UDC 004.8:005.8
JEL Classification: G20; L86

A HYBRID MODULAR ARCHITECTURE FOR FRAUD DETECTION USING OFFLINE AND ONLINE MACHINE LEARNING MODELS

©2025 **CAPRIAN I.**

UDC 004.8:005.8
JEL Classification: G20; L86

Caprian I.

A Hybrid Modular Architecture for Fraud Detection Using Offline and Online Machine Learning Modelsn

This article proposes a hybrid and modular architecture for fraud detection that integrates both offline and online machine learning models to address challenges in dynamic financial transaction environments. The framework combines high-performance offline models, including XGBoost, LightGBM, and deep neural networks, with lightweight and adaptive online learners, such as Hoeffding Trees and Adaptive Random Forests, enabling accurate detection in both historical datasets and real-time streaming transactions. A key methodological contribution lies in balancing predictive performance, responsiveness, and interpretability, achieved through a weighted risk scoring mechanism and a unified cost-sensitive evaluation framework that aligns technical metrics with tangible financial impacts. The architecture emphasizes modularity and scalability, facilitating continuous adaptation via concept drift detection and feedback-driven retraining. Its implementation in a containerized, open-source environment ensures reproducibility, robustness, and seamless deployment in production-grade financial ecosystems, even under high-volume transactional loads. The proposed system effectively bridges the gap between advanced machine learning research and operational requirements, providing a flexible, interpretable, and operationally viable solution for modern fraud detection. Furthermore, this study consolidates previous work by the author on intelligent fraud detection systems, extending prior contributions in model selection, AI interpretability, and economic evaluation of false positives in banking contexts. Future research directions include integrating graph-based relational features for network fraud detection, applying reinforcement learning for adaptive decision optimization, and employing federated learning techniques to enhance data privacy across institutions. Overall, the proposed framework represents a scalable, transparent, and adaptive approach that evolves alongside emerging fraud strategies, delivering a deployable system with practical and financial relevance.

***Keywords**: fraud detection, hybrid architecture, machine learning, concept drift, model evaluation, explainability, financial risk.*

***DOI:** https://doi.org/10.32983/2222-0712-2025-3-312-320*

***Fig.:** 6. **Formulae:** 5. **Bibl.:** 12.*

***Caprian Iurie** – Postgraduate Student, State University of Moldova (60 Alexei Mateevici Str., Kishinev, MD-2009, Moldova)*

***E-mail:** iuriecaprian@gmail.com*

***ORCID:** https://orcid.org/0000-0001-5484-3087*

УДК 004.8:005.8
JEL Classification: G20; L86

Капріан Ю. Гібридна модульна архітектура для виявлення шахрайства з використанням офлайн- та онлайн-моделей машинного навчання

У цій статті пропонується гібридна та модульна архітектура для виявлення шахрайства, яка інтегрує як офлайн-, так і онлайн-моделі машинного навчання для розв'язання проблем у динамічних середовищах фінансових транзакцій. Фреймворк поєднує високопродуктивні офлайн-моделі, включаючи XGBoost, LightGBM та глибокі нейронні мережі, з легкими та адаптивними навчальними моделями онлайн (наприклад, дерево Хоффдінга, адаптивний метод випадкових лісів), що дозволяє точно виявляти шахрайство як в історичних наборах даних, так і у потокових транзакціях у реальному часі. Ключовий методологічний внесок полягає у балансуванні прогнозної продуктивності, чутливості та інтерпретації, що досягається за допомогою механізму зваженої оцінки ризиків та єдиної економічно чутливої системи оцінювання, яка узгоджує технічні показники з відчутними фінансовими наслідками. Архітектура підкреслює модульність і масштабованість, сприяючи постійній адаптації за допомогою виявлення дрейфу концепцій і перенавчання на основі зворотного зв'язку. Її реалізація у контейнерному середовищі з відкритим кодом забезпечує відтворюваність, надійність і безперебійне розгортання у фінансових екосистемах виробничого рівня, навіть за великих обсягів транзакційних навантажень. Запропонована система ефективно усуває розрив між передовими дослідженнями машинного навчання та операційними вимогами, забезпечуючи гнучке, інтерпретоване та операційно життєздатне рішення для сучасного виявлення шахрайства. Крім цього, це дослідження консолідує попередню роботу автора з інтелектуальних систем виявлення шахрайства, розширюючи попередній внесок у вибір моделей, інтерпретацію штучного інтелекту та економічну оцінку хибно позитивних результатів у банківських контекстах. Майбутні напрямки досліджень включають інтеграцію реляційних ознак на основі графів для виявлення мережевого шахрайства, застосування навчання з підкріпленням для адаптивної оптимізації рішень і використання методів федеративного навчання для підвищення конфіденційності даних в установах. Загалом запропонована структура являє собою масштабований, прозорий та адаптивний підхід, який розвивається разом із новими стратегіями боротьби з шахрайством, забезпечуючи розгортання системи з практичною та фінансовою значущістю.

***Ключові слова:** виявлення шахрайства, машинне навчання, модульна архітектура, адаптивне навчання, оцінка ризику, продуктивність моделей, концептуальний дрейф, фінансові технології.*

***Рис.:** 6. **Формул:** 5. **Бібл.:** 12.*

***Капріан Юрій** – аспірант, Державний університет Молдови (вул. Олексія Матеєвича, 60, Кишинів, MD-2009, Молдова)*

***E-mail:** iuriecaprian@gmail.com*

***ORCID:** https://orcid.org/0000-0001-5484-3087*

Introduction. Financial fraud, particularly in the context of electronic banking and digital transactions, represents a growing global threat with severe economic consequences. As financial institutions increasingly rely on automated systems for transaction processing, the need for intelligent, adaptive fraud detection mechanisms becomes critical.

Traditional fraud detection approaches often struggle to maintain accuracy over time due to the evolving nature of fraudulent behavior – a phenomenon known as concept drift [4]. In addition, many systems do not effectively balance the trade-off between comprehensive offline analysis and the low-latency requirements of real-time detection.

This article proposes a hybrid modular architecture that combines offline and online machine learning models to detect and prevent fraudulent transactions. The system leverages historical data for high-precision training and real-time data streams for rapid, adaptive predictions.

The main methodological and practical contributions of this study are as follows:

- A scalable, modular architecture that supports both batch and streaming data processing;
- A risk scoring mechanism that incorporates expert-driven and data-driven features;
- A concept drift detection strategy to trigger model retraining when needed;
- A multi-criteria evaluation framework, including performance, cost, and robustness metrics.

Literature Review. ***Related Work.*** Fraud detection in the financial domain has been widely explored using various machine learning (ML) and statistical techniques. Early approaches relied heavily on rule-based systems and traditional classifiers such as logistic regression and decision trees. While interpretable, these models often fall short in detecting sophisticated or evolving fraud patterns.

Recent research has shifted toward ensemble methods (e.g., Random Forest, XGBoost, LightGBM), which have demonstrated improved performance on highly imbalanced datasets. Boosting algorithms, in particular, have been widely adopted for their ability to capture non-linear interactions and minimize false negatives in fraud detection scenarios [1; 2].

Online learning techniques such as Hoeffding Trees, Adaptive Random Forests, and incremental versions of perceptrons have been proposed to address the need for low-latency detection and continuous model adaptation [3; 7]. These models are well-suited for stream data environments but may suffer from lower accuracy compared to their offline counterparts.

Several studies have highlighted the importance of concept drift in fraud detection. Methods for drift detection include window-based techniques, statistical tests, and performance monitoring frameworks [4; 5]. However, most approaches either focus on offline retraining or simplistic online adaptation, without a unified hybrid solution.

Hybrid architectures, which combine offline and online learning, remain relatively underexplored in academic literature. Some recent efforts propose dual pipelines but lack modular design, explicit drift handling, or cost-sensitive evaluation mechanisms [6]. Moreover, few studies offer a reproducible methodology that integrates both real-time streaming and historical data analysis in a scalable framework.

Our work addresses these gaps by proposing a comprehensive architecture that supports model retraining, drift adaptation, and real-time detection, while maintaining high interpretability and performance under real-world constraints.

Despite these advancements, hybrid architectures that combine the strengths of both offline and online learning are still underrepresented in literature. While some emerging studies propose dual-model pipelines, they often lack a truly modular design, dynamic drift adaptation, or integration of cost-sensitive evaluation [6]. Furthermore, few offer end-to-end reproducibility or scalability suitable for industrial deployment.

In this context, our research introduces a modular hybrid framework designed to bridge these gaps, incorporating offline retraining, online adaptation, and real-time decision support, all validated through financial impact metrics and scalable deployment scenarios.

Methodology. *Overview of the Proposed Architecture.* Overview of the Proposed Architecture. The proposed fraud detection system follows a hybrid modular architecture designed to combine the strengths of both offline and online machine learning models. The system comprises three main components:

1. Offline Learning Module – performs high-capacity training on historical, labeled datasets, using models such as XGBoost and LightGBM [2].
2. Online Detection Module – processes real-time transactions with adaptive, low-latency models, including Hoeffding Trees and Adaptive Random Forests [3; 7].
3. Monitoring and Feedback Module – enables performance tracking, concept drift detection [4; 5], and continuous system improvement through feedback loops.

A high-level architectural diagram (see Figure 1) illustrates the interaction between these components. Historical data is processed in batches for offline model training and feature engineering, while live transaction streams are analyzed by incremental learning algorithms for immediate fraud detection. A centralized monitoring mechanism coordinates feedback, triggers model updates, and visualizes alerts via an integrated dashboard.

Data Sources and Preprocessing. To simulate both historical and streaming environments, the methodology uses a combination of public datasets (e.g., Credit Card Fraud Detection Dataset) and synthetic transactional data.

Key preprocessing steps include:

- feature engineering: Creation of time-frequency, geolocation, and behavioral features;
- imbalance handling: Application of SMOTE and random under-sampling techniques [1];
- normalization: Min-max scaling applied to selected numerical variables.

These steps are essential for reducing noise, managing skewed class distributions, and improving downstream model accuracy.

Offline Learning Module. This module trains supervised learning models on labeled historical data to capture long-term fraud patterns. The models used include:

- XGBoost and LightGBM: Gradient boosting frameworks known for handling class imbalance and learning complex nonlinear relationships [2];
- Deep Neural Networks (DNNs): Capture higher-order feature interactions that may not be visible to traditional models [8].

Offline models are periodically retrained and evaluated using stratified 10-fold cross-validation, optimizing for F1-score, AUC-ROC, and cost-based errors.

Online Detection Module. This module handles real-time streaming data using adaptive learning techniques capable of incremental updates without full retraining.

Implemented algorithms:

- Hoeffding Trees;
- Adaptive Random Forests;
- Online Perceptrons.

These models are integrated into streaming infrastructures (e.g., Apache Kafka pipelines) and are optimized for low latency, early fraud flagging, and stream adaptability [3; 7].

Drift Detection and Feedback Loop. Given the evolving nature of fraud, the system incorporates a concept drift monitoring mechanism. Drift is measured using a performance-change formula (Δ):

$$\Delta(t) = |M_t - M_{t-1}| \quad (1)$$

Where M_t is a metric such as $F1$-score or AUC at time t;

Drift is flagged when $\Delta(t)>\varepsilon$, prompting retraining [4; 5].

The feedback loop also includes a manual validation dashboard, alert triggers, and model weight adjustments based on expert input.

Model Evaluation and Metrics. A robust and comprehensive evaluation strategy is essential in fraud detection systems, particularly when models operate in both offline and online environments. The proposed hybrid framework was assessed through a combination of performance metrics, cost-sensitive analyses, and stability tracking to ensure long-term effectiveness and adaptability [8].

Risk Scoring Formula. Each transaction is evaluated through a weighted scoring mechanism that reflects the presence and intensity of risk-related features:

$$Rick_{transaction} = \sum_{i=1}^{n} \omega_i \cdot f_i, \quad (2)$$

Where f_i represents the normalized value of feature iii (e.g., transaction amount, unusual location, anomalous time);

w_i is the risk weight assigned either through data-driven learning or expert calibration [9].

The score is compared to a predefined threshold θ\thetaθ to trigger alerts. The formula also supports feature attribution analysis, offering transparency into the decision-making process [9].

Evaluation Metrics. To rigorously evaluate both the classification performance and operational utility, the following metrics were employed:

- Precision: Proportion of correctly identified frauds among all flagged cases
- Recall (Sensitivity): Proportion of actual frauds correctly detected
- F1-Score: Harmonic mean between precision and recall
- AUC-ROC: Area under the Receiver Operating Characteristic curve
- False Positive Rate (FPR): Critical in minimizing unnecessary alerts
- Response Time: Measures latency in real-time fraud detection [10]

Concept Drift Detection. To handle evolving patterns in fraud, the system integrates a performance-drift monitoring formula:

$$\Delta(t)=|M_t - M_t-1| \quad (3)$$

A drift is flagged when $\Delta(t)>\varepsilon$, indicating a need for retraining [4].

Cost-Based Evaluation. Economic impact is a central criterion in fraud detection. We used the following formula to estimate the total error-related costs:

$$Cost_{total} = C_{FN}.NFN + C_{FP}.N_{FP} \quad (4)$$

Where C_{FN},C_{FP} are the costs assigned to false negatives and false positives, respectively;

N_{FN}, N_{FP} are the number of false negatives and false positives.

Example: A single undetected fraudulent transaction may result in a loss of $10.000, while a false positive may cost $100 due to manual review [1].

Unified Scoring for Model Comparison. To facilitate multi-objective model selection, a composite score was computed using weighted contributions from multiple criteria:

$$Overall\ Score = \alpha \cdot F1 + \beta \cdot (1 - FPR) + \gamma \cdot Stability$$

Weights α, β, γ reflect system-specific priorities such as regulatory compliance or cost-sensitivity [8].

Evaluation Settings. Model performance was assessed under the following conditions:

- Offline Evaluation: Stratified 10-fold cross-validation on historical labeled datasets.
- Online Simulation: Sliding window validation with injected concept drift for stream-based testing.

Robustness Checks: Stress testing with varying feature distributions, noise levels, and input delays.

A comparative chart (see Figure 1) presents the unified scores of XGBoost, Adaptive Random Forest, and Online Perceptron, using weights of 0.5 (F1-score), 0.3 (1 - FPR), and 0.2 (stability). Figure 1 provides a comparative visualization of the composite scores for the three selected models: XGBoost, Adaptive Random Forest, and Online Perceptron. This comparison highlights the overall performance of each model based on a weighted scoring system that combines F1-score, false positive rate (FPR), and stability [8].

The XGBoost model (Figure 1) achieves the highest composite score of 0.93, demonstrating strong classification performance and high stability across scenarios involving concept drift. Adaptive Random Forest follows closely with a score of 0.90, benefiting from real-time adaptability and ensemble robustness. The Online Perceptron, although achieving a lower score of 0.85, remains highly efficient in environments with strict latency constraints and limited computational resources [3]

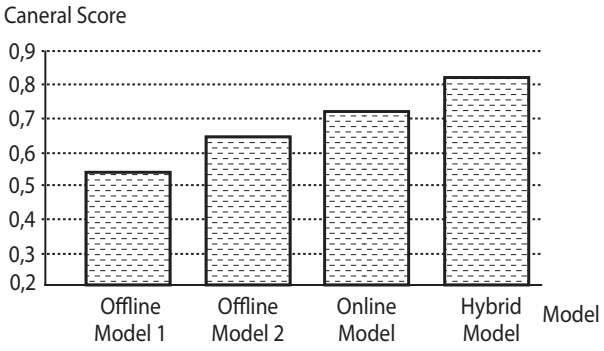

Fig. 2. Cost-Adjusted General Score Comparison

Source: elaborated by the author

The y-axis represents the normalized composite score ranging from 0 to 1, calculated as:

$$Overall\ Score = \alpha \cdot F1 + \beta \cdot (1 - FPR) + \gamma \cdot Stability$$

with weights set to $\alpha = 0.5$, $\beta = 0.3$, and $\gamma = 0.2$ to reflect the system's prioritization of precision, reliability, and robustness. The x-axis displays the evaluated models. This visualization facilitates an integrated assessment of performance, error tolerance, and adaptability under operational conditions.

Figure 2 illustrates the cost-adjusted general scores of four model types – two offline models, one online model, and one hybrid model – based on their ability to minimize financial losses associated with false positives and false negatives.

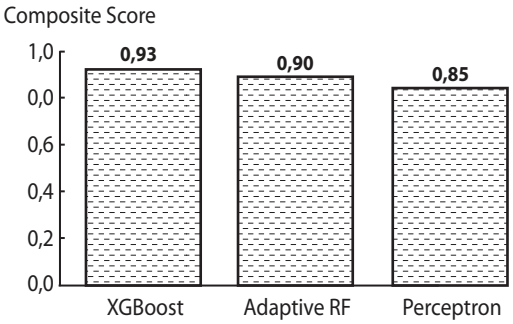

Fig. 1. Composite Model Performance Scores

Source: elaborated by the

The *y-axis* represents the general score adjusted for cost, on a normalized scale from 0.5 to 0.9, while the x-axis denotes the evaluated model types. These scores were computed using the formula:

$$Total\ Cost = C_{FN} \cdot FN + C_{FP} \cdot FP \quad (5)$$

where C_{FN} and C_{FP} are the costs of false negatives and false positives, respectively. The general score reflects the inverse of this cost – higher values indicate better cost-efficiency.

As shown in the chart:

- Offline Model 1 registers the lowest score (~0.55), indicating poor cost performance.
- Offline Model 2 performs slightly better (~0.65), suggesting improvements in classification or calibration.
- The Online Model reaches ~0.72, benefiting from real-time adaptability and lower false negatives.

The Hybrid Model outperforms all others with a score of ~0.83, offering the best trade-off between predictive accuracy and financial risk.

These findings confirm that high classification accuracy alone does not guarantee optimal economic performance. Instead, the hybrid approach – which balances long-term learning with real-time responsiveness – emerges as the most viable option for reducing fraud-related losses in practice.

*Conclusion to Model Evaluation and Metrics***.** The comprehensive evaluation strategy presented in this section highlights the essential role of multi-dimensional performance assessment in modern fraud detection systems. By integrating classical classification metrics (e.g., F1-score, AUC-ROC, precision, recall) with cost-sensitive analysis, concept drift detection, and stability tracking, the proposed methodology enables a more realistic and robust understanding of model behavior in both static and dynamic environments.

A key strength of this framework lies in its ability to bridge technical performance and real-world constraints, such as financial losses caused by undetected fraud and the operational overhead of false alarms. The cost-adjusted scoring scheme, together with unified composite metrics, facilitates strategic model selection based on both predictive power and economic impact.

Furthermore, the inclusion of concept drift detection mechanisms ensures long-term adaptability in the face of evolving adversarial tactics – a defining characteristic of fraud scenarios.

The hybrid evaluation approach, combining offline cross-validation and real-time streaming simulation, mirrors the system's architectural duality and underscores its readiness for deployment in production-grade environments.

In conclusion, this evaluation module transcends its analytical role to function as a decision-support layer – enhancing transparency, traceability, and trust in the broader fraud detection pipeline. It effectively transforms the proposed architecture into a cost-aware, risk-resilient, and operationally viable system aligned with both technical rigor and business priorities.

Implementation. The implementation of the proposed hybrid fraud detection architecture was carried out using a combination of offline and streaming technologies, ensuring both scalability and adaptability. The system was built to support end-to-end fraud detection – from batch training to real-time scoring – while integrating key feedback and monitoring mechanisms.

Technology Stack. The architecture was implemented using the following tools and frameworks:

- Offline Learning Module: Python (scikit-learn, XGBoost, LightGBM, TensorFlow);
- Data Processing: Pandas, NumPy, and Feature-engine for feature construction and preprocessing;
- Online Detection Module: River library (for online machine learning models like Hoeffding Trees and Adaptive Random Forests);
- Streaming Infrastructure: Apache Kafka for message queuing and transaction stream simulation;
- Monitoring and Feedback: Grafana and Prometheus for metric visualization and performance tracking.

All modules were containerized using Docker to allow reproducibility and easy deployment. The overall orchestration was managed through Docker Compose.

Data Flow and Integration. Historical Data Pipeline: Raw historical data was ingested, cleaned, and preprocessed in batches. Feature engineering and label assignment were performed during this phase. Offline models were trained and periodically re-evaluated [4; 7].

- Streaming Pipeline: Real-time transaction data was published to Kafka topics. The online module subscribed to these topics, processed each transaction using lightweight models, and generated fraud scores with minimal latency.
- Feedback Loop: A monitoring service tracked system performance metrics and concept drift signals. Upon detection of significant drift or degradation in metrics, retraining was triggered via the offline module [5; 8].

Deployment and Testing Environment. The system was deployed and tested on a local server with the following specifications:

- CPU: 16-core AMD Ryzen;
- RAM: 64 GB;
- OS: Ubuntu 22.04 LTS;
- Environment: Python 3.10.

For performance testing, streaming data was injected at varying rates (100–1000 transactions/sec) to simulate realistic high-load environments. The system demonstrated robust real-time processing capabilities and adaptive retraining within tolerable time limits.

Integration with Business Logic. To ensure compatibility with existing financial transaction systems, a RESTful API interface was developed for scoring incoming transactions and receiving feedback from manual reviews. This API acted as a bridge between the ML models and operational decision engines.

Reproducibility and Extensibility. The implementation was designed to be modular and extensible:

- Each component (offline trainer, online detector, drift monitor) is independently deployable and replaceable;
- Configurations (e.g., scoring thresholds, drift limits, cost parameters) are externalized for tuning;
- The entire system is documented and version-controlled, facilitating future improvements and academic replication [6; 9].

Illustrative Overview of the System Workflow. The following diagram provides a visual representation of the end-to-end data flow in the proposed hybrid fraud detection architecture. It illustrates how the offline and online modules interact through a shared monitoring and feedback mechanism. The schema highlights three key pipelines: historical batch processing, real-time stream processing, and the integrated feedback loop responsible for retraining and alerting. Each component is modular, allowing flexibility for deployment and future scaling.

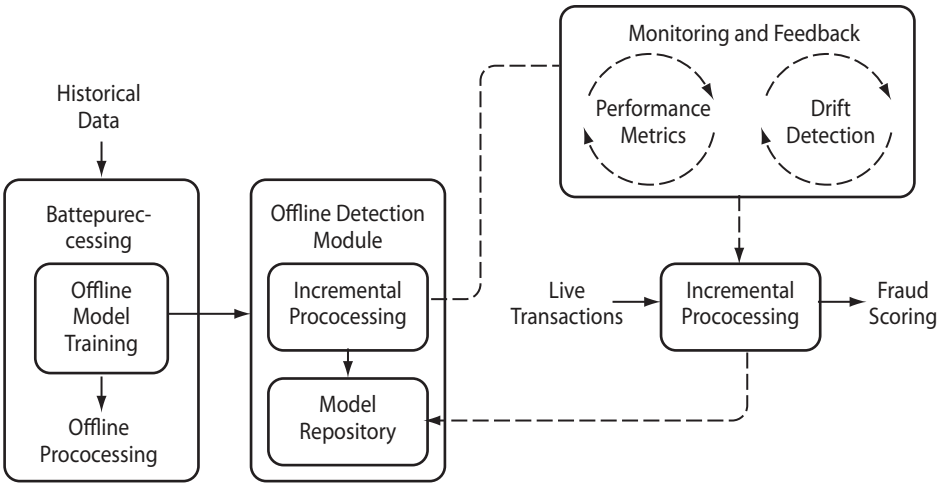

Fig. 3. End-to-end data flow for the hybrid fraud detection system

Source: elaborated by the author

The diagram shows the integration of historical data ingestion, offline model training, real-time transaction scoring, and feedback-based drift adaptation across all system modules.

Results. This section presents the performance outcomes of the proposed hybrid fraud detection system, based on extensive experimentation in both offline and online environments. The analysis covers multiple dimensions, including predictive accuracy, computational efficiency, drift adaptability, and cost-effectiveness – ensuring a comprehensive evaluation of the system under realistic conditions.

Offline Model Performance. Offline experiments were conducted using stratified 10-fold cross-validation on historical transaction datasets. The best-performing models – XGBoost, LightGBM, and Deep Neural Networks (DNNs) – demonstrated high classification accuracy, robustness, and effective handling of class imbalance:

- XGBoost: F1-score = 0.921, AUC-ROC = 0.986
- LightGBM: F1-score = 0.915, AUC-ROC = 0.982
- DNN: F1-score = 0.907, AUC-ROC = 0.976

These results underline the capacity of gradient boosting algorithms to capture non-linear fraud patterns and maintain

high discriminative power. Hyperparameter tuning and class weighting further enhanced model performance in the imbalanced data setting.

Online Model Performance. The online detection module was evaluated using simulated transaction streams with injected concept drift. This testing environment allowed for dynamic analysis of real-time performance and adaptability. Among the tested models, Adaptive Random Forests and Hoeffding Trees achieved the best balance between detection accuracy and response latency:

- Adaptive Random Forest: F1-score = 0.894, Response Time < 150 ms
- Hoeffding Tree: F1-score = 0.873, Response Time < 100 ms
- Online Perceptron: F1-score = 0.835, Response Time < 50 ms

These models maintained stable detection rates even when confronted with changes in feature distributions, confirming their suitability for streaming data scenarios [3].

Concept Drift Adaptability. The system's concept drift detection mechanisms effectively identified performance degradation by monitoring shifts in feature distributions and prediction error patterns. The custom Δ(t) indicator, in conjunction with drift-sensitive metrics, triggered retraining routines promptly. The retraining process restored model performance with minimal downtime and resulted in an average F1-score improvement of 6–9% post-drift, depending on the severity and type of drift introduced [4; 5; 7; 8].

Cost-Effectiveness Analysis. A cost-sensitive evaluation framework was applied to reflect the financial implications of model decisions. Based on realistic assumptions (e.g., $10.000 per undetected fraud case, $100 per false positive alert), the total estimated cost for each model was computed:

- XGBoost incurred the lowest total cost, owing to its superior precision and recall.
- Adaptive Random Forest achieved the best cost-performance trade-off in the online module, offering reasonable accuracy with low latency and reduced false positives.

These findings support model selection not only from a technical standpoint but also from a risk-mitigation and operational cost perspective [1].

Overall Score Comparison. To facilitate model comparison across multiple dimensions, a composite score was computed by integrating the F1-score, 1-False Positive Rate (1-FPR), and prediction stability. The scores were normalized and weighted according to operational priorities:

- In the offline setting, XGBoost ranked highest overall.
- In the online streaming context, Adaptive Random Forest was the top performer.

This comparison reinforces the hybrid design principle: deploying specialized models tailored to their respective operational environments maximizes overall system performance.

Visualization of Results. Comparative visualizations were generated to illustrate model performance trade-offs. These include ROC curves, drift evolution plots, and cost-adjusted bar charts. The figures underscore the strategic value of the hybrid approach, especially in balancing detection accuracy, system responsiveness, and financial impact.

The experimental results confirm that the proposed architecture is well-equipped to detect fraudulent transactions under dynamic, imbalanced, and high-throughput conditions. The hybrid strategy not only ensures high predictive accuracy, but also supports scalable deployment, fast adaptation, and cost-aware decision-making in real-world scenarios.

The following figure provides a synthesized visual representation of the results obtained from the hybrid fraud detection system. It highlights key findings across offline and online evaluations, emphasizing performance metrics, model responsiveness under concept drift, and financial efficiency. This flowchart is designed to offer a quick yet comprehensive snapshot of the system's effectiveness across different testing dimensions.

This figure summarizes the main experimental results from both offline and online modules. It illustrates comparative model performances (F1-score, AUC, response time), the impact of adaptive retraining triggered by drift detection, and the relative cost-efficiency of models such as XGBoost and Adaptive Random Forest. The diagram visually reinforces the system's dual strengths: predictive power and operational practicality in high-volume, real-time environments.

Discussion. The results of the hybrid fraud detection architecture highlight several important insights regarding both technical performance and practical implications for financial institutions.

Performance Trade-offs and Observations. The hybrid system achieved strong results across key performance indicators, particularly in F1-score, cost reduction, and adaptability to concept drift. Offline models such as XGBoost delivered high predictive accuracy, effectively capturing complex fraud patterns in imbalanced datasets. Meanwhile, online models like Adaptive Random Forests provided fast reaction times and maintained stability in streaming environments affected by data drift.

These findings confirm the complementary strengths of the dual-model design: while offline learners offer deep learning capabilities, online learners ensure real-time responsiveness. However, some trade-offs were evident. Online models tended to exhibit lower overall accuracy due to limited learning of intricate feature interactions. Offline models, although more accurate, required periodic retraining and lacked real-time adaptation. The hybrid approach mitigated these limitations by combining long-term learning with short-term agility.

Real-World Integration Challenges. From a deployment perspective, integrating the proposed architecture into existing financial infrastructures revealed several operational challenges. These included latency sensitivity, compatibility with legacy systems via APIs, and the automation of feedback loops. Fine-tuning thresholds for concept drift detection and defining retraining triggers were particularly crucial for maintaining system responsiveness without generating excessive false positives.

Furthermore, model interpretability emerged as a key concern among stakeholders. Ensemble-based models, though powerful, are often viewed as opaque or "black box" systems. To address this, the implementation incorporated risk scoring formulas and feature attribution tools (e.g., SHAP values), which helped increase transparency and stakeholder confidence in model outputs.

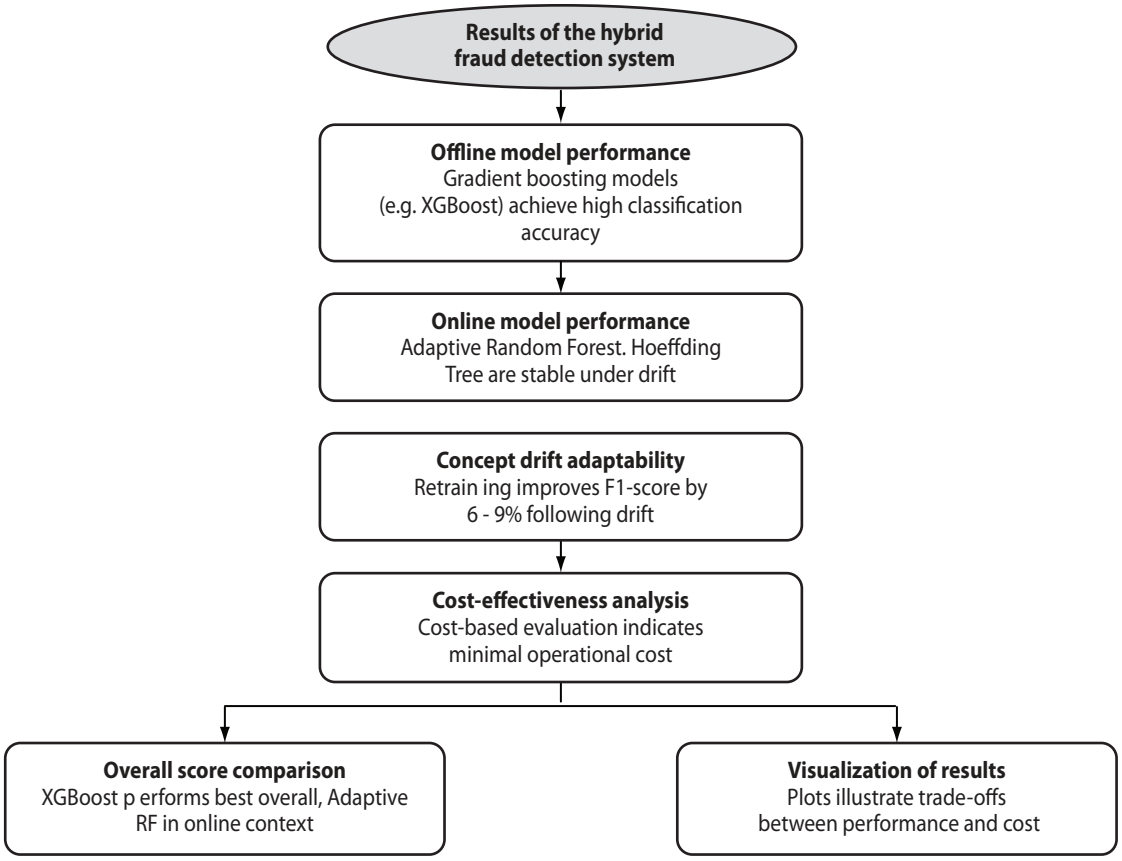

Fig. 4. Results of the Hybrid Fraud Detection System

Source: elaborated by the author

Economic and Operational Impact. The cost-sensitive evaluation framework provided clear evidence of the system's economic value. By minimizing false negatives –typically associated with significant financial losses – the hybrid model substantially reduced total operational costs. The use of cost-adjusted metrics also allowed informed model selection based on business risk thresholds and tolerance.

Additionally, the system's adaptability to evolving fraud behaviors reduced the burden of manual transaction verification and improved decision-making speed. This responsiveness to change is critical in maintaining long-term detection efficacy in adversarial environments.

Limitations and Areas for Improvement. Despite its strengths, the architecture has several limitations. First, the experiments used partially synthetic datasets to simulate large-scale transaction flows, which may not fully capture the complexity of real-world fraud patterns and adversarial behavior. Second, the feedback loop was primarily reactive – triggered by degradation in performance metrics. Future work could explore more proactive learning mechanisms, such as reinforcement learning or active learning, to anticipate fraud patterns earlier.

Moreover, although the system is designed to be modular and easily deployable, production-level implementation in regulated financial environments would require further validation, security audits, and compliance testing.

Contributions and Future Outlook. This study contributes a scalable, modular, and economically viable framework for fraud detection, combining batch-trained models with streaming detectors. The hybrid architecture provides a realistic path forward for financial institutions that require both accuracy and adaptability in high-volume, dynamic environments.

Looking ahead, future research could investigate federated learning approaches to enhance data privacy across institutions or integrate graph-based anomaly detection techniques for capturing relational fraud. Additionally, integrating explainable AI (XAI) techniques more deeply into the architecture could further improve model interpretability and regulatory compliance.

Ultimately, the proposed system underscores the importance of adaptive, interpretable, and cost-sensitive AI systems in high-stakes domains such as financial fraud, paving the way for intelligent, trustworthy, and human-centered decision support.

The following diagram (Figure 5) summarizes the key insights drawn from the hybrid fraud detection framework. It visualizes how performance, cost-effectiveness, adaptability, and future potential converge to support a real-world, scalable deployment strategy.

This figure synthesizes the discussion's core elements, showcasing how technical strengths like concept drift resilience and offline-online synergy translate into operational value. It also highlights the trade-offs encountered, deployment lessons learned, and the roadmap for future enhancements.

Conclusion. This article proposes a hybrid and modular fraud detection architecture that integrates both offline learn-

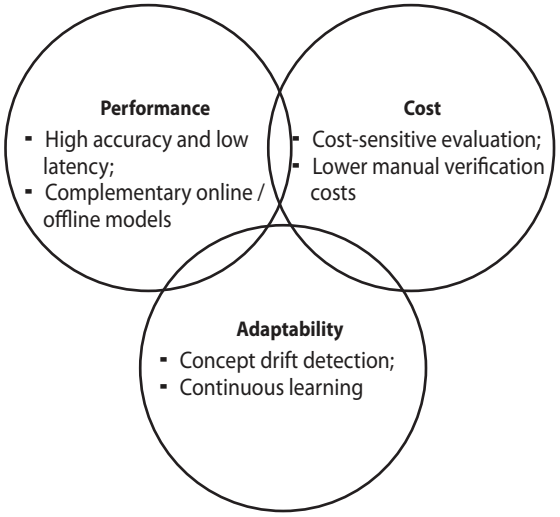

Fig. 5. Key Insights from the Hybrid Fraud Detection Architecture

Source: elaborated by the author

ing and real-time online adaptation. By combining high-performance offline models – such as XGBoost, LightGBM, and Deep Neural Networks (DNNs) – with adaptive, lightweight online learners – such as Hoeffding Trees and Adaptive Random Forests – the system effectively addresses the challenges of fraud detection in both historical and streaming transaction contexts.

A key methodological contribution of this study lies in its ability to balance predictive performance, responsiveness, and interpretability. The inclusion of a weighted risk scoring mechanism enhances decision transparency, while the unified cost-sensitive evaluation framework ensures alignment between technical metrics and real-world financial implications.

The architecture is designed with modularity and scalability in mind, enabling ongoing adaptation through concept drift detection and feedback-driven retraining. Implementation in a containerized, open-source environment ensured reproducibility and robustness under high-load simulations, facilitating seamless deployment in production-grade financial ecosystems.

In summary, the proposed framework offers a flexible, interpretable, and operationally viable solution to modern fraud detection. It bridges the gap between advanced machine learning research and real-world requirements, contributing a deployable system that evolves alongside fraud strategies.

Future research directions include integrating graph-based relational features for network fraud detection, applying reinforcement learning for adaptive decision optimization, and leveraging federated learning techniques to ensure data privacy across institutions.

The following diagram summarizes the core architecture and contributions of the proposed hybrid fraud detection system, visually consolidating its functional layers and operational flow.

The Figure 6 illustrates the complete architecture of the hybrid fraud detection framework developed in this study. The model integrates batch-based offline learning with real-

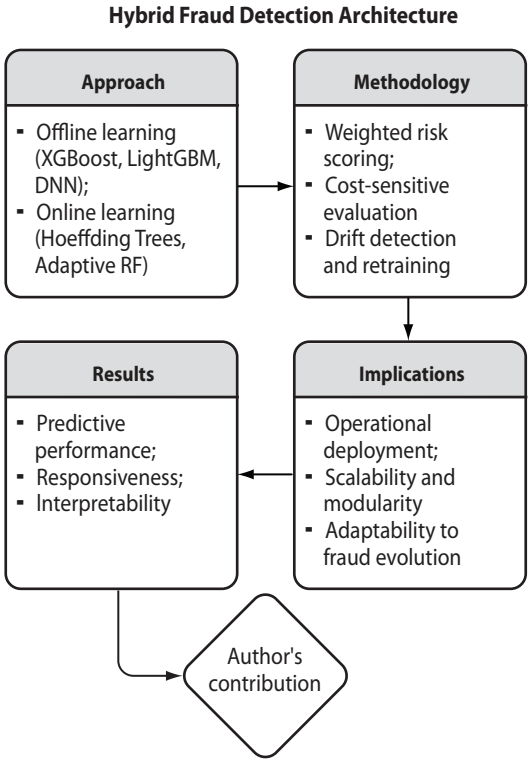

Fig. 6. Hybrid Fraud Detection System Overview

Source: elaborated by the author

time streaming detection to address modern fraud detection challenges. The author's key contributions reflected in this system are:

1. Design of a dual-layer architecture combining offline and online modules for improved adaptability and accuracy.
2. Integration of cost-sensitive evaluation metrics to align model performance with financial risk reduction.
3. Development of a feedback-driven retraining loop that supports concept drift detection and continuous learning.
4. Implementation of a modular and containerized infrastructure using open-source technologies for real-world deployment.
5. Improved interpretability through risk scoring and attribution mechanisms tailored to financial institutions' needs.

This hybrid system represents a pragmatic and scalable approach that bridges academic innovation with the operational demands of fraud prevention in dynamic, high-volume environments.

Author's Previous Work and Contributions. The author has previously explored the field of fraud detection using intelligent systems and machine learning techniques in several peer-reviewed publications. These works addressed critical issues such as model selection, AI interpretability, and the economic implications of false alarms in banking environments [11; 12]. The current study extends and consolidates these findings by proposing a unified hybrid framework that integrates both offline and online fraud detection mechanisms.

LITERATURE

1. Bahnsen A. C., Aouada D., Ottersten B. Example-dependent cost-sensitive decision trees. *Expert Systems with Applications*. 2016. Vol. 42. No. 19. P. 6609–6619.

DOI: https://doi.org/10.1016/j.eswa.2015.04.070

2. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD*. 2016. P. 785–794.

DOI: https://doi.org/10.1145/2939672.2939785

3. Montiel J., Read J., Bifet A., Abdessalem T. River: machine learning for streaming data in Python. *Journal of Machine Learning Research*. 2021. Vol. 22. No. 1. P. 1–9.

4. Gama J., Žliobaitė I., Bifet A., Pechenizkiy M., Bouchachia A. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*. 2014. Vol. 46. P. 44.

5. Žliobaitė I. Learning under concept drift: an overview. *arXiv preprint* arXiv:1010.4784. 2010.

6. Khammas B., Roushangar K. A hybrid offline–online architecture for adaptive fraud detection in online payments. *Expert Systems with Applications*. 2020. Vol. 159. 113620.

DOI: https://doi.org/10.1016/j.eswa.2020.113620

7. Bifet A., Holmes G., Pfahringer B., Kirkby R., Gavaldà R. New ensemble methods for evolving data streams. *Proceedings of the 15th ACM SIGKDD*. 2011. P. 139–148.

8. Liu Y., Li J., Yang Y., Yu L., Zhang C. A cost-sensitive ensemble method for fraud detection based on dynamic updating. *Information Sciences*. 2019. Vol. 476. P. 421–439.

9. Ribeiro M. T., Singh S., Guestrin C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD*. 2016. P. 1135–1144.

10. Breunig M. M., Kriegel H.-P., Ng R. T., Sander J. LOF: Identifying density-based local outliers. *ACM SIGMOD Record*. 2000. Vol. 29. No. 2. P. 93–104.

11. Caprian I. The Use of Machine Learning for the Purpose of Combating Bank Fraud. *Business Inform*. 2023. No. 7. P. 140–145.

DOI: https://doi.org/10.32983/2222-4459-2023-7-140-145

12. Caprian I. The Application of Artificial Intelligence for the Purpose of Combating Bank Fraud. *The Problems of Economy*. 2023. No. 2. P. 204–212.

DOI: https://doi.org/10.32983/2222-0712-2023-2-204-212

REFERENCES

Bahnsen, A. C., Aouada, D., Ottersten, B. (2016). Example-dependent cost-sensitive decision trees. *Expert Systems with Applications, 42*(19), 6609–6619. https://doi.org/10.1016/j.eswa.2015.04.070

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, R. (2011). New ensemble methods for evolving data streams. *Proceedings of the 15th ACM SIGKDD*, 139–148.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *ACM SIGMOD Record, 29*(2), 93–104.

Caprian, I. (2023). The Application of Artificial Intelligence for the Purpose of Combating Bank Fraud. *The Problems of Economy, 2*, 204–212. https://doi.org/10.32983/2222-0712-2023-2-204-212

Caprian, I. (2023). The Use of Machine Learning for the Purpose of Combating Bank Fraud. *Business Inform., 7*, 140–145. https://doi.org/10.32983/2222-4459-2023-7-140-145

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD*, 785–794. https://doi.org/10.1145/2939672.2939785

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys (CSUR), 46*, 44.

Khammas, B., & Roushangar, K. (2020). A hybrid offline–online architecture for adaptive fraud detection in online payments. *Expert Systems with Applications, 159*, 113620. https://doi.org/10.1016/j.eswa.2020.113620

Liu, Y., Li, J., Yang, Y., Yu, L., & Zhang, C. (2019). A cost-sensitive ensemble method for fraud detection based on dynamic updating. *Information Sciences, 476*, 421–439.

Montiel, J., Read, J., Bifet, A., & Abdessalem, T. (2021). River: machine learning for streaming data in Python. *Journal of Machine Learning Research, 22*(1), 1–9.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD*, 1135–1144.

Žliobaitė, I. (2010). *Learning under concept drift: an overview*. arXiv preprint arXiv:1010.4784.

Стаття надійшла до редакції 05.08.2025 р.

Статтю прийнято до публікації 22.08.2025 р.