The OSCAR PUBLISHING Services
American Journal of Applied Science and Technology

VOLUME Vol.05 Issue 11 2025
PAGE NO. 258-264

Hybrid Streaming Intelligence For Real-Time Transactional Fraud Detection: A Kafka-Centric Machine Learning Framework

Dr. Arjun Mehra
Department of Computer Science, University of Auckland, New Zealand

Received: 05 October 2025; **Accepted:** 03 November 2025; **Published:** 30 November 2025

Abstract: Financial transaction fraud in contemporary digital ecosystems is a fast-moving adversarial problem that requires detection systems to be not only accurate but immediate, adaptive, and auditable. This article develops a comprehensive theoretical and design-oriented framework for hybrid streaming intelligence that unites stateful event streaming (with Apache Kafka as the canonical substrate), multiple machine learning paradigms (supervised classification, anomaly detection, graph-based link analysis, and ensemble voting), and alarm-verification mechanisms including text analytics for enriched context. Drawing upon prior technical and academic work on streaming architectures, ML workflow automation, hybrid alarm verification, transaction synthesis, and practical implementations of fraud scoring, the framework articulates how Kafka-centric event topologies can host low-latency triage, mid-path contextual scoring, and deferred deep analysis while preserving system stability and supporting continuous model adaptation (Dunning & Friedman, 2016; Crettaz & Dean, 2019; Sima et al., 2018). The article provides a detailed methodological rationale for mapping ML models into stream processors and microservices, explains feature engineering patterns suitable for streaming contexts, addresses class imbalance and concept drift, and presents a layered alarm-verification design that reduces false positives and operational costs (Singh, 2019; Sahai & Gursoy, 2019). It further examines how graph analytics, generative models, and ensemble adaptive online learning contribute to network-level detection and adversarial resilience (Molloy et al., 2016; Goodfellow et al., 2014; Roshan & Zafar, 2024). The discussion evaluates trade-offs in latency, interpretability, and governance, and proposes an empirical research agenda including sandbox pilots, red-team adversarial testing, and standards for forensic logging. The contribution is a publication-ready synthesis intended to guide both researchers and practitioners seeking to deploy robust, production-grade real-time fraud detection in high-throughput FinTech environments.

Keywords: Real-time fraud detection; Kafka streams; streaming machine learning; alarm verification; ensemble learning; graph analytics.

INTRODUCTION:

The proliferation of digital payments, mobile banking, peer-to-peer transfers, card-not-present commerce, and real-time settlement rails has transformed financial services while simultaneously magnifying the speed, scale, and sophistication of transactional fraud. Fraudsters exploit distributed platforms, automated botnets, synthetic identities, and cross-channel orchestration to maximize exploitation. Consequently, fraud detection has moved from a post-hoc investigative activity to an operational necessity that must function at event velocity. Real-time detection systems are expected to ingest, enrich, score, and act on transactions within tight latency budgets while maintaining high detection fidelity and minimizing friction for legitimate customers. This environment imposes a set of hard engineering and theoretical constraints that traditional batch-oriented analytics and rule-based systems cannot satisfy: throughput and latency requirements, online adaptability to adversarial drift, the need for contextual fusion across heterogeneous signals, and governance expectations for explainability and forensic traceability.

Research and industry responses have converged on several complementary trends. First, streaming architectures—exemplified by Apache Kafka, Kafka Streams, and similar platforms—facilitate high-throughput, ordered, and fault-tolerant ingestion and processing of event flows, enabling continuous feature construction and low-latency scoring (Dunning & Friedman, 2016; Crettaz & Dean, 2019). Second, machine learning has evolved beyond static classifiers to encompass ensembles, anomaly detectors, graph-based models, and deep generative approaches that can identify both known fraud signatures and emergent patterns (Shakya, 2018; Goodfellow et al., 2014; Molloy et al., 2016). Third, alarm verification research has shown that hybrid approaches — combining automated scoring, stream processing, and contextual text analytics — materially reduce false alarm rates and operational workload (Sima et al., 2018). Finally, advances in ML operations (MLOps) and automated workflows provide the mechanisms required to continuously update, evaluate, and deploy models without interrupting high-availability production systems (Rangineeni & Kothandaraman, 2018; Quddus, 2018).

Despite these advances, a gap remains in the academic literature and practitioner guidance: how to design, reason about, and operationalize a hybrid streaming intelligence architecture that integrates these disparate techniques into a coherent, scalable, and auditable real-time fraud detection system. Many existing studies address isolated components—model design, stream processing, or graph analysis—but do not thoroughly synthesize the engineering and theoretical consequences of their integration. The present article aims to fill this gap by offering a rigorous conceptual and design framework for hybrid streaming intelligence centered on Kafka streams, explicating how models map to streaming primitives, how alarm verification and text analytics can be embedded, how ensemble and online adaptive techniques mitigate drift and adversarial adaptation, and how graph analytics and generative methods enhance network-level detection.

The remainder of the paper proceeds as follows. First, the methodological approach for synthesizing streaming and ML design patterns is explained. Second, the proposed hybrid streaming intelligence architecture is described in detail, including dataflow topologies, model placements, feature engineering techniques, and alarm-verification pipelines. Third, theoretical results and behavioral expectations are articulated, analyzing latency-accuracy trade-offs, resilience considerations, and governance implications. Finally, the paper discusses limitations, practical deployment guidance, and an empirical research agenda to validate and refine the framework in real-world FinTech environments.

METHODOLOGY

This research employs a structured conceptual synthesis methodology, combining critical literature analysis with systems-design reasoning. The objective is not empirical benchmarking (no novel dataset is introduced here) but rather to derive an integrative theoretical model grounded in existing validated techniques and engineering practices. The methodology comprises four interdependent activities: corpus synthesis, streaming–ML mapping, architectural design derivation, and operational validation rationale.

Corpus synthesis began with a targeted selection of pivotal works spanning streaming architectures, alarm verification, transaction synthesis, and ML techniques for fraud detection. Core sources include applied streaming literature (Dunning & Friedman, 2016; Crettaz & Dean, 2019), hybrid alarm verification studies (Sima et al., 2018), empirical ML theses and studies in credit card fraud (Shakya, 2018; Singh, 2019; Sahai & Gursoy, 2019), and advanced ML methodologies including gradient boosting (Friedman, 2001; Chen & Guestrin, 2016), CatBoost (Dorogush et al., 2018), deep generative modeling (Goodfellow et al., 2014), and streaming ensemble approaches (Roshan & Zafar, 2024). Additionally, technical and practitioner-oriented works inform the systems-level design and operational constraints (Quddus, 2018; Rangineeni & Kothandaraman, 2018). This diverse corpus ensures that the resulting framework rests on both theoretical robustness and practical feasibility.

The streaming–ML mapping activity translates ML model families and operational requirements into stream-processing primitives. This mapping is conceptualized as a set of logical layers—ingestion, stateful enrichment, low-latency inference, mid-path contextual scoring, and deferred deep analysis—each with specific timing, state, and fault-tolerance requirements that align with Kafka's KStreams/KTables semantics (Dunning & Friedman, 2016). For example, stateless lightweight models align with microservice-based inference colocated on the authorization path for minimal latency, whereas stateful graph analyses operate on micro-batched aggregates read from compacted topics or materialized views. The mapping clarifies placement constraints for models based on update frequency, input dimensionality, and computational footprint.

Architectural design derivation constructs a canonical

hybrid streaming intelligence topology where event flows are partitioned, enriched, scored, verified, and archived. Dataflow choreography leverages Kafka topics for persistence and ordered delivery, KStream processors for low-latency transformations and stateful windowed aggregations, and model-serving microservices for both synchronous and asynchronous inference. Key design principles—idempotent processing, graceful degradation under back-pressure, bounded state stores, and feature-store synchronization—are specified to ensure operational viability.

Operational validation rationale sets forth how the theoretical design can be empirically validated through sandbox pilots, shadow-mode deployments, adversarial red-team testing, and continuous monitoring metrics that measure time-to-detection, false-positive impact, model drift indices, and throughput under scaled workloads. The rationale draws on prior empirical protocols in similar domains and recommends metrics and experiment structures compatible with highly regulated financial environments (Powers, 2011; Magomedov et al., 2018).

Throughout the methodology, the epistemic stance emphasizes transparent assumptions: that event order is preserved within partitions when keys are chosen carefully; that instrumentation and observability are available for latency and drift measurement; that labeled adjudication data can be generated by human-review workflows; and that privacy and compliance constraints will guide storage and retention policies. Under these assumptions, the hybrid streaming intelligence framework is derived and analyzed to set expectations for real-world deployments.

RESULTS

The primary "results" of this conceptual and design research are a detailed hybrid architecture, explicit design prescriptions for mapping ML models into streaming primitives, a set of behavioral expectations (latency, detection fidelity, resilience), and a prioritized empirical validation program. Each of these outputs is presented in-depth below.

Hybrid Streaming Intelligence Architecture — Overview and Rationale

At the highest level, the architecture divides processing into five interacting strata:

1. Event Ingestion and Normalization — Raw transaction events, device telemetry, authentication events, and contextual signals (merchant metadata, geolocation, channel data) are ingested into Kafka topics. Events are keyed by a domain-appropriate routing key (for example, primary account number hash, customer identifier, or device fingerprint) to preserve per-entity ordering and to enable locality of stateful aggregations (Dunning & Friedman, 2016). Normalization includes timestamp alignment, canonical merchant code mapping, and lightweight validation.

2. Low-Latency Feature Extraction and Fast-Path Scoring — Immediately downstream, KStream processors compute ultralow-latency features suitable for authorization-time decisions: recent velocity (counts in last minute), device-match indicators, geolocation distance from prior transaction, merchant risk flags, and lightweight behavioral embeddings. These features feed compact, explainable models—logistic regression with a small feature set, shallow decision trees, or calibrated gradient-boosted models operating in a constrained feature window—that produce a fast-path risk score used for immediate mitigations (allow, step-up authentication, soft decline). This path aims for minimal added latency (single-to-few-hundreds of milliseconds) compatible with payment authorization windows (Crettaz & Dean, 2019; Quddus, 2018).

3. Contextual Enrichment and Mid-Path Scoring — Parallel to the fast path, the enrichment plane aggregates longer-window features (hourly/24-hour aggregates, rolling average amounts), draws from persistent KTable materialized views (for user history or device reputations), and performs link-centric micro-computations (e.g., shared-IP or shared-device counts). The mid-path scoring engines—more expressive models such as medium-sized gradient-boosted trees (XGBoost, CatBoost) or compact neural nets—operate under relaxed latency (sub-second to a few seconds) and refine the initial score, possibly escalating cases to human review or stronger remediation.

4. Deep Analysis and Graph-Based Network Detection (Deferred) — For complex patterns—money laundering chains, synthetic identity rings, or low-and-slow campaigns—deferred processing ingests event windows into graph analytics and deep learning pipelines. This plane includes graph construction (streamed edge creation, rolling community detection), graph representation learning (LaundroGraph-like self-supervised embeddings), and deep autoencoders or generative models to flag anomalies emergent at network scales (Molloy et al., 2016; Cardoso et al., 2022; Raman et al., 2020). These heavy analyses are intentionally asynchronous; they do not block authorization but provide investigative leads, enrich model training datasets, and drive

watchlist updates.

5. Alarm Verification, Text Analytics, and Case Management — All flagged events enter an alarm-verification workflow that reduces false positives and prioritizes cases. The workflow combines rule-based filters, text-analytics (for user-reported notes, merchant descriptions, or external threat intelligence scraped via feeds), and lightweight ML classifiers trained to suppress non-actionable alerts (Sima et al., 2018). Verified alarms are then routed to case-management systems for human adjudication, whose outcomes (confirmed fraud or false positive) feed back into continuous training loops.

This stratified design marries the immediacy of fast-path scoring with the depth of deferred analyses, ensuring that urgent interventions occur without sacrificing the capacity to discover sophisticated, temporally diffuse fraud.

Model Placement and Streaming Primitives Mapping

Placing models into streaming systems requires careful alignment of their operational characteristics with stream-processing semantics.

- Stateless lightweight classifiers (e.g., logistic regression, small decision trees) are deployed as synchronous model servers or embedded within KStream processors to minimize inter-service network calls. Because these models do not require extensive historical context, they operate directly on per-event feature vectors computed from low-latency windowed aggregates.
- Stateful models (e.g., models requiring rolling user histories or device reputations) rely on KTables or external feature stores materialized through compacted topics. The use of exactly-once processing and idempotent writes ensures state consistency during failover (Dunning & Friedman, 2016).
- Ensemble and voting systems combine outputs from multiple models (fast-path, mid-path, and heuristics) via a merge operator that computes a calibrated ensemble risk. Ensemble voting reduces single-model brittleness and allows models with different biases and variances to complement each other (Singh, 2019).
- Graph analytics require specialized streaming graph construction. Edges representing interactions (card–merchant, device–account) are emitted to graph ingestion topics, where micro-batch processors build graph snapshots at defined intervals and execute community detection or motif-finding algorithms (Molloy et al., 2016; Magomedov et al., 2018). Streaming sketches and approximate data structures preserve memory constraints while retaining analytic fidelity.
- Anomaly detectors and deep networks are generally executed asynchronously in a deferred processing plane, consuming windowed aggregates and providing investigative signals and synthetic negative sampling data for adversarial robustness testing (Goodfellow et al., 2014; Raman et al., 2020).

Feature Engineering Patterns for Streaming Contexts

Streaming contexts impose restrictions that change how features are computed and updated. Several patterns are recommended:

- Bounded sliding windows to compute velocity and amount aggregates with strictly bounded state; windows must be tuned to fraud temporalities (instantaneous card testing vs. slow-laundering patterns).
- Exponential decay counters that favor recent behavior while retaining long-term baseline trends; these counters are efficient and adaptive to behavior shifts.
- Micro-embedding updates: compact embeddings for user or device behavior updated incrementally using streaming-aware update rules; these embeddings enable similarity search and drift detection without reprocessing entire histories.
- Probabilistic sketches (HyperLogLog, Count-Min sketches) for approximate distinct counts and frequency estimates, saving memory while providing effective signals for link analysis (e.g., number of unique devices touching multiple accounts).
- Feature provenance tagging: each feature carries a provenance tag (timestamp, source topic, preprocessing transform version) enabling forensic traceability and reproducibility of decisions.

Alarm Verification and Text Analytics Integration

Alarm verification is central to reducing operational costs caused by false positives. The hybrid approach uses three coordinated layers:

1. Automated Suppression Rules: rule-aware filters remove obviously non-actionable alerts (e.g., merchant-initiated refunds within normal bands), based on policy logic and supervised rule-learning.

2. Text Analytics and Context Fusion: free-text merchant descriptions, user comments, and external threat feeds are ingested and processed with lightweight NLP (tokenization, named-entity recognition, semantic similarity) to enrich event context. For example, a merchant name change detected in text feeds may explain a surge of transactions that would otherwise be flagged.

3. Verification Classifiers: supervised models

trained on adjudicated alarm outcomes predict the probability that an alert requires human attention. These classifiers leverage both structured features and text-derived embeddings to improve precision (Sima et al., 2018).

Ensemble and Adaptive Online Learning Strategies

Ensemble strategies provide both robustness and a mechanism for controlled adaptation. Recommended approaches include:

- Stacked ensembles where fast-path models supply inputs to meta-learners that calibrate final risk scores based on historical performance.
- Weight adaptation using online convex optimization: ensemble weights are dynamically adjusted using performance feedback, thus enabling micro-adaptation to shifting fraud tactics.
- Active learning loops that prioritize ambiguous cases for human labeling, improving label efficiency in streaming contexts (Labanca et al., 2022).
- Concept drift detectors that monitor feature distribution shifts and score calibration divergence; upon detection, the system triggers targeted retraining pipelines using recent labeled data.

Generative and Adversarial Methods for Resilience

Generative techniques such as GANs and variational autoencoders (VAEs) play dual roles: data augmentation for rare fraud patterns and probing model vulnerabilities through adversarial sample generation (Goodfellow et al., 2014; Raman et al., 2020). An operational workflow integrates these techniques to synthesize plausible fraudulent variants, test detection coverage, and strengthen models through adversarial training.

Behavioral Expectations and Trade-Off Analysis

The hybrid architecture yields measurable behavioral expectations:

- Latency: Fast-path scoring aims for sub-200ms added latency; mid-path enrichments operate under sub-second constraints; deep-analysis is deliberately asynchronous.
- Detection Rate and Precision: Ensembles and graph analytics improve true-positive detection for coordinated fraud, while alarm verification substantially reduces false positives relative to single-classifier systems (Sima et al., 2018; Singh, 2019).
- Adaptability: Online weight adaptation and active learning ensure responsiveness to drift, reducing undetected fraud windows compared to periodic retraining alone.
- Scalability: Kafka's partitioning supports horizontal scaling; state stores must be bounded and checkpointed to handle failover.

DISCUSSION

The hybrid streaming intelligence architecture reconciles competing objectives: immediacy versus sophistication, explainability versus predictive power, and throughput versus statefulness. The following subsections analyze these tensions, discuss counter-arguments, and examine governance, privacy, and operationalization issues.

Latency versus Model Complexity

A perennial tension exists between deploying powerful, high-capacity models and meeting authorization time constraints. The architecture's tiered design resolves this by decoupling immediate action from deeper analysis: ultrafast decisions are based on compact features and explainable models, while complex inference runs asynchronously. This approach respects user experience requirements while still leveraging deep models for network-level detection and model improvement. Empirical implementations must carefully tune thresholds where asynchronous analysis modifies subsequent behavior (e.g., dynamic watchlist updates).

Explainability, Accountability, and Regulatory Compliance

Financial institutions operate under intense regulatory scrutiny; model decisions affecting customer access require interpretability and documented evidence. The architecture favors explainable models in high-impact decision paths and maintains feature provenance and audit logs to support regulatory review. Moreover, verification classifiers and human adjudication provide the narrative context necessary in dispute handling. There is a counter-argument advocating for all-powerful black-box models to maximize detection; however, in regulated contexts, opaque models increase legal risk and customer dissatisfaction. A principled compromise is to use black-box models for advisory insights and transparent models for direct action, with recorded rationale linking the two.

Adversarial Dynamics and Systemic Risk

Fraud detection operates in an adversarial ecology. Attackers adapt to detection signals, exploiting blind spots and orchestrating distributed, low-and-slow campaigns. Continuous adversarial testing—GAN-based sample generation, red-team penetration of model inputs, and monitoring for suspiciously engineered feature distributions—is essential. The architecture's ensemble and active learning loops are

designed to shrink attack surface by increasing modeling diversity and improving label acquisition efficiency. Importantly, system-level defenses (anomaly thresholds, rate-limiting, and immutable audit trails) complement model-based detection.

Privacy, Data Governance, and Forensics

High-fidelity detection often requires rich personal data; privacy regulations mandate careful governance. The architecture supports privacy-preserving techniques: retention policies, differential access controls, data minimization in streams, and cryptographic commitment of logs when necessary for forensic integrity. For cross-institution network detection, privacy-preserving federated learning or secure multiparty computation are promising directions but introduce complexity and require legal frameworks for data sharing.

Operational Complexity and Organizational Readiness

Deploying hybrid streaming intelligence demands advanced engineering capabilities: stream engineering, MLOps, model validation, and robust observability. Smaller institutions may lack such resources; managed platforms or consortium-based approaches can distribute costs but require governance alignment. The architecture emphasizes modularity to allow incremental adoption: start with fast-path scoring and alarm verification, add mid-path enrichments, and progressively integrate deferred graph analytics.

Limitations and Research Directions

This conceptual synthesis leaves several empirical questions open. Quantitative trade-offs (e.g., exact latency-performance curves under specific workloads) require production-scale pilots. The relative value of graph analytics versus enhanced feature engineering in different financial contexts requires measured evaluation. Adversarial testing across multi-party ecosystems remains nascent. Future research should implement sandbox pilots with shadow deployments, formal red-team protocols, and rigorous measurement of operational KPIs (time-to-detection, false-positive impact, cost-weighted utility). Additionally, formal studies of legal admissibility of derived evidence (e.g., graph-match outputs) and consumer responses to automated mitigations will guide responsible deployment.

CONCLUSION

This article presents a rigorous hybrid streaming intelligence framework for real-time transactional fraud detection anchored on Kafka streams and a spectrum of machine learning techniques. By partitioning responsibilities into fast-path scoring, mid-path enrichment, deferred deep analysis, and alarm verification with text analytics, the architecture balances the urgent need for low-latency mitigation with the analytical depth required to detect sophisticated, network-level fraud. Ensemble and adaptive online learning strategies provide robustness against concept drift and adversarial evolution, while generative and adversarial methods strengthen resilience by exposing model blind spots. Practical deployment requires careful engineering—bounded state, idempotent processing, observability, and privacy governance—and an incremental adoption strategy that begins with explainable fast-path models and progressively incorporates deeper analytics.

Real-world validation is the immediate next step: shadow deployments in production streams, adversarial red-team testing, and inter-organizational pilots for cross-channel fraud detection. In doing so, the research and practitioner communities can transform hybrid streaming intelligence from a promising design pattern into a resilient operational capability that materially reduces fraud losses while preserving customer trust and regulatory compliance.

REFERENCES

1. Charron, Justin, et al. Performing Transaction Synthesis through Machine Learning Models. 2017.
2. Shakya, Ronish. Application of machine learning techniques in credit card fraud detection. MS thesis. University of Nevada, Las Vegas, 2018.
3. Crettaz, Valentin, and Alexander Dean. Event Streams in Action: Real-time event systems with Kafka and Kinesis. Simon and Schuster, 2019.
4. Dunning, Ted, and Ellen Friedman. Streaming architecture: new designs using Apache Kafka and MapR streams. O'Reilly Media, Inc., 2016.
5. Quddus, Jillur. Machine Learning with Apache Spark Quick Start Guide: Uncover patterns, derive actionable insights, and learn from big data using MLlib. Packt Publishing Ltd, 2018.
6. Rangineeni, Yasodhara Varma, and Manivannan Kothandaraman. Automating and Scaling ML Workflows for Large Scale Machine Learning Models. JOURNAL OF RECENT TRENDS IN COMPUTER SCIENCE AND ENGINEERING (JRTCSE), vol. 6, no. 1, May 2018, pp. 28–41.
7. Singh, Kuldeep Kaur Ragbir. A Voting-Based Hybrid Machine Learning Approach for Fraudulent Financial Data Classification. MS thesis. University of Malaya, 2019.

8. Ellis, Byron. Real-time analytics: Techniques to analyze and visualize streaming data. John Wiley & Sons, 2014.
9. Sahai, Lakshya, and Kemal Gursoy. Real-time credit card fraud detection. 2019.
10. Molloy, I., Chari, S., Finkler, U., Wiggerman, M., Jonker, C., Habeck, T., Park, Y., Jordens, F., & Schaik, R. Graph analytics for real-time scoring of cross-channel transactional fraud. 2016.
11. Ariyaluran Habeeb, R. A., Nasaruddin, F., Gani, A., Amanullah, M. A., Hashem, A. T. I., Ahmed, E., & Imran, M. Clustering-based real-time anomaly detection—a breakthrough in big data technologies. Transactions on Emerging Telecommunications Technologies, 33(8):3647, 2022.
12. Roshan, K., & Zafar, A. Ensemble adaptive online machine learning in data stream: a case study in cyber intrusion detection system. International Journal of Information Technology, 2024.
13. Srinivas, K., Prasanth, N., Trivedi, R., Bindra, N., & Raja, S. A novel machine learning inspired algorithm to predict real-time network intrusions. International Journal of Information Technology, 14(7):3471–3480, 2022.
14. Surianarayanan, C., Kunasekaran, S., & Chelliah, P. R. A high-throughput architecture for anomaly detection in streaming data using machine learning algorithms. International Journal of Information Technology, 16(1):493–506, 2024.
15. Richardson, L., Amundsen, M., & Ruby, S. RESTful web APIs. O'Reilly Media, Sebastopol, 2013.
16. Flask. Projects P. Flask documentation. https://flask.palletsprojects.com/. Accessed 19 Feb 2024.
17. Streamlit Inc. Streamlit: the fastest way to build and share data apps. https://streamlit.io/. Accessed 20 Apr 2023.
18. Hebbar, K. S. AI-DRIVEN REAL-TIME FRAUD DETECTION USING KAFKA STREAMS IN FINTECH. International Journal of Applied Mathematics, 38(6s), 770–782, 2025.
19. Badrulhisham, F., Pogatzki-Zahn, E., Segelcke, D., Spisak, T., & Vollert, J. Machine learning and artificial intelligence in neuroscience: a primer for researchers. Brain, Behavior, and Immunity, 115:470–479, 2024.
20. Raman, M. G., Dong, W., & Mathur, A. Deep autoencoders as anomaly detectors: method and case study in a distributed water treatment plant. Computers & Security, 99:102055, 2020.
21. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. Generative adversarial nets. Advances in Neural Information Processing Systems, 27, 2014.
22. Friedman, J. H. Greedy function approximation: a gradient boosting machine. Annals of Statistics, 2001.
23. Dorogush, A. V., Ershov, V., & Gulin, A. CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363, 2018.
24. Chen, T., & Guestrin, C. XGBoost: a scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
25. Powers, D. M. W. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. Tech Science Press, Henderson, 2011.
26. Magomedov, S., Pavelyev, S., Ivanova, I., Dobrotvorsky, A., Khrestina, M., & Yusubaliev, T. Anomaly detection with machine learning and graph databases in fraud management. International Journal of Advanced Computer Science and Applications, 9(11), 2018.