-process assumptions.* Takeover bursts are a one-parameter guess ($\phi = 0.4$, delay mean five minutes), incident losses ignore the victim and are split by Dirichlet(2) proportions, technique is independent of channel, and the nine smallest techniques are pooled, which hides the cost of internal collusion and server hacking. Section 4.3 shows that these choices drive benchmark difficulty.
4. *Merchant effect.* The factors are judgements (Table E8 (Supplementary Appendix E)); the published risk score is their rescaling and is not evidence of merchant risk, so the default models do not use it.
5. *Structure not modelled.* Table B2 (Supplementary Appendix B) lists, with the consequence for users, what the generator does not reproduce, for example merchant preferences, age, location and bank effects, device or geographic data, and long-term customer behaviour.
6. *Many parameters are assumptions* (Table A1 (Supplementary Appendix A)), and there is no independent validation against real transactions.
7. *Residual history drift.* The median number of prior transactions is 56 in the training period and 125 in the test period, so cumulative features such as prior transaction count still drift; time-split studies should consider excluding it or lengthening the burn-in.
8. *Small test sets.* The test period has 277 fraud rows (172 incidents), so benchmark intervals are wide and models cannot be ranked; larger customer counts or pooled seeds reduce the width.

7 Conclusion

We present FraudBench-NG, a parametric simulation calibrated to 2023 NIBSS aggregates, with a released dataset of 2.1 million transactions from 25,000 customers at 0.1% fraud. Fraud is generated as incidents with one loss per case, every behavioural feature uses only prior transactions, and every parameter is labelled as NIBSS or assumed. The generator reproduces the NIBSS channel, monthly, technique and per-case loss aggregates (by construction), passes internal and temporal consistency tests, and avoids the leakage routes we tested. Benchmark performance is modest (PR-AUC 0.016–0.036 against a no-skill 0.00066), the three model families cannot be ranked, and at a 1% false-positive rate the models raise about 70 false alerts per fraud row caught and flag only about 19% of incidents at their first transaction. Difficulty depends largely on assumptions: the takeover-burst probability, and, as an ablation shows, the rule that splits an incident's loss across its transactions. A model trained at one burst probability transfers only partly to another. Results should therefore be reported for first

fraud events and follow-ons separately, and at more than one value of the generator's assumptions.

FraudBench-NG is a transparent, auditable and assumption-aware benchmark. It is not a substitute for real transaction data, and we make no claim that it is realistic in every detail. Future work should calibrate the assumed timing and amount parameters when anonymised bank data become available, add device and location variables, and release a variant at the real NIBSS prevalence for evaluation.

Declarations. **Funding:** none. **Competing interests:** none declared. **Ethics:** the dataset is synthetic and contains no personal data. **Data availability:** the generator (`generate_dataset.py`), the validation, leakage-check, sensitivity and figure scripts, the figures and the result files are available at https://github.com/otujacob/FraudBench-NG; running `python generate_dataset.py --customers 25000 --prevalence 0.001 --seed 42` reproduces the dataset used in this paper (about 2.1 million rows). The dataset (Parquet) will be deposited in a public repository with a DOI before publication, and can be regenerated exactly from the generator with seed 42. **Supplementary material:** Supplementary Appendices A–F (parameter register, items not modelled, variable reference, NIBSS calibration data, assumed weights, additional validation) are provided as a separate file. **Author contributions:** O.J.: conceptualisation, software, validation, writing; C.K. and A.K.: supervision, review and editing; C.E.O.: review and editing.

References

[1] Nigeria Inter-Bank Settlement System (NIBSS): Annual fraud landscape: January–december 2023. Technical report, NIBSS, Abuja, Nigeria (Apr 2024). https://nibss-plc.com.ng/wp-content/uploads/2024/04/2023-Annual-Fraud-Landscape.pdf

[2] Ojoko, I.: Nigeria's financial sector suffers NGN 52.26 billion loss to fraud in 2024 – NIBSS report. Nairametrics, 26 February 2025. Accessed 9 October 2026 (2025). https://nairametrics.com/2025/02/26/nigerias-financial-sector-suffers-n52-26-billion-loss-to-fraud-in-2024-nibss-report/

[3] Nigeria Inter-Bank Settlement System (NIBSS): Digital payment fraud drops 51% to NGN 25.85b, Lagos accounts for 63%. NIBSS press release, 22 January 2026. Accessed 9 October 2026 (2026). https://nibss-plc.com.ng/digital-payment-fraud-drops-51-to-n25-85b-lagos-accounts-for-63/

[4] Dal Pozzolo, A., Caelen, O., Johnson, R.A., Bontempi, G.: Calibrating probability with undersampling for unbalanced classification. In: Proc. 2015 IEEE Symp. Series Comput. Intell. (SSCI), Cape Town, South Africa, pp. 159–166 (2015). https://doi.org/10.1109/SSCI.2015.33

[5] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G.: Credit card fraud detection: A realistic modeling and a novel learning strategy. IEEE

Trans. Neural Netw. Learn. Syst. **29**(8), 3784–3797 (2018) https://doi.org/10.1109/TNNLS.2017.2736643

[6] Vesta Corporation: IEEE-CIS fraud detection competition dataset. Kaggle. https://www.kaggle.com/c/ieee-fraud-detection (2019)

[7] Lopez-Rojas, E., Elmir, A., Axelsson, S.: PaySim: A financial mobile money simulator for fraud detection. In: Proceedings of the 28th European Modeling and Simulation Symposium (EMSS), Larnaca, Cyprus, pp. 249–255 (2016)

[8] Lopez-Rojas, E.A., Axelsson, S.: BankSim: A bank payments simulator for fraud detection research. In: Proceedings of the 26th European Modeling and Simulation Symposium (EMSS), Bordeaux, France, pp. 144–152 (2014)

[9] Bahnsen, A.C., Aouada, D., Stojanovic, A., Ottersten, B.: Feature engineering strategies for credit card fraud detection. Expert Systems with Applications **51**, 134–142 (2016) https://doi.org/10.1016/j.eswa.2015.12.030

[10] Kaggle user hendurhance: nibsss-fraud-dataset. Kaggle dataset listing. Accessed 9 October 2026. https://www.kaggle.com/datasets/hendurhance/nibsss-fraud-dataset

[11] A2F Nigeria: EFInA report: Nigeria's formal financial inclusion grows to 64% in 2023. https://a2f.ng/efina-report-nigerias-formal-financial-inclusion-grows-to-64-in-2023/. Reports the EFInA Access to Finance Survey 2023: 52% of adults banked, 93% mobile-phone adoption, 37% of rural and 17% of urban adults financially excluded. Press report of the survey; accessed 16 October 2026 (2024)

[12] National Bureau of Statistics: Demographic Bulletin 2023: projected population by state, as reported by 21st Century Chronicle. https://21stcenturychronicle.com/kano-records-15-7m-lagos-13-7m-as-nigerias-population-hits-221m-in-2023-nbs/. Accessed 13 October 2026 (2023)

[13] Limpert, E., Stahel, W.A., Abbt, M.: Log-normal distributions across the sciences: keys and clues. BioScience **51**(5), 341–352 (2001) https://doi.org/10.1641/0006-3568(2001)051[0341:LNDATS]2.0.CO;2

[14] Barabási, A.-L.: The origin of bursts and heavy tails in human dynamics. Nature **435**, 207–211 (2005)

[15] Welford, B.P.: Note on a method for calculating corrected sums of squares and products. Technometrics **4**(3), 419–420 (1962)

[16] Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE

10(3), 0118432 (2015)

[17] Davis, J., Goadrich, M.: The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240 (2006)

[18] Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall, New York, NY (1993)

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryMaterials.zip