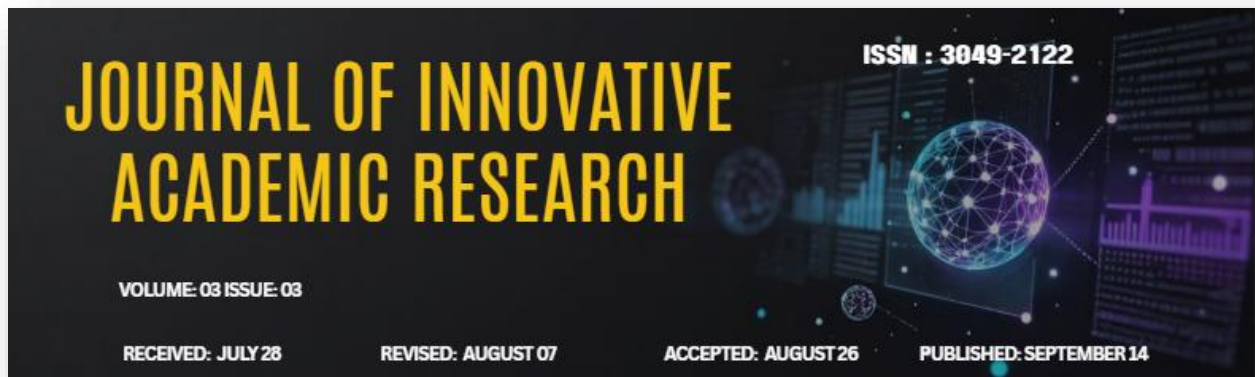




Deep Agents for Cloud-Native Fraud Detection in Insurance

Dimitris Plexousakis

Asst Professor

Abstract

Insurance fraud contributes greatly to policyholder losses and the yearly drain on industry resources. Existing deterrent methods typically react to fraud events, affecting genuine customers. An approach using Deep Agentic AI to sense abnormal activity and potential fraud is applied. The underlying principle is viewing fraud detection as an agentic activity, employing real-time inference with streaming data analytics to monitor for fraud risk. The proposed architecture integrates microservices with serverless components in a cloud-native design aligned with the features of true agentic AI. Infrastructure is established for autonomous decision-making and incident response, operationalizing real-time fraud detection at scale. Although investigated in fraud detection, the approach is generalizable and applicable across domains in which agentic decision-making is necessary and proven agentic systems are in place.

Detecting fraud in mutual insurance companies, where expenses are borne by policyholders, is globally important. Fraudulent pricing, accident claims, and pay-outs directly reduce profits. Sensitive agents have failed to establish deterrence, as is evident from the rising share of fraud cases. Detecting fraud pre-emptively offers the potential to limit client inconvenience and to avoid incorrect insurance decisions. Investors have indicated a desire for insurance companies to increase investment in AI, with almost three-quarters considering insurance fraud a suitable application. Therefore, the development of appropriate tooling will facilitate meeting investor expectations.

Keywords : Agentic AI; Cloud-Native Environments; Concept Drift; Fraud Detection; Insurance; Risk Prediction; Time-Series Data; Deep Agentic AI Systems; Real-Time Insurance Fraud Detection; Autonomous Decision-Making Agents; Risk Prediction and Scoring Models; Cloud-Native AI Architectures; Event-Driven Fraud Analytics; Multi-Agent Reinforcement Learning; Streaming Data Intelligence; Explainable AI (XAI) for Insurance; Scalable Mops for Financial Services.

1. Introduction

The explosive growth of digital payments globally and the accompanying complexity of financial services offer fertile ground for significant increases in technology-enabled fraud. AI has become an indispensable component of enterprise applications supporting dozens of business functions, including helping to detect transactions that have an unusually high probability of being fraudulent. Although the analytical components are now mature and operationalized in large banking institutions and payment processing

organizations', trustworthiness, governance, and scaling remain unresolved challenges.

This research introduces the concept of agentic AI and applies it to building a flexible and scalable cloud-native system to detect insurance fraud in real-time. An Agentic System is defined, and the properties of real-time inference on streaming data are described. Current cloud-native architectures supporting fraud detection either focus only on model performance, without considering operational aspects, or provide very coarse architectural guidelines that do not enable a production-ready, end-to-end cloud-native

implementation. The proposed architecture supports fraud detection for transactional and non-transactional events and provides the operational components expected of a production system. These components include governance of data quality, feature engineering, model training, testing, and deployment, together with transparency in risk prediction through uncertainty quantification by model calibration and drift.



Fig 1: Agentic AI for Insurance

1.1. Background and Significance

Insurance fraud exists in various forms, including false claims, non-disclosure, and misrepresentation of risk. The ever-changing nature of technology enables new fraudulent methods to emerge and the writing of cyber risk insurance exposes the industry to new challenges. Although complex to detect and comprehend, sufficient amounts of high-quality claim data can enable fraud to be detected and predicted. Various reasons make this an important area of research: first, the scale of the impact of fraud on individual companies, the industry as a whole, and society in general; second, insurance companies' perspectives of fraud; third, cyber insurance fraud can be classified as a new type of fraud that is executed in a dynamic environment; and last, the main ways to address such issues.

Real-time detection and prediction of fraud is an area of growing interest. Being able to perceive fraud-related situations across different environments and at any time is a key feature of such a system. An increasingly popular

approach uses data-driven techniques at its core, applying statistical techniques to detect previous unknown fraudulent activity. Recent methods using deep learning techniques in fraud detection have shown promising results that outperform traditional techniques in related problems. On the other hand, these methods are not suitable for real-time applications.

Equation 1: Real-time fraud scoring as probabilistic classification

1.1 Model output

Let each incoming event/claim be represented by features x (structured + unstructured, streaming ingestion).

A fraud model outputs a probability:

$$p(x) = \mathbb{P}(Y = 1 \mid x)$$

where:

- $Y = 1$ = fraud, $Y = 0$ = genuine.

1.2 Decision rule via thresholding

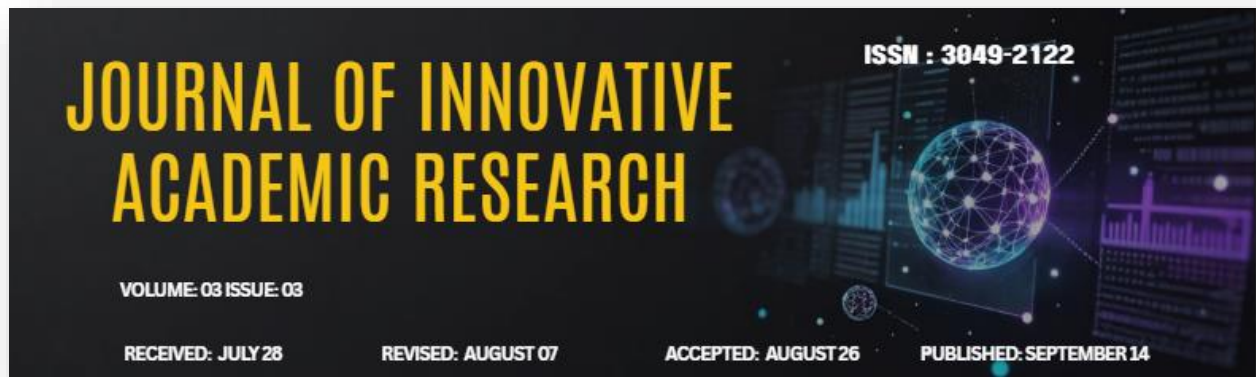
The simplest real-time “fraud score $\rightarrow$ action” rule is:

$$\hat{y}(x; t) = \begin{cases} 1 & \text{if } p(x) \geq t \\ 0 & \text{if } p(x) < t \end{cases}$$

where $t \in [0,1]$ is the operating threshold (often tuned to meet SLA + business risk appetite). This aligns with the paper’s “fraud score... assess new claims in real time” idea.

1.2. Research design

The presented work establishes and examines the real-time production deployment of two distinct capabilities based on agentic deep learning: fraud detection and risk prediction. The research validates theoretical and architectural assumptions surrounding agentic deep learning within cloud-native environments and involves a deployment at scale



using diverse real-world insurance datasets integrated into operational systems.

The theoretical foundations focus on how the concept of agentic systems can be mapped to fraud detection applications. An agentic deep neural network operates within a cloud-native environment, dynamically scaling compute resources based on streaming data volume. Deep learning is applied as well to risk-prediction tasks, where uncertainty quantification, adaptation to concept drift, and risk-sensitivity calibration are addressed. Concepts and implementations associated with risk prediction are presented, with a particular focus on data-governance aspects of the cloud-native deployment for real-time applications.

2. Theoretical Foundations of Agentic AI in Fraud Detection

Agentic Systems and Autonomy

Agentic systems are characterized by a specific notion of agency. A system is said to exhibit agency if it can decide how to act when faced with a set of options, even when it is otherwise incapable of acting independently. This means that the system can establish contingencies and select a course of action based on possible future events. Alex Huth's work in neuroscience demonstrates that an agentic system can infer logical relations that define simpler roles: a dog and a cat, for instance, would behave similarly when meeting a rabbit but differently when faced with a bear. Alex Huth's work for physics and neuroscience considers proximity to expected future events. This idea of agency can also be drawn from the ongoing developments in quantum physics. Systems that feature quantum-level correlations are capable of making decisions that are outside the capabilities of their components. Framing agency in terms of decisions opens up its adoption for any systems that optimize the future probability of achieving goal states.

It is important to differentiate between agentic systems and their architectures. Agentic capabilities can come into play when reasoning tasks are defined over the system, and it can find its way into larger systems. The application to fraud detection can emerge through the combination of Cloud Data Warehouses, Serverless Computing components, and a selection of model architectures. The Self-Organizing Maps with Continuous Reinforcement Learning can supply the necessary notions of autonomy. Agentic capabilities focus on achieving control while solving decision-making problems. Autonomy emerges when control ceases to be an external requirement and becomes part of the problem. The language game addresses control in the sense of achieving the transition to one state or another.

Real-Time Inference and Streaming Analytics

Real-time inference refers to the ability of a model or system to perform predictions on new instances in a continuous manner, integrating new information on-the-fly. This notion aims to satisfy the operational business requirements of real-time fraud detection, such as providing all decisions within the SLA defined as part of operational service-level agreements. The real-time aspect is achieved through a Cloud-Native architecture designed for streaming analytics.

The Cloud-Native architecture is based on microservice principles using characteristics of serverless computing. The data-architectural aspect focuses on data ingestion and processing tasks supporting the fraud-detection models. The different phases of the ingestion pipeline have an attention-sensitive dimension defined according to business-value considerations.

2.1. Agentic Systems and Autonomy

Whether an agentic system is trustworthy and conservative enough for the current operating context is typically determined by its learning model. An agentic system behaves competitively with respect to its learning model and conservatively with respect to its operating model, switching to inference mode whenever the operating model is

sufficiently trustworthy. In fraud detection, this translates to the capability of an agentic system to dark-learn the penalty costs associated with false positives and false negatives, thereby answering an important question: for a given operating model of the world, how much penalty for fraud must be paid for the AI system to erode its trustworthiness? A fundamental aspect of such systems is that truthfully incorporating the penalty costs of making errors improves the real-world decision-making value of the AI outputs.

Confidence quantification models that go beyond traditional DL techniques within insurance data instances can illustrate and measure this mechanism in action. The uncertainty quantification module complements a standard DL architecture—like the error-imbalance-independent Deep Augmentation–Asymmetric Cost–Estimation Neural Network used for model training—but preserves the inherent bias-variance balance of the DL with training noise properties that break the typical deterministic behaviour of neural networks during inference. Moreover, the proposed uncertainty quantification module measures the decision-theoretic risk associated with trustworthiness during AI implementation by capitalizing on reformulating the Rodrigues Representation Theorem for Deep Augmentation Asymmetric Cost. It addresses the research question by determining the trustworthiness envelope of a model governing fraudulent activity.

Equation 2: Asymmetric cost / penalty learning (false positives vs false negatives)

2.1 Define cost matrix

Let:

- C_{FP} = cost of flagging a genuine claim (false positive)
- C_{FN} = cost of missing a fraud (false negative)

Costs per decision:

$$\text{Cost}(\hat{y}, y) = \begin{cases} 0 & \hat{y} = 0, y = 0(\text{TN}) \\ 0 & \hat{y} = 1, y = 1(\text{TP}) \\ C_{FP} & \hat{y} = 1, y = 0(\text{FP}) \\ C_{FN} & \hat{y} = 0, y = 1(\text{FN}) \end{cases}$$

2.2 Expected cost for a given threshold t

For a batch/window of streamed decisions (say in a minute), define counts $FP(t)$, $FN(t)$. Then:

$$\mathbb{E}[\text{TotalCost}(t)] = C_{FP} \cdot FP(t) + C_{FN} \cdot FN(t)$$

This is exactly what the “**Expected decision cost vs threshold**” graph (shown above) demonstrates.

2.3 Deriving the *Bayes-optimal* decision boundary (per-instance)

Instead of tuning by sweep, you can derive the per-instance rule.

Predict fraud (choose $\hat{y} = 1$) if the expected cost of predicting 1 is lower than predicting 0:

- If choose $\hat{y} = 1$: cost occurs only when $y = 0$:

$$\mathbb{E}[\text{Cost} \mid \hat{y} = 1, x] = C_{FP} \cdot \mathbb{P}(y = 0 \mid x) = C_{FP}(1 - p(x))$$

- If choose $\hat{y} = 0$: cost occurs only when $y = 1$:

$$\mathbb{E}[\text{Cost} \mid \hat{y} = 0, x] = C_{FN} \cdot \mathbb{P}(y = 1 \mid x) = C_{FN}p(x)$$

Choose fraud if:

$$C_{FP}(1 - p(x)) < C_{FN}p(x)$$

Solve step-by-step:

1. Expand left:

$$C_{FP} - C_{FP}p(x) < C_{FN}p(x)$$

2. Bring $p(x)$ terms to one side:

$$C_{FP} < (C_{FN} + C_{FP}) p(x)$$

3. Divide:

$$p(x) > \frac{C_{FP}}{C_{FP} + C_{FN}}$$

So, the **cost-optimal threshold** (under calibrated probabilities) is:

$$t^* = \frac{C_{FP}}{C_{FP} + C_{FN}}$$

That is the cleanest mathematical expression of the article's "penalty costs" concept.

2.2. Real-Time Inference and Streaming Analytics

Fraud detection in insurance is a recurring theme in knowledge databases. Research efforts predominantly address predictions based on batch processing paradigms except where operationalization takes precedence. Integrating prediction and streaming analytics into a production-grade agentic AI system for real-time detection remains an open challenge. Data locality and policies restrict information sharing among companies. Thus, prediction quality is susceptible to exposure settings. Uncertainty quantification, calibration, and concept drift adaptation encompass critical elements of agentic-system deployment.

Agentic AI embodies autonomous systems capable of choosing optimal actions based on continuous state representations. Autonomy comprises perception, planning, and action without manual intervention. Fraud mitigation demands a complex approach. Indicators must be interpreted in conjunction, hence the integration of detection and risk prediction. Based on corresponding conditions, timely actions trigger alerts, enable further investigation, or escalate to fraud detection.

3. Cloud-Native Architectures for Real-Time Fraud Detection

Microservice architectures—composed of loosely coupled, organized web services that use containerization for scalable deployment and dynamic orchestration—are now predominant in cloud environments, and many ML components increasingly use serverless infrastructure. Cloud-native fraud detection supports these ML components, managing streaming events such as insurance claims and policies as they are generated. These event streams are copied and handled by a set of microservices specialized for compliance policies of security, privacy, and confidentiality. Important aspects include regionalizing data, implementing risk-averse models, and enabling secure data sharing among insurance partners and competitor consortiums. A regulatory-as-a-service technology offers a way to easily integrate compliance policies with streaming fraud detection services. Fraud detection services can thereby become responsive, supporting incident response and decision-making as needed for credible-case signatures and case management.

Modern digital natives generate large amounts of data from third-party platforms. These data streams must be cleaned, transformed, and processed. Cloud-native architectures and serverless event-driven components ensure that fraud detection services—supporting decision, incident response, and fraud case management—adapt and respond to data generation and fraud attack signatures. Reusable components, developed for different types of fraud attacks and risk assessment, cut development efforts. The architecture scales in accordance with service demand and event volume using cloud-native microservice and serverless paradigms, implementing compliance services—notably region data localization, privacy and security policies, and secure data sharing on both source and destination.

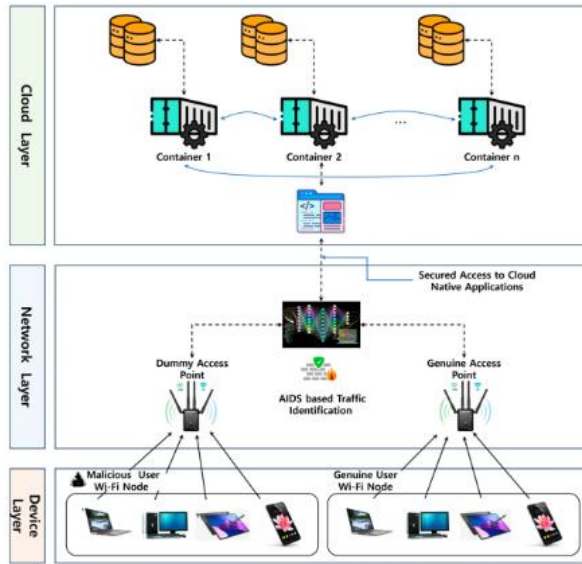


Fig 2: Cloud-Native Architectures for Real-Time Fraud Detection

3.1. Microservices and Serverless Components Cloud-native environments provide general-purpose cloud platforms with built-in machine learning offerings, running-business processing applications and support mainly through microservices hosted in containers or as serverless components. Fraud detection in insurance is also supported by many cloud providers with a demand-driven scalability option, enabling real-time processing of burst requests. Streams of incoming transactions, claims and other fraud-related data from multiple sources, including claims and policy systems, law enforcement agencies, third-party vendors and social media, are processed by a combination of serverless and microservice components. Serverless functions perform ingestion and aggregation of raw textual data; microservices consume the pre-processed information for model-based prediction; and the predicted results are inserted back into cloud databases for further investigation or action.

The auxiliary data feature space is a broad collection of attributes ingested from external sources and continually updated for consumption by machine-learning models. Information consumed in a given time window is typically probabilistic in nature, reflecting the uncertainty behind its acquisition. Suitable methods are applied for quantifying this uncertainty and assessing the impact on the model output, in order to support real-time decision making on fraud detection and risk prediction. Since insurance markets and processes are inherently complex and affected by constant regulatory changes, the generated prediction support systems must also accommodate concept drift in model functionalities.

Equation 3: Training loss with asymmetric costs (cost-sensitive learning)

If you train a neural classifier with probabilities $p_{\theta}(x)$, standard binary cross-entropy is:

$$\mathcal{L}_{\text{BCE}}(\theta) = - \sum_{i=1}^N (y_i \log p_{\theta}(x_i) + (1 - y_i) \log(1 - p_{\theta}(x_i)))$$

Cost-sensitive / weighted BCE (to reflect C_{FN} and C_{FP}):

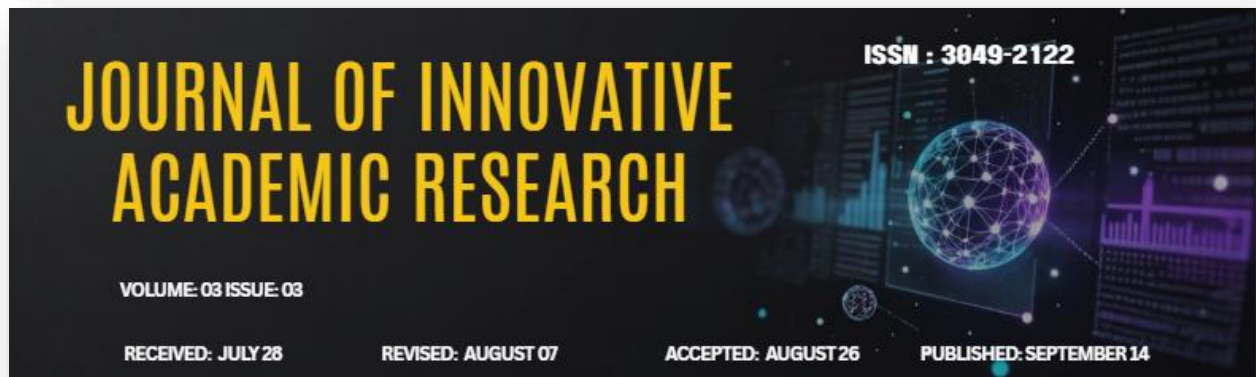
$$\mathcal{L}_{\text{WBCE}}(\theta) = - \sum_{i=1}^N (w_1 y_i \log p_{\theta}(x_i) + w_0 (1 - y_i) \log(1 - p_{\theta}(x_i)))$$

A common mapping is:

$$w_1 \propto C_{\text{FN}}, w_0 \propto C_{\text{FP}}$$

because:

- A positive example misclassified causes FN-type harm.
- A negative example misclassified causes FP-type harm.



This matches the paper's emphasis on incorporating penalty costs into learning/decision-making value.

3.2. Data Ingestion, Processing, and Compliance

Real-time insurance fraud detection requires both rapid data ingestion and compliance with various data-oriented regulations and guidelines. For compliance with European regulations on the processing and storage of personally identifiable information, sensitive information, and possible discrimination features, careful masking and anonymisation procedures are needed.

The architecture can support multiple incoming data streams from different sources. Incoming and processing services may therefore be orchestrated as a serverless mixture of microservices and event-driven components. Incoming data streams, originating for instance from web forms such as subscription requests, can be filtered by category and routed to a suitable processing pipeline. Such differentiation allows the introduction of privacy-preserving techniques, such as the encryption of personally identifiable and sensitive information or masking of features susceptible to discriminatory bias. These processing steps may be brought close to the data source, using dedicated functions or service units running in processes or containers respectively. Data from various sources can then be merged. Ingestion is supported for categorical features accommodating a multitude of possible values, as in the case of insurance claim texts. Incoming textual data may be classified before processing with models sensitive to the correct nature of the data. The architecture supports rapid model retraining to comply with possible concept drift in low-frequency, high-impact events such as floods or pandemics.

4. Model Development, Evaluation, and Deployment

The integration of AI applications for real-time fraud detection heightens the requirement of dependable prediction models. One prime reason is the public-sector authorization

and consequent requirement for a solid data governance wrapper that encompasses not only the fairness of the models but also ensures that the data-driven mathematical functions are interpretable and auditable. Several executable declarative data governance rules classify the data according to sensitive characteristics; these include gender, age groups, and claim location municipalities, and enable the identification of possible proxy discrimination. The set of features generated through the data governance and feature-engineering episodes are crucial in the construction of appropriate prediction models.

The concepts of real-time prediction maximization and actuated design of streaming predictive architecture apply not just to the fraud-detection prediction stage, but also naturally extend to the risk-calculation prediction of insurance business. The former considers ingestion, infrastructure, computing, and maintenance costs; the latter naturally sharpens the algorithmic explanation and transparency of insurance price-setting calculations. Application of such principles ensures that the model deployed satisfies accuracy and speed requirements while also forming a reliable base for further layers of intelligent processing.

4.1. Data Governance and Feature Engineering

Fittingly for an agentic system with a focus on real-time decisions, the architecture is designed for streaming inference on edges close to data generation. Features optimized for pragmatic consumption are paramount for true insurance practitioners to achieve agility during crises.

Data governance principles define high-level specifications for data detection readiness. Micro-architectures enable ingestion of a multitude of data points: telematics data from each car's sensor for real-time anomaly detection; cyber-intelligence information from external providers; event streams for catastrophe and crisis detection, such as climatic events or Covid-19 disease propagation; voice conversation interaction from the Call Centre, and hundreds of others triggered from the operational core. Feature engineering

techniques transform Big Data to Smart Data. The essential principle is that feature engineering should go beyond being merely a demanding phase of data science but rather should require as much, if not even more, expertise than actual novel data models. Only by releasing fast and ready-to-consume features can the data science process be accelerated and thus ignored by business experts.

4.2. Model Architectures for Real-Time Detection

The efficiency and accuracy of real-time insurance fraud detection hinge on the architecture of predictive models and the underlying data analytics. The first line of defence against illicit claims and suspicious behaviour is a simple fraud score, a lightweight prediction set to assess new claims in real time, derived from a straightforward model trained on historical data. Independence of the Teradata Causal Spotlight® system allows its free interpretation of business rules, so that predictions can serve as signals for amplifying alerts. Subsequently, a deep-learning model harnessing visual documentation in the claim process has been developed. Its centralized location enables easy integration in a serverless architecture, which targets API calls with image URLs and returns a risk score. The architecture is designed for detections on each claim's inspection, and during processing so that risks may be considered in the decision.

The fraud-detection solution focuses on a specific claim, assessing real-time data structured around that particular event. A combined transit microservice collates information from all business lines for each claim, no matter the type. This real-time-processing unit assembles tables leveraged by the NLP fraud-detection classifier, which detects emerging patterns in unstructured, textual data from external sources. Additional event-based NLP models, capturing perceptions of related events, are also optional. The streamlined design allows energy-efficient cloud operations while optimizing accuracy.

Equation 4: Uncertainty quantification (UQ) for “trustworthiness”

4.1 Predictive uncertainty decomposition (classification)

In Bayesian terms:

$$p(y | x, \mathcal{D}) = \int p(y | x, \theta) p(\theta | \mathcal{D}) d\theta$$

Practical approximation (MC Dropout / deep ensembles): sample T models (or dropout masks) $\theta^{(t)}$:

$$\hat{p}(x) = \frac{1}{T} \sum_{t=1}^T p_{\theta^{(t)}}(y = 1 | x)$$

A simple uncertainty proxy is the variance:

$$\widehat{\text{Var}}(p(x)) = \frac{1}{T-1} \sum_{t=1}^T (p_{\theta^{(t)}}(y = 1 | x) - \hat{p}(x))^2$$

This gives you a numerical “how stable is my probability?” which is one operational notion of trustworthiness.

4.2 Risk-aware decision using uncertainty (one clean way)

Let $\sigma(x)$ be a predictive uncertainty measure. A conservative agent can adjust threshold upward when uncertainty is high, e.g.:

$$t(x) = t^* + k \sigma(x)$$

so the model flags fraud only when both probability is high **and** uncertainty is low enough for the action policy.

That is consistent with the article’s “conservative operating model” framing.

5. Risk Prediction under Dynamic Environments

Risk-prediction models are often assumed to be static. However, environments change and forecasts usually

become uncertain. Quantifying prediction uncertainty is essential for understanding how much to trust the predictions. Since the introduction of deep learning, attempts to quantify uncertainty in deep neural networks have advanced considerably. Creating calibrated scalable classifiers using large amounts of data has also become possible, and such techniques make it feasible to create classifiers for risk prediction in terms of likelihoods based on containment times.

The insurance business environment changes continuously. Consequently, the ability to rapidly and effectively track changes in risk or vulnerability and quickly adjust prediction models is crucial. Various mechanisms to detect the occurrence of such concept drifts should be set up. After a change has been detected, the corresponding expert knowledge is needed to assess how prediction models should be adapted. Concept drift has been analyzed in various application domains, and detection techniques have been developed and reviewed for different areas such as computer vision, weather forecasting, and classification problems.

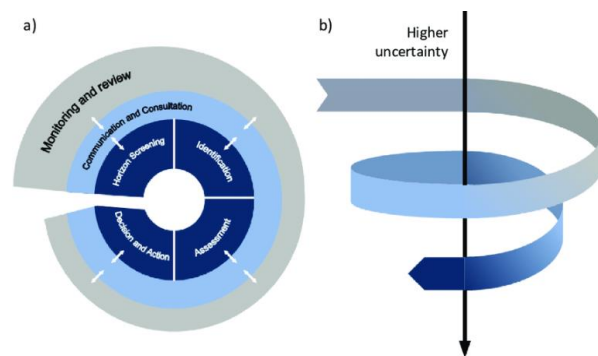


Fig 3: Dynamic risk management framework

5.1. Uncertainty Quantification and Calibration

Developing an optimal dynamic model for risk prediction interfaces the uncertainty that affects model predictions. Insurance risk prediction tasks, such as estimation of claim severity or hazard rate of new policies, often require

calibrated outputs for decision making. Insurance fraud detection is an evolving problem. Labelled training data must continuously be created from confirmed label changes, and the model must be periodically updated to account for changing terminology, evolving methods of perpetration and changing insurance products. An enabling technology for such continuous machine learning is concept drift adaptation, which allows learnt models to track changes in the data-generating process without the need to retrain from scratch on large cohorts of historical data.

Past studies have developed various architectures for concept drift adaptation. These structural templates rely on different mechanisms for detecting changes in the prediction function, and often employ either an ensemble or continual learning approach for incremental model updates. The combination of architecture templates and a given learning algorithm usually produces an effective solution for a specific dataset. Framework-level source code is therefore important to support proofs-of-concept with. driven detection at a Cloud-based operations environment.

Equation 5: Calibration (turning scores into trustworthy probabilities)

5.1 What calibration means

A model is calibrated if among all events where it predicts ~0.7, about 70% are truly fraud.

Formally:

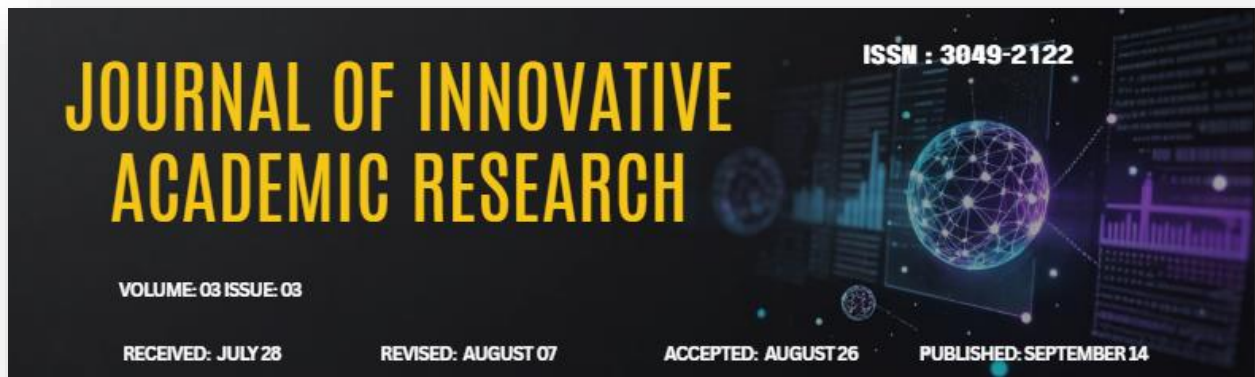
$$\mathbb{P}(Y = 1 \mid \hat{p}(X) = q) = q$$

5.2 Reliability diagram (what the plot above shows)

Partition predictions into bins B_m . For bin m :

- average predicted probability:

$$\text{conf}(m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}(x_i)$$



- empirical fraud rate:

$$\text{acc}(m) = \frac{1}{|B_m|} \sum_{i \in B_m} y_i$$

Plot points $(\text{conf}(m), \text{acc}(m))$. Perfect calibration lies on $y = x$ (the dashed diagonal in the graph).

5.3 Expected Calibration Error (ECE)

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{N} |\text{acc}(m) - \text{conf}(m)|$$

This is a single number you can monitor online in production.

5.2. Concept Drift Adaptation in Insurance Data

Concept Drift adaptation in digital records linked to insurance claims is non-trivial and remains only partially addressed. The heterogeneity of insurance services means that concepts in different types of policies and territories have diverse changing dynamics. Solving this issue usually entails having different risk prediction models for each of these types. However, a large proportion of claims come from travel insurance, and very few occur outside of peak seasons. In these cases, the temporal effect is strong enough to drive the recommendations, and a model using data within that temporal range would be considerably more accurate. A growing proportion of predictions in these situations would either be out of range (would recommend coverage and entail a resource consumption) or low risk (should recommend not purchasing coverage).

Learning a dedicated model focused on peak seasons generates a non-exploited opportunity in the fraud detection process. Evidence shows that concept drift on insurance digital records also exists. Models built on historical data not only require periodic re-training but also become less reliable when making predictions for fresh records. However, readily building and maintaining dedicated models

for every concept-drifted sub-cluster of the full dataset is not feasible in fraud detection since the number of flagged claims is typically small. The models get built on a periodic basis; therefore, drift may already have occurred when they get used in real operational environments.

6. Operationalizing Fraud Detection at Scale

A strategic location for operating a Deep Agentic AI for real-time insurance fraud detection is determined in order to facilitate an effective novel three-part decision-making interaction model that supports human-in-the-loop decision, incident response, and fraud risk management activities, along with the required operational processes. Such a system can be designed to streamline the handling of alert notifications issued during the fraud detection phase.

An incident response capability is realized as an initial responder system, in which the primary task is to triage alerts generated during the fraud detection phase of the digital insurance division by performing confirmatory testing on motorists identified as fraudulent, screen their claims for fraudulent activity, and adjudicate on reinstatements of detected fraudulent claims. The fraud case management function focuses on monitoring detected fraud cases and supporting investigations into fraud events detected by the pilot fraud function. A standard operating procedure has been developed to guide incident response, enabling incident responders to handle fraud alerts received from the Deep Agentic AI system efficiently and consistently.

6.1. Decision-Making Interfaces and Human-AI Interaction

AI systems that operate in safety-critical settings, such as fraud detection, need to be reliable and trustworthy. These requirements also extend to the decision-making interfaces that connect deep neural networks and humans. User interfaces must be designed for the target user group, users' risk tolerance must be considered, and deep networks'

decisions and underlying uncertainty should be fully transparent. In general, a human-AI hybrid interface strives for a synergy between explorable models (i.e., models that the users can interpret, question, and explore) and any AI-enhanced service that reduces the users' cognitive load without putting too much weight on the recommendation.

Hybrid interface designs rely on the complementary roles of humans and AI. AI is usually able to process large amounts of data quickly and deliver recommendations based on many aspects that a human might not be able to consider. On the contrary, humans are better equipped in complex and uncertain situations where the information prompting those decisions is often incomplete or inconsistent. Depending on the nature of the specific task and its associated risk, either the AI system or the human should be given the decision latitude.

6.2. Incident Response and Fraud Case Management

The link between abnormal situations driving AI-triggered alerts and decision-making processes is critical for operationalizing insurance fraud detection at scale. A human-in-the-loop fraud case management system is proposed to efficiently include an analyst (or team of analysts) in the loop of agents and significantly reduce risk to acceptable levels in these distributed agentic systems. Contribution of the untrusted human actor is naturally included in the concept drift adaptation process, relying on adaptation-outcome data being the decisions assigned by humans on real-time fraud detections.

A fraud case management UI was developed enabling operators to review a ready-to-action fraud alert report (or fraud case) and decide whether detected fraud should be confirmed or rejected. Changes to the SQL database are implemented through TWIL and incorporated into subsequent concept drift analysis. The UI was designed to permit multiple operators viewing concurrent fraud case reports.

Equation 6: Risk prediction: hazard rate (mentioned explicitly)

Let T be time-to-claim (or time-to-fraud/incident).

Survival function:

$$S(t) = \mathbb{P}(T > t)$$

Density $f(t)$, then hazard:

$$h(t) = \lim_{\Delta t \rightarrow 0} \frac{\mathbb{P}(t \leq T < t + \Delta t \mid T \geq t)}{\Delta t}$$

Derive hazard step-by-step:

1. Conditional probability:

$$\mathbb{P}(t \leq T < t + \Delta t \mid T \geq t) = \frac{\mathbb{P}(t \leq T < t + \Delta t)}{\mathbb{P}(T \geq t)}$$

2. Numerator $\sim f(t)\Delta t$, denominator $S(t)$:

$$\approx \frac{f(t)\Delta t}{S(t)}$$

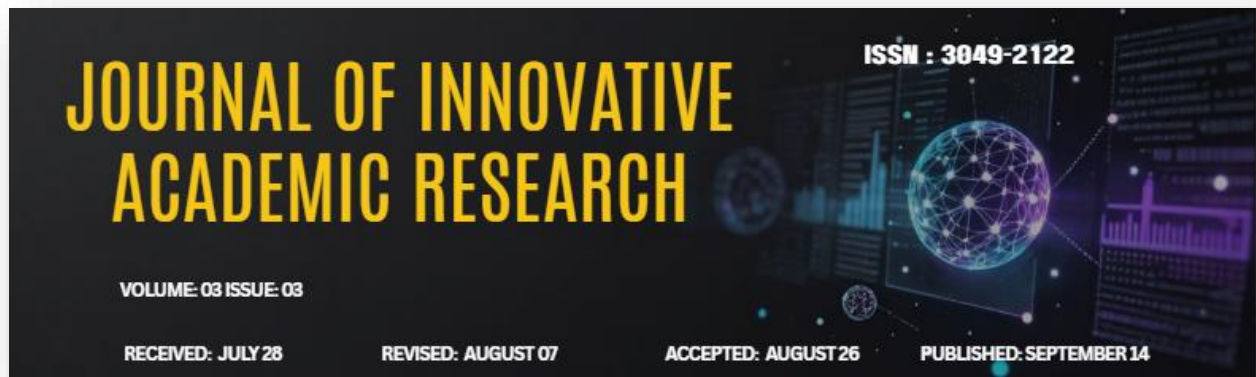
3. Divide by Δt , take limit:

$$h(t) = \frac{f(t)}{S(t)}$$

This connects “risk likelihood over containment times” to a principled equation.

7. Conclusion

The development, deployment, and operationalization of real-time fraud detection and risk prediction in insurance companies contribute significantly to the efficiency of the related companies. The feasibility and performance of the operationalization of the fraud detection system aided by a



deep agentic component support the directions of research development and implementation. In addition, the rapid growth and wide application of agentic AI open new opportunities for improving business value.

The future trends of agentic AI in fraud detection can be classified into three categories. First, due to the increasing amount of data produced by the continuous insurance policy and claim processes, it is necessary for agentic AI functions to be developed for business domains other than fraud detection. These include insurance risk prediction and the implementation of other use-case-specific systems. Second, the benefits of agentic AI-supported model deployment with a decision-making interface should also be investigated in other machine learning tasks. Third, with the growing importance of it for improving business value and the increasing availability of open-source projects, deep agentic AI is expected to gain importance in the fraud detection field.

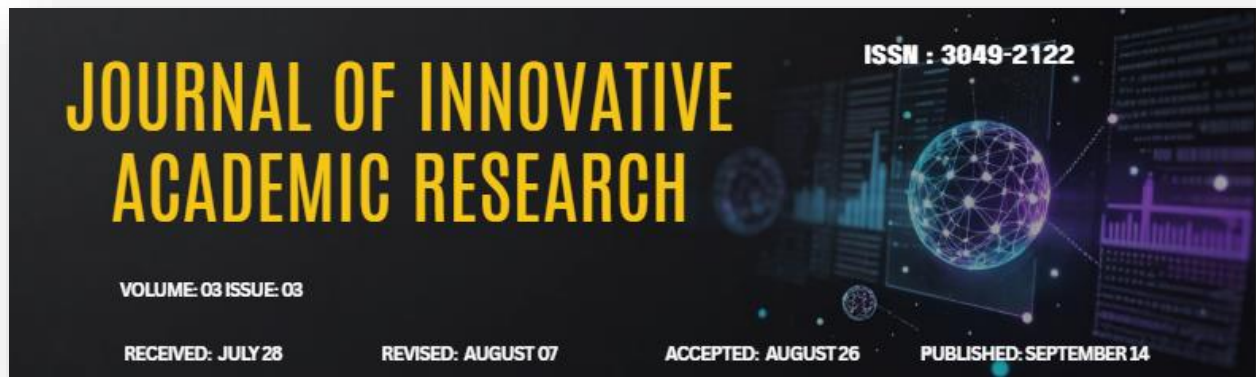
7.1. Future Trends

Today's agentic systems are primarily crafted around a distinct task or narrowly defined domain. The research underpinning real-time insurance fraud detection is no exception, featuring models that target fraud detection at a single time point. However, the real world is constantly changing. Data sources do not always remain active and contain the same type of data, yet the risk remains that data leaking at another location at another time could still show fraudulent behaviour. Accidents happen, processes change, new technologies come to market, and Konjunkturzyklen changes. As a result, the need for a general-purpose risk prediction service emerges. Is an incident risky or not, and why? What types of incidents are regarded as risky? Can the system learn also from the incidents that are not labelled as risky by humans and yet regarded to belong to a risk area by the model? These are the questions such a service should be able to address, and with the correct setup, it can.

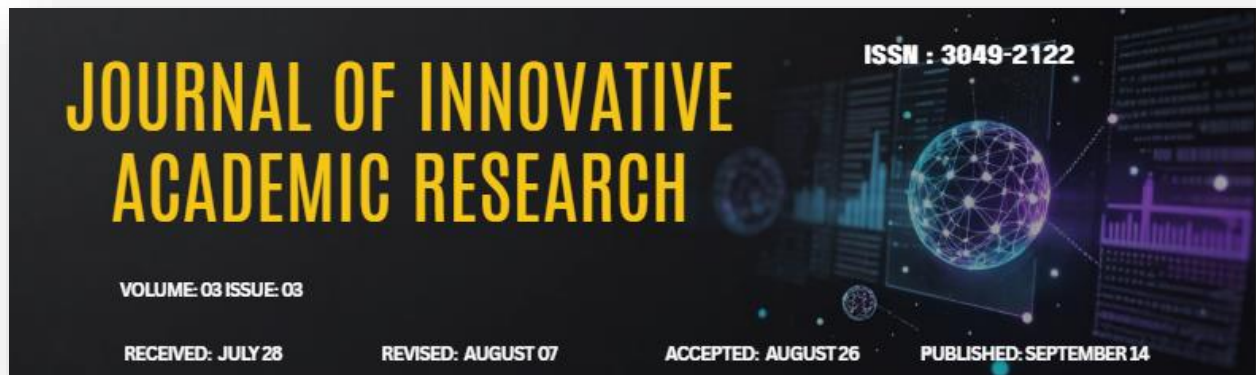
Calibrated uncertainty quantification is desired both to account for newly emerging classes of incidents as they flow through the fraud detection service and for supporting risk prediction in domains without a priori labels on risk. Feedback loops are required in both directions. Automatic detection of concept drift under uncertainty and an online ensemble model for mitigating label deficiency are thus additional components in the service stack.

8. References

1. Al-Hashedi, K. G., & Magalingam, P. (2021). Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Computer Science Review*, 40, 100402.
2. Błaszczyński, J., de Almeida Filho, A. T., Matuszyk, A., Szeląg, M., & Słowiński, R. (2021). Auto loan fraud detection using dominance-based rough set approach versus machine learning methods. *Expert Systems with Applications*, 163, 113740.
3. Gomes, C., Jin, Z., & Yang, H. (2021). Insurance fraud detection with unsupervised deep learning. *Journal of Risk and Insurance*, 88(3), 591–624.
4. Mangalampalli, B. M. (2024). Transparent Intelligence Explainability Frameworks for AI-Driven Clinical Decision Support in Healthcare Business Intelligence. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)*, 7(3), 10566-10579.
5. Hilal, W., Gadsden, S. A., & Yawney, J. (2022). Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Systems with Applications*, 193, 116429.
6. Aslam, F., Hunjra, A. I., Ftiti, Z., Louhichi, W., & Shams, T. (2022). Insurance fraud detection: Evidence from artificial intelligence and machine learning. *Research in International Business and Finance*, 62, 101744.



7. Ali, A., Abd Razak, S., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M., Elhassan, T., Elshafie, H., & Saif, A. (2022). Financial fraud detection based on machine learning: A systematic literature review. *Applied Sciences*, 12(19), 9637.
8. Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., & Ahmed, M. (2022). Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access*, 10, 39700–39715.
9. Alsuwailem, A. A. S., Salem, E., & Saudagar, A. K. J. (2023). Performance of different machine learning algorithms in detecting financial fraud. *Computational Economics*, 62, 1631–1667.
10. Mao, X., Liu, M., & Wang, Y. (2022). Using GNN to detect financial fraud based on the related party transactions network. *Procedia Computer Science*, 214, 351–358.
11. Van Belle, R., Van Damme, C., Tytgat, H., & De Weerd, J. (2022). Inductive graph representation learning for fraud detection. *Expert Systems with Applications*, 193, 116463.
12. Lin, K., & Gao, Y. (2022). Model interpretability of financial fraud detection by group SHAP. *Expert Systems with Applications*, 210, 118354.
13. Debener, J., Heinke, V., & Kriebel, J. (2023). Detecting insurance fraud using supervised and unsupervised machine learning. *Journal of Risk and Insurance*, 90, 743–768.
14. Aravind, R., & Surabhi, S. N. R. D. (2024). Smart charging: AI solutions for efficient battery power management in automotive applications. *Educational Administration: Theory and Practice*, 30(5), 14257–1467.
15. Banulescu-Radu, D., & Yankol-Schalck, M. (2024). Practical guideline to efficiently detect insurance fraud in the era of machine learning: A household insurance case. *Journal of Risk and Insurance*, 91, 867–913.
16. Zhou, Y., Li, H., Xiao, Z., & Qiu, J. (2023). A user-centered explainable artificial intelligence approach for financial fraud detection. *Finance Research Letters*, 58, 104309.
17. Aftabi, S. Z., Ahmadi, A., & Farzi, S. (2023). Fraud detection in financial statements using data mining and GAN models. *Expert Systems with Applications*, 227, 120144.
18. Fanai, H., & Abbasimehr, H. (2023). A novel combined approach based on deep autoencoder and deep classifiers for credit card fraud detection. *Expert Systems with Applications*, 217, 119562.
19. Chen, Z.-Y., & Han, D. (2023). Detecting corporate financial fraud via two-stage mapping in joint temporal and financial feature domain. *Expert Systems with Applications*, 217, 119559.
20. Yi, Z., Cao, X., Pu, X., Wu, Y., Chen, Z., Khan, A. T., Francis, A., & Li, S. (2023). Fraud detection in capital markets: A novel machine learning approach. *Expert Systems with Applications*, 231, 120760.
21. Motie, S., & Raahemi, B. (2024). Financial fraud detection using graph neural networks: A systematic review. *Expert Systems with Applications*, 240, 122156.
22. Tang, Y., & Liang, Y. (2024). Credit card fraud detection based on federated graph learning. *Expert Systems with Applications*, 256, 124979.
23. Abdul Salam, M., Fouad, K. M., Elbably, D. L., & Elsayed, S. M. (2024). Federated learning model for credit card fraud detection with data balancing techniques. *Neural Computing and Applications*, 36, 6231–6256.
24. Reddy, V. V. K., Reddy, R. V. K., Munaga, M. S. K., Karnam, B., Maddila, S. K., & Kolli, C. S. (2024). Deep learning-based credit card fraud detection in federated learning. *Expert Systems with Applications*, 255, 124493.
25. Cai, S., & Xie, Z. (2024). Explainable fraud detection of financial statement data driven by two-layer knowledge graph. *Expert Systems with Applications*, 246, 123126.
26. Koç, O., Başer, F., & Kestel, A. S. (2024). Clustering based fuzzy classification with a noise cluster in detecting fraud in insurance. *Applied Soft Computing*, 167, 112430.



27. Yandamuri, U. S. (2024). AI-Driven Decision Support Systems for Operational Optimization in Hospitality Technology. Metallurgical and Materials Engineering.
28. Vanini, P., Rossi, S., Zvizdic, E., & Domenig, T. (2023). Online payment fraud: From anomaly detection to risk management. Financial Innovation, 9, 66.