

Anomaly Detection with a Blockchain Audit Trail for Open Banking Anti-Fraud: Prototype Evidence and Research Agenda

Krzysztof Piech

0000-0001-8120-3675

Łazarski University, Warsaw, Poland

krzysztof.piech@lazarski.edu.pl

Abstract—This paper presents preliminary prototype evidence from an industrial research project on an Open Banking Anti-Fraud PSD2 system. The proposed approach combines anomaly detection based on customer-behaviour-relative features with a blockchain-based audit trail. The system was designed to address three practical challenges in PSD2/Open Banking environments: real-time fraud screening, reduction of false positives, and auditable recording of monitoring activities. The prototype integrates transaction and customer-context data from a Polish payment institution with an anomaly-detection layer and a permissioned blockchain component used to preserve tamper-evident records of analytical actions and results. More than 30 input features were developed. Prototype evidence suggests that customer-behaviour-relative features improved discrimination relative to absolute features, with AUC increasing from 0.939 to 0.996 in the reported comparison. The final anomaly-detection model supported sub-second operational scoring, while a simulated cross-institution feature-sharing procedure completed in approximately 10.3 seconds. The paper also clarifies the validation scope: the anomaly-detection results are preliminary and internally evaluated, whereas broader production-level validation remains future work. The contribution of the paper is twofold. First, it proposes an information-systems architecture linking AI-based anomaly detection, explainable scoring, and blockchain-supported auditability. Second, it formulates a research agenda for validating such systems across institutions, datasets, and regulatory settings.

Index Terms—Open Banking, fraud detection, anomaly detection, blockchain audit trail, PSD2, financial compliance

I. INTRODUCTION

THE expansion of Open Banking under the PSD2 directive has improved interoperability between financial institutions and enabled new payment and account-information services [1], [2]. At the same time, it has increased the complexity of transaction monitoring, fraud prevention, and evidence preservation in regulated financial environments. The growing role of third-party providers (TPPs) creates a need not only for real-time risk screening but also for trustworthy records of who accessed financial data, what actions were performed, and how suspicious activity was assessed.

Traditional anti-fraud systems are often based on manually defined rules and static scenarios. Such systems may be effective for known patterns, but they tend to generate excessive false positives and have limited ability to adapt to heterogeneous customer behaviour. This limitation becomes

particularly visible in Open Banking environments, where customers differ substantially in transaction frequency, transaction size, product usage, geographic profile, account history, and behavioural stability. A practically useful anti-fraud system must therefore address two problems simultaneously: it must detect anomalous behaviour quickly, and it must do so without treating all customers as if they followed the same behavioural baseline.

The project was titled (in translation): “Research and development work on an Open Banking Anti-Fraud PSD2 system based on blockchain technology and machine learning in transaction monitoring and fraud detection processes” and is referred to here as OBAF. It was conceived as an R&D effort addressing this gap through the integration of three elements: (1) Open Banking data ingestion, (2) machine-learning-based fraud screening, and (3) blockchain-supported evidence preservation. The original project rationale assumed that the blockchain layer would support immutable recording of selected analytical events and outcomes, while the machine-learning layer would evaluate customer behaviour and support anti-fraud decisions in real time. In this design, blockchain is not used to replace financial supervision, compliance processes, or fraud analytics. Instead, it functions as a tamper-evident audit-trail layer supporting automated risk assessment.

This paper presents OBAF not as a fully validated production system, but as a prototype. The empirical material comes from R&D results obtained in the development of a prototype anti-fraud system for a payment-institution setting. The paper focuses on four aspects: (1) the system architecture, (2) the rationale for anomaly detection instead of fully supervised classification, (3) the role of customer-behaviour-relative feature engineering, and (4) the use of blockchain hash anchoring for auditability. The intention is to clarify what the prototype demonstrates, what remains unvalidated, and what research steps are required before production-grade claims can be made.

The paper reports preliminary prototype evidence rather than statistically generalizable fraud-reduction effects. The anomaly-detection results should be interpreted as internal prototype evidence concerning model discrimination, feature design, operational scoring time, and audit-trail feasibility. They do not constitute proof of production-level generalization

or confirmed reduction of fraud losses in routine institutional use. This distinction is central to the positioning of the paper as a position paper and research agenda rather than as a completed deployment study.

The contribution of the paper is twofold. First, it presents a practical information-systems architecture combining AI-based anomaly detection, explainable scoring, and a blockchain audit trail in a PSD2/Open Banking setting. Second, it identifies methodological and operational issues that future research must address: external validation, temporal holdout testing, comparison with stronger baselines, cross-institution robustness, explainability governance, and the operational latency of the blockchain evidence layer.

The remainder of the paper is organized as follows. Section II positions the work in relation to Open Banking security, anomaly detection, class imbalance, explainable AI, and tamper-evident audit logs. Section III presents the OBAF system concept and audit-trail architecture. Section IV describes the data, features, and experimental setup. Section V reports prototype evidence, distinguishing feature-engineering results, algorithm comparisons, operational timing, and blockchain auditability. Section VI discusses design implications and limitations. Section VII formulates the research agenda and concludes the paper.

II. RELATED WORK AND POSITIONING

The paper is positioned at the intersection of Open Banking security, financial fraud detection, anomaly detection, explainable machine learning, and tamper-evident audit logging. It does not propose a new machine-learning algorithm. Instead, it examines how existing anomaly detection and audit-trail mechanisms can be integrated into a practical information-systems architecture for PSD2/Open Banking anti-fraud monitoring.

Open Banking under PSD2 creates new interoperability opportunities, but also increases the operational burden placed on payment institutions and third-party providers. Compliance is no longer limited to access control or transaction authorization; it also includes the ability to monitor suspicious behaviour, document analytical actions, and reconstruct the basis on which fraud-related decisions were made [1], [2]. In this setting, fraud detection is not only a predictive task, but also a governance and evidence-preservation problem.

Financial fraud detection has long been studied as an application area for data mining and machine learning [3]. However, financial fraud differs from many standard classification problems because confirmed fraud cases are rare, the cost of false negatives is asymmetric, and behavioural patterns differ strongly across customers. This creates severe class imbalance, a well-known problem in machine learning [5]. In credit-card and payment-fraud settings, prior studies have also emphasized the practical difficulty of building models that are both accurate and operationally usable under changing customer behaviour and limited labelled fraud data [4].

These characteristics motivate anomaly-detection approaches. In anomaly detection, the objective is not necessarily

to learn a stable binary fraud/non-fraud boundary from labelled examples, but to identify observations or entities that deviate from an expected pattern of normal behaviour [6]. This framing is particularly relevant when labelled fraud examples are scarce, incomplete, or absent in some customer cohorts. The OBAF prototype therefore evaluates unsupervised anomaly-detection methods while using confirmed fraud labels only ex post to assess discrimination.

The three anomaly-detection methods considered in the prototype represent different modelling principles. Isolation Forest isolates anomalies through random partitioning and is attractive in operational settings because of its computational efficiency [7]. One-Class SVM estimates the support of a high-dimensional distribution and is a classical novelty-detection method [8]. Local Outlier Factor detects local density deviations and is useful when outliers are defined relative to neighbourhood structure rather than to a global distribution [9]. The comparison of these methods is relevant not because the paper introduces new algorithms, but because the industrial prototype required a model that combined discrimination quality, speed, explainability, and operational coverage across customer cohorts.

A separate issue is explainability. In regulated financial environments, anomaly scores that may influence escalation, transaction review, or temporary holding of funds cannot be treated as opaque numerical outputs only. They need to be interpretable by compliance or operational teams. Model-agnostic feature attribution methods such as SHAP (SHapley Additive exPlanations) provide one way to explain model outputs by assigning to each input feature an estimated contribution to a particular prediction or anomaly score [10]. This is especially important when fraud-screening systems move from static rule execution toward behavioural models. More broadly, recent work on regulated artificial intelligence stresses the need to monitor and evidence the behaviour of AI modules over time [11].

The auditability problem has a longer history in security research. Tamper-evident audit logs were proposed as a way to support forensic reconstruction after security incidents [12]. Permissioned blockchain systems later extended this idea by enabling append-only, tamper-resistant records across organizations that may not fully trust one another [13]. In such architectures, blockchain does not need to store the full business object or personal data. A hybrid off-chain/on-chain design may keep detailed records in a controlled database while anchoring cryptographic hashes on-chain, thereby supporting later verification without exposing sensitive data.

Against this background, the gap addressed by the present paper is not the absence of anomaly-detection algorithms or blockchain logging mechanisms as such. The gap lies in their integration into an Open Banking anti-fraud architecture. The OBAF prototype combines customer-behaviour-relative feature engineering, unsupervised anomaly detection, explainable scoring, and blockchain-based hash anchoring of selected analytical records. This integration is the main object of the paper. The following sections therefore focus on architecture,

experimental setup, preliminary prototype evidence, and the research agenda required for broader validation.

III. OBAF SYSTEM CONCEPT AND AUDIT-TRAIL ARCHITECTURE

The OBAF prototype was designed as an integrated information system rather than as a standalone machine-learning model. Its purpose was to combine Open Banking data ingestion, fraud-oriented anomaly scoring, and tamper-evident recording of selected analytical actions. Figure 1 summarizes the conceptual architecture.

The first layer of the system acquires transaction-related and contextual data from the Open Banking environment. These data include information about clients, accounts, and account transactions. The system is designed around the client as the main unit of analysis: fraud risk is treated as a property of the broader account situation and behavioural context, rather than as a property of a single isolated transaction. This design choice follows from the industrial project evidence, where suspicious activity often became visible only after aggregating multiple behavioural signals.

The second layer transforms raw transactional and customer-context data into analytical features. A central design principle was to avoid relying only on fixed global thresholds. Instead, the prototype introduced customer-behaviour-relative features, defined in relation to a client's own historical activity and behavioural baseline. This makes it possible to distinguish between behaviour that is unusual in absolute terms and behaviour that is unusual for a particular customer. The same transaction amount or account activity may be ordinary for one client and anomalous for another.

The third layer performs anomaly scoring. The prototype evaluated several methods, including Isolation Forest, One-Class SVM, and Local Outlier Factor, and compared them with a supervised Random Forest baseline. The final architectural preference was for an anomaly-detection approach because it can operate across customer cohorts where confirmed fraud labels are rare or absent. This does not mean that supervised learning was ineffective in the prototype; rather, it means that supervised learning did not solve the operational coverage problem for all client cohorts. The detailed modelling setup and validation status are discussed in Section IV.

The fourth layer supports compliance and operational decision making. The anomaly score is not intended to replace human review or institutional compliance procedures. Instead, it can support escalation, additional verification, temporary holding, or review of suspicious cases. In this sense, the prototype follows a decision-support logic: it provides risk signals and explanations that may assist compliance teams, but it does not by itself determine the final legal or operational treatment of a transaction.

The blockchain layer is designed as an audit-trail mechanism. Detailed analytical records remain off-chain in a controlled relational store. A cryptographic hash of each relevant scoring record is then anchored on a permissioned blockchain. In the project design, the on-chain component does not store

personal data or full transaction details. It stores only a verification anchor that can later be used to check whether a given off-chain analytical record has been altered. The blockchain therefore does not improve the predictive power of the model directly. Its function is to support integrity, non-repudiation, and later reconstruction of monitoring activity.

This hybrid off-chain/on-chain design is important for regulated financial environments. Storing full customer or transaction data on-chain would create privacy, data-protection, and operational governance problems. Hash anchoring avoids this problem by separating business data from evidentiary commitments. The full record remains under institutional control, while the blockchain preserves a tamper-evident trace that a given analytical action and result existed at a given point in time.

The resulting architecture should therefore be understood as a combined analytics-and-evidence system. Its value does not come from anomaly detection alone, nor from blockchain alone. It comes from the integration of customer-behaviour-relative feature engineering, fast anomaly scoring, explainability, and tamper-evident evidence preservation in a single Open Banking anti-fraud workflow.

IV. DATA, FEATURES AND EXPERIMENTAL SETUP

The prototype was developed and evaluated using data from Igoria Trade S.A., a Polish payment institution listed on the NewConnect Polish alternative market. The available data covered clients, their accounts, and account transactions. Fraud labels were derived from confirmed fraud cases recorded by the institution. Confirmed fraud was rare, with an estimated prevalence of approximately 0.8% of clients in the modelled sample. The absolute scale of the dataset—number of clients, number of transactions, number of confirmed fraud cases, and observation period—is not disclosed in this version, as these figures constitute trade secrets of the payment institution.

The unit of analysis in the prototype was the client rather than the individual transaction. This modelling choice follows from the structure of the industrial problem. In the analysed setting, fraud was treated as a property of the broader account situation and customer behaviour, rather than as a property of a single isolated transaction. A single transaction may appear ordinary when considered in isolation, while the sequence of account events, inflows, outflows, customer history, and transactional concentration may reveal anomalous behaviour. The model therefore assigns an anomaly score at the client level and updates the assessment as new transactional information becomes available.

Feature engineering was central to the prototype. More than 30 input features were developed as functions of client, account, and transaction data. The project deliberately avoided relying exclusively on arbitrary fixed thresholds, because such thresholds may be easy to evade and may not generalize across heterogeneous customer profiles. Instead, features were designed to capture behavioural patterns and deviations from client-specific baselines.

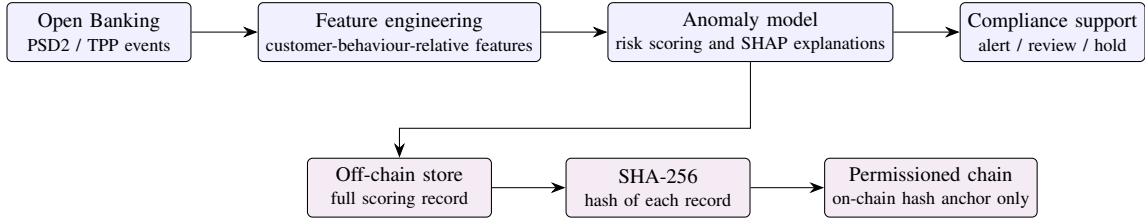


Fig. 1. OBAF architecture: analytics pipeline and blockchain audit trail. Per-client analytical records are stored off-chain; only a cryptographic hash of each record is anchored on a permissioned blockchain, with no personal data stored on-chain.

TABLE I
EXPERIMENTAL SETUP AND VALIDATION SCOPE OF THE OBAF PROTOTYPE.

Component	Configuration	Validation status	Role in the prototype
Isolation Forest	<code>n_estimators=200;</code> <code>contamination=0.035;</code> <code>random_state=1013</code>	In-sample AUC	Selected anomaly detector
One-Class SVM	RBF; $\nu \approx 0.067$; $\gamma \approx 1.29$	In-sample AUC	Alternative anomaly detector
Local Outlier Factor	$k = 10$; Euclidean metric	In-sample AUC	Alternative anomaly detector
Random Forest	12-fold shuffled stratified CV; fraud class weighted	Cross-validated AUC	Supervised baseline
Primary metric	AUC	Appropriate under class imbalance	Discrimination assessment
Scoring stack	Python; scikit-learn; Pandas; Flask	Prototype implementation	Real-time scoring service
Audit API	FastAPI; blockchain connector	Prototype implementation	Hash anchoring and retrieval

Representative feature groups included transaction intensity, inflows and outflows, forwarding of accumulated funds, card-based money-transfer activity, BLIK-related activity, credits and debits, everyday “living” spending inferred from Merchant Category Codes, account tenure, number of active months, number of accounts, total account balance converted to PLN, client age, and indicators of a client’s ties to Poland. Some features were constructed from transaction sequences, for example by matching inflows and outflows to approximate the forwarding of accumulated funds. Other features captured scale, frequency, account history, or behavioural concentration.

The key refinement was the construction of customer-behaviour-relative features. Instead of evaluating only absolute transaction values or global population-wide thresholds, selected variables were relativised against the client’s own behavioural history, including activity per active month and ratios to the client’s ordinary “living” spending. This encoding was intended to capture whether a given behaviour was unusual for a specific client, not merely whether it was large or unusual in absolute terms. This distinction is important in Open Banking environments, where legitimate customers may have very different transaction scales and usage patterns.

The prototype evaluated three unsupervised anomaly-detection algorithms: Isolation Forest, One-Class SVM, and Local Outlier Factor. Isolation Forest was configured with `n_estimators=200`, `contamination=0.035`, and `random_state=1013`. One-Class SVM used an RBF kernel, with $\nu \approx 0.067$ and $\gamma \approx 1.29$. Local Outlier Factor used $k = 10$ neighbours and the Euclidean metric. These methods were selected because they represent different anomaly-detection principles: random isolation, support estimation, and local-density comparison, respectively.

As a supervised baseline, a Random Forest classifier was also evaluated. This baseline is important for interpreting the results. The argument for anomaly detection is not that supervised learning performed poorly in the prototype. On the contrary, the Random Forest baseline reportedly achieved very strong discrimination under 12-fold shuffled, stratified cross-validation. The operational problem was different: confirmed fraud labels were not sufficiently available across all client cohorts. A supervised classifier may therefore perform well where labels exist, while still failing to provide full operational coverage across the client base. Unsupervised anomaly detection was preferred because it can be applied also to cohorts with few or no confirmed fraud cases.

AUC was used as the primary discrimination metric. This choice follows from the strong class imbalance of the problem: simple accuracy is not informative when confirmed fraud represents a very small share of the population. For the supervised Random Forest baseline, the minority fraud class was weighted more heavily to reflect the asymmetric cost of a missed fraud. For anomaly-detection models, fraud labels were not used during training; they were used only ex post to compute AUC and assess whether known fraud cases received abnormality scores.

The validation status of the reported results must be stated carefully. The anomaly-detection results reported in the project documentation, including the comparison of customer-behaviour-relative and absolute features, should be interpreted as in-sample discrimination on the modelled data only. By contrast, the Random Forest supervised baseline was evaluated with cross-validation. The paper therefore distinguishes between three evidentiary layers: internal anomaly-detection discrimination, cross-validated supervised baseline performance,

and architectural feasibility of the blockchain-supported audit trail. This distinction is central to the positioning of the paper as preliminary prototype evidence rather than as a completed deployment study.

The implementation used a Python-based modelling stack, including scikit-learn and Pandas, with a Flask-based scoring service. A separate FastAPI service mediated communication with the blockchain audit-trail layer. Table I summarizes the experimental setup and marks the validation scope of each component.

V. PROTOTYPE EVIDENCE

This section reports prototype evidence from the OBAF industrial R&D project. The results should be interpreted in line with the validation scope defined in Section IV. They demonstrate feasibility, discrimination on the modelled data, operational speed, and audit-trail design, but they do not constitute production-level validation.

A. Discrimination of Evaluated Methods

Table II separates two issues that should not be confused: the comparison of anomaly-detection algorithms and the separate comparison between absolute and customer-behaviour-relative features. In the anomaly-detection comparison, Isolation Forest achieved AUC of approximately 0.99, One-Class SVM approximately 0.98, and Local Outlier Factor approximately 0.94. These values refer to in-sample discrimination on the modelled data, with labels used only ex post for AUC computation.

The supervised Random Forest baseline reportedly achieved a comparable AUC of approximately 0.995 under 12-fold shuffled, stratified cross-validation. This result is important because it shows that the decision to use anomaly detection was not driven by poor supervised-model performance. The reason was operational coverage: confirmed fraud labels were too sparse or absent in some client cohorts, whereas anomaly detection can be applied also where labelled fraud cases are not available. In other words, the supervised baseline was strong where labels existed, but the anomaly-detection approach was better aligned with the need to cover the whole client base.

Given similar discrimination among the best-performing methods, Isolation Forest was preferred as the final anomaly-detection model because it combined strong discrimination, fast scoring, relatively low computational cost, compatibility with SHAP-based explanation, and the possibility of incremental warm-start updates. This is an operational rather than purely statistical selection criterion: in a fraud-screening system, inference latency, explainability, and model maintainability are part of model quality.

B. Effect of Customer-Behaviour-Relative Features

The most important feature-engineering result concerns the difference between absolute and customer-behaviour-relative features. Absolute features evaluate transactions or customer activity by their raw values or global scale. Customer-behaviour-relative features instead compare behaviour with the

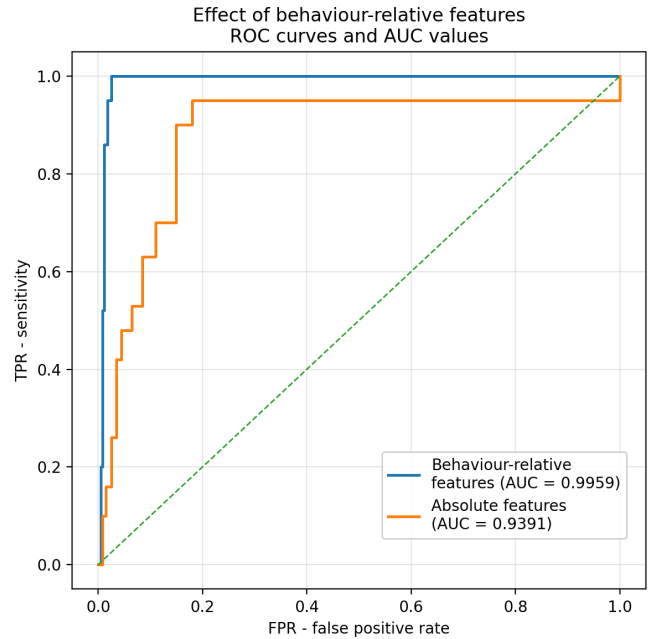


Fig. 2. Effect of customer-behaviour-relative features on prototype discrimination quality. The comparison concerns feature construction, not the choice of anomaly-detection algorithm.

client's own historical baseline. This distinction is crucial in Open Banking: a transfer pattern that is ordinary for one client may be highly unusual for another.

In the reported comparison, customer-behaviour-relative features improved AUC from 0.939 to 0.996 relative to absolute features. Figure 2 presents this comparison as ROC curves. This result should not be confused with the algorithm comparison reported above: the value around 0.94 in the feature-engineering comparison refers to the absolute-feature variant, whereas the value around 0.94 in the algorithm comparison refers to Local Outlier Factor. They are different comparisons and answer different questions.

The interpretation is that customer-specific baselines carry substantial information for anti-fraud monitoring. They reduce the ambiguity created by heterogeneous customer behaviour and help avoid treating all clients as if they followed the same transactional norm. This supports the central design choice of the OBAF prototype: fraud screening should be based on behavioural deviation relative to the client, not only on static global thresholds.

C. Cross-Cohort Robustness

The project also reports cross-cohort checks in which the model was re-estimated on disjoint customer groups. Preliminary in-sample results suggest that discrimination remained high across age-based cohorts (AUC 0.963 for younger clients and 0.965 for older clients), tax-residency cohorts (0.973 for Polish tax residents and 0.955 for foreign tax residents), and wealth-based cohorts (0.979 for less-wealthy clients and 0.933 for more-wealthy clients). These checks should not be

TABLE II
PROTOTYPE DISCRIMINATION RESULTS AND VALIDATION STATUS.

Comparison	Method or variant	Reported AUC	Validation status / interpretation
Anomaly-detection algorithm	Isolation Forest	≈ 0.99	In-sample; selected model
Anomaly-detection algorithm	One-Class SVM	≈ 0.98	In-sample; strong but more parameter-sensitive
Anomaly-detection algorithm	Local Outlier Factor	≈ 0.94	In-sample; weaker discrimination
Supervised baseline	Random Forest	≈ 0.995	12-fold stratified CV; limited by label coverage
Feature engineering	Absolute features	0.939	In-sample feature comparison
Feature engineering	Customer-behaviour-relative features	0.996	In-sample feature comparison; headline result

interpreted as external validation. They are better understood as preliminary robustness evidence within the industrial prototype. Their value lies in showing where transferability appears stronger and where it may degrade. The weaker result for the wealthier-customer group suggests that customer segmentation remains an important topic for future validation.

D. Operational Speed

Operational speed is a necessary condition for deployment in fraud screening. A model with high discrimination but high latency would be difficult to integrate into payment-institution workflows. In the OBAF prototype, single-client scoring reportedly took approximately 0.037 seconds per customer. Batch scoring reduced the average scoring time to approximately 5.9×10^{-5} seconds per customer. The difference suggests that single-client latency is dominated by model-invocation overhead, while the scoring decision itself is computationally negligible in batch mode.

The project also reports that the production API returned a score in approximately 100 ms. These timing figures are best interpreted as prototype-level evidence that anomaly scoring can be performed without introducing substantial delay into the transaction monitoring process. They do not by themselves demonstrate production readiness, since production deployment would also require stress testing, monitoring of failure modes, and integration with institutional decision procedures.

E. Cross-Institution Feature Sharing

The project also explored a mechanism for transferring anti-fraud knowledge between institutions without sharing customer data or a trained model. The reported approach transfers the source code of feature definitions as a compressed and timestamped artifact, rather than raw transaction data or customer-level records. The receiving institution recomputes features on its own data and trains locally. This design is aimed at reducing privacy, bank-secrecy, and distribution-shift risks while supporting privacy-preserving, data-protection-friendly information exchange.

A reported simulation completed in approximately 10.3 seconds. Most of this time was reportedly spent on feature recomputation rather than file transfer. This result is operationally relevant because it suggests that feature-definition sharing may be a practical mechanism for cross-institution learning without direct pooling of sensitive data. At the same time, the result is only a prototype simulation and does not yet show generalization across multiple financial institutions.

F. Blockchain Audit Trail

The blockchain component was designed to support auditability, not prediction. Per-client analytical records are kept off-chain in a controlled relational store. A cryptographic hash of selected records is then anchored on a permissioned blockchain. The project design therefore follows a hybrid off-chain/on-chain pattern: the full record remains under institutional control, while the blockchain stores a tamper-evident commitment that can later be used to verify whether the off-chain record has changed.

In the reported design, the hashed record included selected fields such as client identifier, normality score, last transaction operation identifier, and timestamp. The on-chain component stored only hash anchors, not personal data or full transaction records. This is important for regulatory and data-protection reasons. A blockchain audit trail that stored personal data directly on-chain would create serious governance and privacy problems. Hash anchoring avoids this by separating evidentiary integrity from data disclosure.

The audit-trail layer therefore complements the anomaly-detection workflow. It does not increase AUC and should not be evaluated as a predictive component. Its value lies in traceability, non-repudiation, and the ability to reconstruct whether selected monitoring actions were performed and whether analytical records were later altered. This is especially relevant in regulated financial environments where compliance teams may need to justify escalation, review, or holding decisions after the fact.

VI. DISCUSSION: DESIGN IMPLICATIONS AND LIMITATIONS

The OBAF prototype suggests three design implications for Open Banking anti-fraud systems. First, feature engineering may be as important as the choice of the anomaly-detection algorithm. The strongest prototype result is not only that Isolation Forest performed well, but that customer-behaviour-relative features substantially outperformed absolute features. This supports a practical design principle: suspicious behaviour should be evaluated relative to the customer's own behavioural baseline whenever possible, while global thresholds should be treated as auxiliary escalation rules rather than the main detection logic.

Second, the Random Forest baseline changes the interpretation of anomaly detection. The argument for anomaly detection is not that supervised models are inherently ineffective. The

supervised baseline reportedly achieved strong cross-validated discrimination where labelled fraud cases were available. The issue is operational coverage: if confirmed fraud labels are sparse or absent in some cohorts, a supervised model cannot be applied uniformly across the whole client base without assuming that labels transfer across heterogeneous groups. Anomaly detection offers a different operational logic because it can assign abnormality scores also where labelled fraud examples are insufficient.

Third, the blockchain component should be evaluated as an evidentiary layer, not as a predictive layer. It does not improve AUC directly. Its role is to preserve tamper-evident commitments to selected analytical records and monitoring actions. This narrow use of blockchain is deliberate: full customer or transaction data remain off-chain, while the ledger stores only verification anchors. The resulting architecture integrates behavioural anomaly detection, explainable scoring, and tamper-evident evidence preservation in a single workflow.

The limitations are substantial. The anomaly-detection AUC results are in-sample discrimination on the modelled data, not held-out, temporal, or production-level validation. The same caution applies to the comparison between absolute and customer-behaviour-relative features. A further limitation is that absolute dataset counts and the observation period are not disclosed. Customer behaviour, product mix, label availability, and compliance procedures may differ across institutions.

The fraud labels are also an operational artifact. Confirmed fraud depends on reporting, investigation, detection procedures, and internal record-keeping, and therefore may not cover all fraudulent behaviour. Moreover, the supervised and unsupervised models answer different questions: Random Forest is trained on available labels, while Isolation Forest, One-Class SVM, and LOF use labels only *ex post* for AUC computation. Their comparison is therefore operationally relevant, but not a pure like-for-like contest.

Finally, the timing and blockchain results should be read as prototype evidence. Sub-second scoring and hash anchoring are necessary conditions for practical use, but production deployment would still require load testing, failure handling, model-drift monitoring, access-control auditing, key management, incident-response procedures, and human oversight. The current evidence supports feasibility of the integrated architecture; it does not yet support claims of generalizable fraud-loss reduction or production-grade model performance.

VII. RESEARCH AGENDA AND CONCLUSION

Moving from prototype evidence to production-grade validation requires five next steps. First, anomaly-detection results should be tested on held-out and later-period data to assess temporal stability under changing customer behaviour and fraud strategies. Second, the architecture should be evaluated across additional institutions with different customer profiles, transaction structures, product mixes, and compliance procedures. Third, future work should compare anomaly detection with additional supervised, semi-supervised, cost-sensitive,

and rule-based baselines, and should include ablation studies of feature groups, customer-behaviour-relative transformations, SHAP-based explanations, and blockchain anchoring. Fourth, explainability governance should be tested with operational users, because explanations influence escalation, review, and documentation. Fifth, the audit-trail layer should be evaluated under realistic latency, throughput, resilience, access-control, key-management, and evidence-reconstruction requirements.

In conclusion, this paper has presented OBAF as a prototype-based position paper on Open Banking anti-fraud monitoring. The proposed architecture combines customer-behaviour-relative anomaly detection, explainable scoring, and blockchain-based hash anchoring of selected analytical records. Prototype results suggest high discrimination on the modelled data, strong value of customer-behaviour-relative features, sub-second scoring, and feasible hash-anchored auditability. At the same time, the evidence remains preliminary: the anomaly-detection results are not yet production-level validation, and the system has not been tested across multiple independent institutions.

The main contribution is architectural and programmatic. The paper identifies a practical design pattern for regulated Open Banking anti-fraud systems and formulates the validation agenda required to turn that pattern into a production-grade solution. The broader claim is not that blockchain or anomaly detection alone solves financial fraud. Rather, fraud monitoring may benefit from a combined approach in which adaptive behavioural models are paired with tamper-evident evidence preservation and clear governance boundaries.

ACKNOWLEDGMENT

This paper draws on results obtained in the R&D project “Research and development work on an Open Banking Anti-Fraud PSD2 system based on blockchain technology and machine learning in transaction monitoring and fraud detection processes” (grant no. POIR.01.01.01-00-0976/20), implemented by Igoria Trade S.A. The project was co-financed by the European Union from the European Regional Development Fund under Sub-measure 1.1.1 of the Smart Growth Operational Programme 2014–2020, administered by the National Centre for Research and Development (NCBR), Poland. The author contributed to the project as R&D manager.

The author thanks the Igoria Trade S.A. project team, especially Kacper Wikeł, for technical documentation and discussions used as source material for this paper.

Parts of the underlying technical solution are publicly disclosed in Polish Igoria Trade patent application (pending) [18].

The author used large language model tools to support translation of Polish source material into English, language editing, error correction, LaTeX formatting, and restructuring of selected manuscript passages under the author’s direction. All substantive claims, methodological descriptions, numerical results, references, and the final wording were verified by the author, who takes full responsibility for the content.

REFERENCES

- [1] Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, OJ L 337, 23.12.2015, pp. 35–127. ELI: <http://data.europa.eu/eli/dir/2015/2366/oj>.
- [2] M. Gounari, G. Stergiopoulos, K. Pipiros, et al., “Harmonizing open banking in the European Union: an analysis of PSD2 compliance and interrelation with cybersecurity frameworks and standards,” *International Cybersecurity Law Review*, vol. 5, pp. 79–120, 2024, doi: 10.1365/s43439-023-00108-8.
- [3] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,” *Decision Support Systems*, vol. 50, no. 3, pp. 559–569, 2011, doi: 10.1016/j.dss.2010.08.006.
- [4] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, “Learned lessons in credit card fraud detection from a practitioner perspective,” *Expert Systems with Applications*, vol. 41, no. 10, pp. 4915–4928, 2014, doi: 10.1016/j.eswa.2014.02.026.
- [5] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009, doi: 10.1109/TKDE.2008.239.
- [6] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys*, vol. 41, no. 3, Article 15, 2009, doi: 10.1145/1541880.1541882.
- [7] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in *Proceedings of the 2008 IEEE International Conference on Data Mining*, Pisa, Italy, 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.
- [8] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001, doi: 10.1162/089976601750264965.
- [9] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, TX, USA, 2000, pp. 93–104, doi: 10.1145/342009.335388.
- [10] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017, pp. 4765–4774.
- [11] M. Esche, L. Ho, M. Nischwitz, and S. Glesner, “Practical security of evidence for regulated artificial intelligence modules,” *Annals of Computer Science and Information Systems*, vol. 44, pp. 23–30, 2025, doi: 10.15439/2025F4971.
- [12] B. Schneier and J. Kelsey, “Secure audit logs to support computer forensics,” *ACM Transactions on Information and System Security*, vol. 2, no. 2, pp. 159–176, 1999, doi: 10.1145/317087.317089.
- [13] B. Putz, F. Menges, and G. Pernul, “A secure and auditable logging infrastructure based on a permissioned blockchain,” *Computers & Security*, vol. 87, Article 101602, 2019, doi: 10.1016/j.cose.2019.101602.
- [14] Igoria Trade S.A., “Research and development work on an Open Banking Anti-Fraud PSD2 system based on blockchain technology and machine learning in transaction monitoring and fraud detection processes – project assumptions from the grant application,” internal project document, 2020.
- [15] Igoria Trade S.A., “Open Banking Anti Fraud PSD2 Report: Stage 3 – Anti-Fraud System,” internal R&D report, 2023.
- [16] Igoria Trade S.A., “Open Banking Anti Fraud PSD2 Report: Stage 5 – Technical architecture, implementation and deployment documentation,” internal R&D report, draft, 2023.
- [17] W. Kuliński and K. Piech, “Blockchain oraz uczenie maszynowe w identyfikowaniu oszustw finansowych,” in *Kompendium Blockchain. Zagadnienia prawne rynku kryptoaktywów*, K. Piech, G. Sobiecki, and W. Srokosz, Eds. Gliwice, Poland: Wydawnictwo Naukowe Helion, 2026, ISBN 978-83-289-0664-8, forthcoming.
- [18] W. Kuliński, “Computer-aided method of making settlements by banks and other financial institutions, computer-aided method of assessing the risk of transactions made by banks or other banking institutions and a durable data carrier,” Polish patent application P.435406, filed Sept. 22, 2020, published Mar. 28, 2022. Applicant: Igoria Trade S.A.; status: pending.