DevOps-Enabled Agentic Deep Learning for Insurance Fraud Intelligence

Dhanaraj Sathiri
Independent Researcher
dhanrajsathiri@gmail.com

Abstract

The study investigates DevOps-driven data engineering for agentic AI in insurance fraud detection with a focus on cloud-native automations. Data are centrally secured, accessible, governed, of reproducible quality, properly profiled, and efficiently prepared. Design, development, and deployment encompass extensive safeguards for risk-sensitive decision-making. A superset of continuous integration/continuous deployment tailored to machine learning as well as data governance underlie the approach, demystifying the technology stack and enabling success at operational scale. The investigation is motivated by regulatory, economic, and ethical pressures on the industry, demand for predictive solutions, and pain points of legacy monitoring systems. Automation of machine-learning model development and deployment mitigates issues of scalability, reproducibility, and security while incorporating risk assessment.

Organizations worldwide are increasingly turning to artificial intelligence in a bid to detect fraudulent activity in all its forms. Such systems, however, generally stem from one-off proof-of-concept initiatives, operate in isolation, and lack sufficient controls for operational deployment in support of everyday decision-making. Nevertheless, agentic artificial intelligence—actant solutions capable of executing decisions without direct human intervention—promises a transformative competitive advantage for business leaders. Integrating governance and control ensures that these automated systems operate within an acceptable risk tolerance, align with organizational values, and minimize unintended consequences. Insurance fraud detection solutions typically fall short of these requirements.

Keywords: DevOps Driven Data Engineering, Agentic Artificial Intelligence, Insurance Fraud Detection, Cloud Native Automation, Machine Learning Operations, Continuous Integration Continuous Deployment, Data Governance Frameworks, Risk Sensitive Decision Making, Automated Model Deployment, Predictive Fraud Analytics, Scalable AI Systems, Reproducible Data Pipelines, Secure Data Architectures, Regulatory Compliance In Insurance, Ethical AI Governance, Legacy System Modernization, Autonomous Decision Systems, Operational AI Control, Model Lifecycle Management, Enterprise AI Adoption.

1. Introduction

Insurance fraud, encompassing deliberate misreporting of claims, is a costly global problem that erodes corporate earnings and increases annual customer expenses. The prevention of fraudulent activity requires constant vigilance by insurance companies, which dedicate substantial resources to fraud detection. However, the sheer volume of claims makes manual assessment prohibitively expensive. Consequently, organizations deploy analytical models capable of detecting suspicious and potentially fraudulent claims for further investigation—ideally using enterprise data lakes that integrate multiple sources of structured and unstructured information and external databases. Machine

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 22 ACCEPTED: MARCH 02 PUBLISHED: MARCH 27

Learning (ML) processes are well-suited to this task, as they can analyze the information within, derive predictive patterns, and then automatically apply those patterns to future claims.

Nevertheless, the application of ML models to support fraud detection remains constrained by a range of factors. Data and ML development remain mainly siloed, thus preventing stakeholder-driven, responsive development, while produce-scand-operate processes remain disjointed rather than collaborative and continuous. The resulting tools lack transparent governance and control, leading to security concerns and limiting their deployment in production systems; the empirical evidence for the payoff from such AI still remains patchy. Hence, the models cannot be deployed as services, scaling the analytic capability of the enterprise, or rapidly operationalized via pipelines that enable continuous integration/continuous delivery (CI/CD) through data.

1.1. Objectives and Scope of the Study

The primary goal of this study is to explore, develop, and validate a cloud-native DevOps-automated deep learning system to detect fraudulent insurance claims. While the agentic AI perspective offers the promise of scaling machine learning (ML) model training, monitoring, and evaluation, the principal emphasis here is on how the combination of DevOps and certain elements of agentic AI can bring transparency, security, governance, and auditability to the data preparation and model development life cycles. Additional issues under the DevOps umbrella—reproducibility, continuous integration/continuous deployment (CI/CD), supply-chain security, and quality—are also significant but less critical. Nevertheless, testing and evaluation across nearly all these dimensions will maximize usefulness and reliability.

Research activity in the insurance sector has long focused on fraud, reflecting the extensive financial losses incurred around the world. Despite the gradual implementation of machine learning solutions along the fraud-detection process, the application of state-of-the-art (SOTA) deep learning models remains limited. The main drivers of this gap are the complexity of deep learning networks and the difficulties involved in data preparation. Nevertheless, interest in insurance fraud detection persists, and recent developments in data engineering enabled by a combination of cloud-native DevOps and SOTA models can enhance scalability, reproducibility, and security while providing the infrastructure required for the deployment of agentic AI. Such developments could therefore increase the attractiveness of deep learning, closing the gap between established and emerging methods.

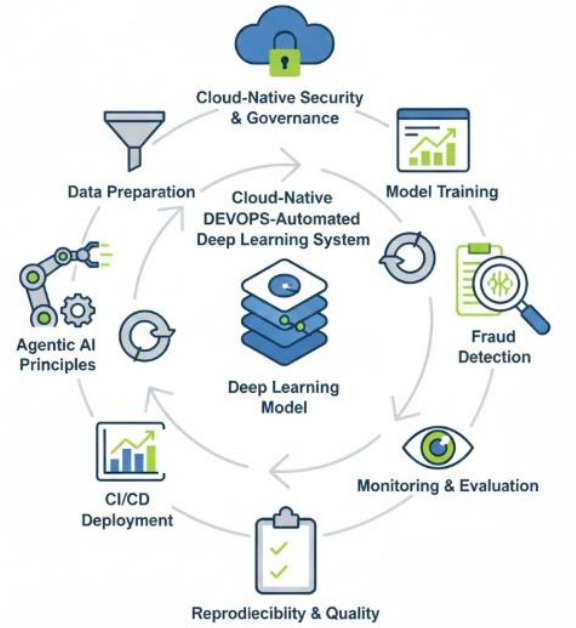


Fig 1: Scaling Deep Learning for Fraud Detection: A Cloud-Native DevOps Framework for Automated Governance and Agentic AI Integration

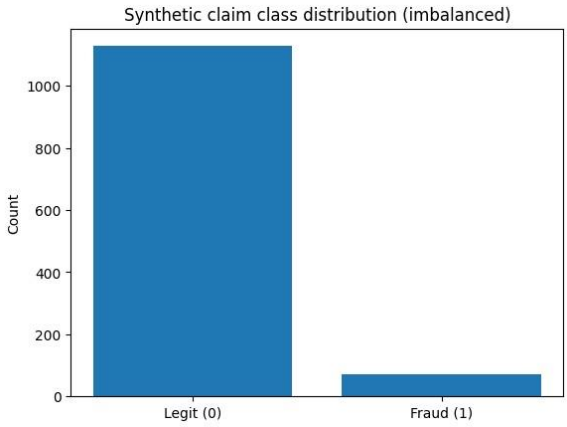
2. Background and Motivation

Detecting fraud during insurance transactions has grown into a pressing concern for the industry. Reports from several countries confirm the growth of fraudulent transactions, together with a lack of transparency of the underlying data. Countries such as the Netherlands have introduced regulatory measures to tackle these concerns;

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 22 ACCEPTED: MARCH 02 PUBLISHED: MARCH 27

such measures also increase pressure on the insurance industry to invest in machine learning solutions that support transparency and compliance while improving detection performance. While recent years have seen a substantial rise in interest in the research community, associations and industry, most efforts are still in a proof-of-concept stage – few, if any, results have made it into production.

The growing importance of fraud detection as part of the insurance value chain creates additional pressures: developing models that work at scale using industry-grade machine learning practices. Combining the principles established by DevOps for Data Engineering with industry-graded development and deployment considerations allows the creation of solutions that address these drivers. Scalability of the developed and integrated solutions appears to be the term that has been leant on most heavily; fulfilling the additional requirements of compliance, governance, transparency and reproducibility is a clear by-product of the DevOps-for-Data-Engineering principles. Beyond this, it is expected that the underlying business problems in the case of insurance transaction fraud detection can be sufficiently addressed; using the performance of the models in the support of insurance companies and associations can be a key source of motivation.



Equation 1: Data preparation equations

1.1 Min–Max normalization (MinMaxScaler)

Let:

- original feature value = x
- minimum of the feature column = $x_{\min}$
- maximum of the feature column = $x_{\max}$

Goal: map $x \in [x_{\min}, x_{\max}]$ to $\hat{x} \in [0,1]$.

Step-by-step derivation

1. Shift so the minimum becomes 0

   $x - x_{\min}$

2. Scale so the maximum becomes 1
   The max shifted value is $x_{\max} - x_{\min}$. Divide by that:

   $\hat{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$

3. (General range $[a, b]$ if needed)
   First compute $\hat{x} \in [0,1]$ as above, then stretch/shift:

   $x' = a + (b - a)\hat{x} = a + (b - a)\frac{x - x_{\min}}{x_{\max} - x_{\min}}$

2.1. Key Drivers of Research and Innovation

Scaling, reproducibility, security, and governance are becoming increasingly relevant for deploying ML solutions in production. The insurance sector is facing continuous pressure to deliver models that address key business questions and support operational decisions. However, the industry has yet to embrace innovation at scale, often relying on traditional technologies, bespoke approaches, and partly redundant ML initiatives that are rarely integrated into business processes. Pressure for innovation is mounting in insurance companies and new insurance players are entering the market, forcing the insurance industry to compete through continued innovation. Agentic AI comprises continuous integration and delivery principles applied to ML workflows, enabling the creation

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 22 ACCEPTED: MARCH 02 PUBLISHED: MARCH 27

of automated, monitored pipelines to regularly produce and deploy new ML models. Regulatory requirements such as the General Data Protection Regulation are forcing insurance companies to implement stricter data governance processes. Data ownership, access management, data quality, data classification, and data lineage tooling are becoming core data engineering issues. Insurance companies can leverage modernization and cloud services to integrate security into the design and transpose cloud-native principles and patterns into data engineering and ML process automation.

Class	Meaning	Count	Share
0	Legitimate Claims	1129	94.1%
1	Fraudulent Claims	71	5.9%

3. Theoretical Framework

DevOps principles provide valuable underpinnings for scalable, reproducible data engineering. Agentic AI strives for intelligent self-management of complex systems. Together, these ideas structure the research and guide the design of cloud-native, deep-learning-enabled software that addresses longstanding pain points in insurance fraud detection.

Continuous integration/continuous deployment (CI/CD) for machine learning, data governance, automated monitoring, and ethical considerations shape the investigation. Cloud-native processing meshes streaming and batch data, engenders a data-automation layer that supports ingestion, access, security, quality, lineage, profiling, and transformation, and provides a second automation layer for cloud-native ML pipelines.

Deep-learning-automation-as-a-service solutions easeproof-of-concept work by growing the four Tenet-specific pains and supporting three more—lack of production-ready ML, reproducibility, and security—before being taken up by the business-facing ML crew. Service teams outside the O&M squad own related service performance, stability, and security at service-data pipelines that form the dumb enablers (for safety, et al.).

CI/CD for ML and agentic AI's Monitoring tenets telescope to deliverproof of concept, O&M grows support for six company Operations pain points, and the master plan then steers operationalization of automated FaaS and ML pipelines. The latter deal with workloads Dw and Dd—crafting, hosting, and automating Thessala, probably doing the same for any approved MLOps job-lot as proof of concept, and supporting by touch-related-en)))) and Terraform/Docker-for-Dummies transports.

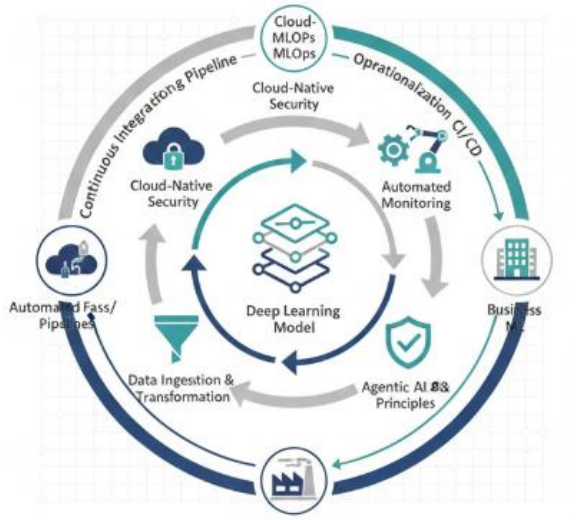


Fig 2: Synergizing DevOps and Agentic AI: A Cloud-Native Architectural Framework for Scalable Deep Learning and Automated Fraud Mitigation

3.1. Conceptual Overview of Key Principles

Continuous Integration/Continuous Deployment of Machine Learning

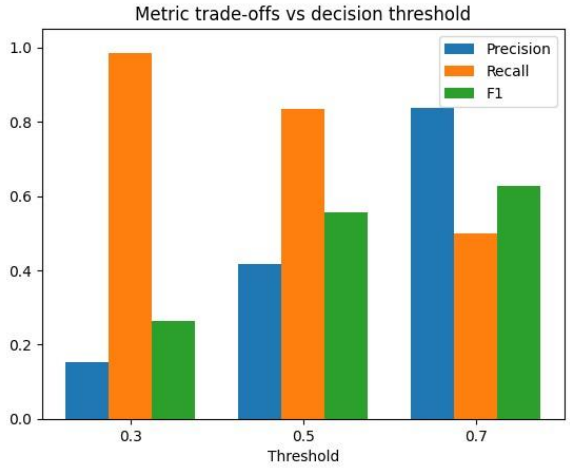
Although DevOps was conceived for software development, its principles of continuous integration/continuous deployment (CI/CD) can also be applied to machine learning (ML) models to enable the timely availability of high-quality algorithms. CI/CD focuses on three distinct phases: a) continuous integration, which includes the development, testing, and validation of

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 22 ACCEPTED: MARCH 02 PUBLISHED: MARCH 27

ML pipelines; b) continuous deployment of a candidate model that is updated regularly; and c) continuous training, which acknowledges that the underlying conditions affect the performance of deployed models. This requires an end-to-end ML pipeline (e.g. data ingestion, cleaning, preparation, modeling) to be in place, which, besides automating the modeling process, allows the management of model accuracy and robustness throughout the entire model development and production lifecycle.

Data Governance

Data are the lifeblood of successful ML solutions, and specific governance strategies, tailored to the scale and complexity of the organization, are needed to ensure that data is always available, secure, and compliant with regulatory requirements. Data governance comprises the design and execution of policies, standards, and processes that enable the acquisition and use of data, guarantee confidentiality, availability, and integrity, and mitigate relevant risks, such as security breaches, data leaks, and regulatory noncompliance. Effective data governance provides an inventory of the data assets of an enterprise, including metadata such as data classification, schema, access rights, lineage, source, quality metrics, documentation, and contact information, to ensure data consistency, security, and quality. It creates a shared understanding of data assets, minimizes risks, and generates a cycle of ongoing improvement.



4. System Architecture

The architecture provides a high-level view of the solution, detailing components, data flows, and integration patterns. Each process involved in insurance fraud detection is framed as a pipeline. The choice for cloud-native deployment is justified, and the multi-layer automation keeps complexity in check and security considerations addressed.

A. Data Ingestion Pipelines

Data ingestion covers the collection, access, lineage tracking, profiling, and quality assurance of data in separate processes for batch and streaming feeds. Access is controlled for both data producers and consumers according to governance principles. For batch loads, point-in-time access is provided (with snapshots enabled) and correctness ensured with integrity checks. A quality pipeline checks for records exceeding profile quality thresholds. Data sources include client lists (ground truth), transactional databases, claims history, user sign-up profiles, and social media postings.

B. Cloud-Native ML Pipelines

ML training is also organized as a set of pipelines. It comprises readiness checks, orchestration, containerization, training, monitoring, model registry, and CI/CD elements and supports the online production of models for direct

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 22 ACCEPTED: MARCH 02 PUBLISHED: MARCH 27

consumption by the business. Readiness checks verify the secure setup of cloud resources, design specifications, and data availability. The orchestration component handles resource provisioning and model training. In terms of model reproducibility, a container image is prepared before training—containing the environment libraries, the model training code with hyperparameter definitions, and the model storage bucket. The scan module periodically verifies the training setup based on CI principles.

Threshold	TP	FP	FN	TN	Precision	Recall (TPR)	F1 Score	FPR
0.30	70	392	1	737	0.15	0.99	0.26	0.35
0.50	59	83	12	1046	0.42	0.83	0.56	0.07
0.70	36	7	35	1122	0.84	0.51	0.63	0.01

4.1. Data Ingestion and Governance

Ingestion pipelines ensure quality, governance, and security, delivering trustworthy datasets for consumption. A Batch Loading Zone (BLZ) maintains structured data accessible for business users and analysts. The Cleaning Data Repository (CDR) holds refined datasets ready for machine learning. Stream and batch ingestions differ for incremental fraud monitoring and periodic training data updates. Batch operations handle files staged in the Cloud Storage Landing Zone. A comprehensive audit trail, established via DataFlow, tracks every operational aspect. Automated data quality checks, covering schema validation, completeness, accuracy, timeliness, and consistency, are vital for archived fraud detection data. For sensitive data, tools like Double-Check create a secondary copy with schema masking, ensuring secure development/testing access. Governance mechanisms enable appropriate authorizations at every stage. Data egress employs Data Transfer Service with fpp-level control, and Data-Lake-Information-Climate exposes additional technical metadata (e.g., data quality scores).

Equation 2: Outlier capping at three standard deviations

Let:

- dataset values: $x_1, \dots, x_n$
- mean:

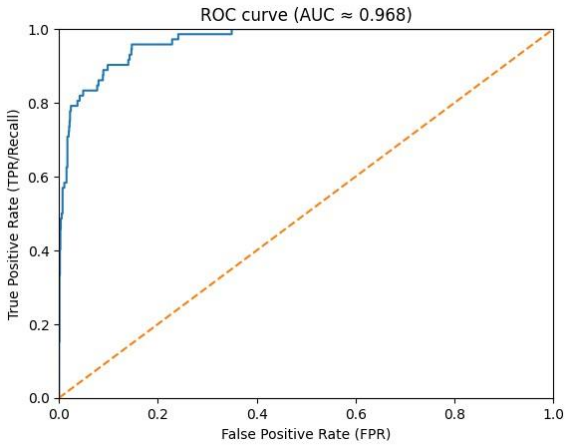
  $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
- population standard deviation:

  $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$
- bounds:

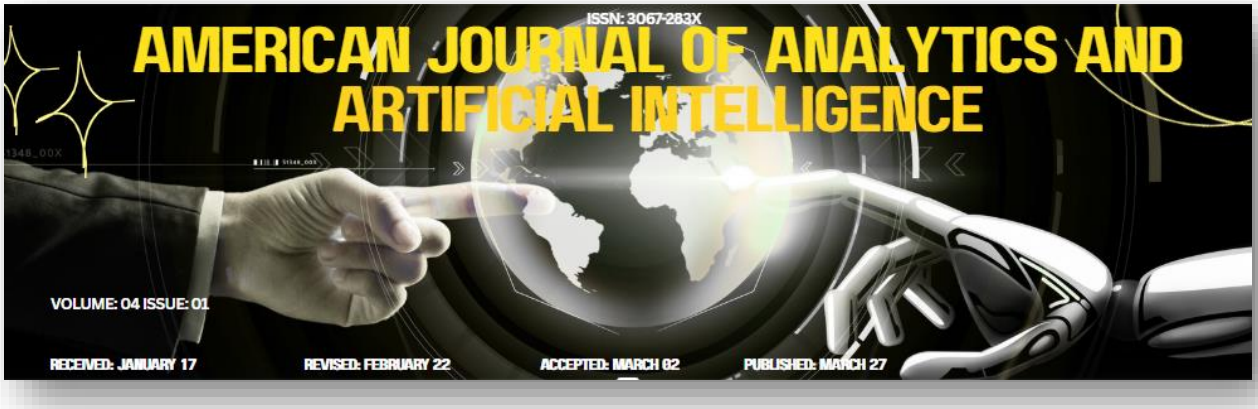
  $L = \mu - 3\sigma, \quad U = \mu + 3\sigma$

Capping rule (piecewise)

$$x^{(cap)} = \begin{cases} L, & x < L \\ x, & L \le x \le U \\ U, & x > U \end{cases}$$



4.2. Cloud-Native Machine Learning Pipelines



Pipelines ingesting the data into the machine learning (ML) module enable both model training and model orchestration at inference time. ML candidate models are deployed to a Google Cloud Vertex AI model registry where they can be evaluated, versioned, and monitored. Containerization enables new model versions to be generated in CI/CD for ML style, leveraging the full power of cloud-native continuous integration pipelines. New models can be trained with new hyperparameter searches when new training data is available, or when a new candidate architecture needs to be explored from among a shortlist. As the training process is encapsulated in a container, failure to train, timeout, or over-consumption of resources are easily preventable within limits automatically enforced by the orchestration system. Quality of training and production candidate models is enabled by automating the processes of building datasets, data and model profiling, exploring trained models, and discovering and evaluating hyperparameter search spaces.

Automated pipelines of operational characteristics indicated previously for data ingestion can also be used to prepare datasets ready for training or retraining of a fraud detection setup. Careful control of the relational algebra underlying these preparations underpins the constant availability of datasets suitable for high quality reproduction of fraud detection candidate models. A separate process regularly evaluates chessboard comparison matrices of candidate fraud detection model performance on representative datasets. Such processes help to remove unsuitable candidate models from the registry before reuse at inference time, augmenting the checks that should be in place by any ML Ops module. These candidate models are selected primarily by F1 score as precision and recall are commonly known to have traditionally opposing forces in the fraud domain.

5. Data Preparation and Feature Engineering

Extensive data preparation is necessary to transform in-house and publicly available datasets into appropriate model training artifacts. Apart from data cleaning and the normalization of categorical and continuous features, privacy-sensitive personal identifiers such as names, dates of birth, and addresses are removed. Normalization of sensitive features indirectly strengthens the governance around models exposed via APIs. Data from the skewed distribution of fraudulent cases must be selected carefully. Thus, synthetic samples that resemble the fraud detection process of insurance claim requests are added, and noise is injected into normal claims, enhancing robustness to overfitting. While the model considers simple features such as claim amount and policy number in its decisions, the Rewards and Fraud scenarios discussed earlier can signal the occurrence of higher fraud propensity in a broader set of features with disparate scales.

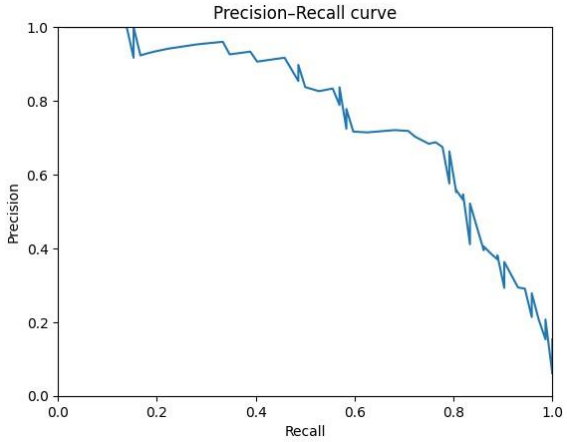
Data cleaning begins with the removal of duplicates, previously expelled claims from the combination of a unique policy number and claim amount, and all-zero feature vectors. The long and lat categorical features are then transformed into geographic location categorical features since they do not hold any semantic information for the detection process. Afterward, time dependency for key features is also injected. Finally, one-hot encoding is employed for categorical features with more than thirty unique values. Normalization is subsequently performed using MinMaxScaler, which scales the values to a specified range (defaulting to 0, 1) while preserving base intervals. For other features with minor pI, outliers are capped, i.e., values beyond three standard deviations are replaced with the closest in-bound value.

Feature selection narrows down from thirty potential features to the following five indicators of insurance fraud propensity: digits in policy number, elapsed days since policy was taken, elapsed days since claim was lodged, last claim amount made on the policy, and time since last claim. Resampling is then performed to address class imbalance challenges during model training. The min-max scaling approach described earlier is used, injecting noise via the following function:

main,9933$nelvalbitinewtjtgstqitunlirumrUfatitiswwrufatittqwsjtbitwrinitnflutnmRflutratutwRflutdittnflutnitutatwj5wd9ttsrgklZflutnitnitwgl$

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 22 ACCEPTED: MARCH 02 PUBLISHED: MARCH 27

ഇദ്ദേഹമു. Finally, after the garbage in–garbage out principle prompts the inclusion of multiple feature profiles, feature profiling and characterization across the entire data preparation process ensure data quality and forecasting capacity for machine learning models.



5.1. Data Cleaning and Transformation Techniques

The data preparation workflow relies on a combination of data-catalog services, ETL orchestrators, and user-developed scripts or notebooks. The first step focuses on the cleansing of the InsuranceCompany Fraud Detection Dataset (ICFDD) (Kaggle 2021) prior to being used in the training, testing, and validation of machine learning models. After almost 1900 records of missing values were removed, the categorical attributes (underwriting, route, make, type, and fraud) had their values transformed into one-hot variables to allow for proper processing by machine learning algorithms. The fraud value was inversely coded via bitwise not and then multiplied by −1 to prepare the dataset for plotting of the Fraud Detection Fractal, thus producing a stronger contrast between fraud and nonfraud patterns. Continuous, discrete, and categorical attributes were then analysed using K-means profiling in order to leverage the data's natural structure for future model training.

A separate data preparation workflow conforms the ICFDD for ingestion into an NLP-based fraud detection pipeline. User-defined Python scripts perform additional cleaning duties, including removal of URL and HTML coding information that could otherwise interfere with the NLP engine's predictive capability. Normalization transforms any anomalous or redundant strings into a common format and part-of-speech tagging supplies structure to every word token in a description to facilitate intelligent selection of target language for the NLP engine. Word-count detection further identifies portions of the comments made by an insurance agent that may be inadequate for high-quality translation into the report's target language.

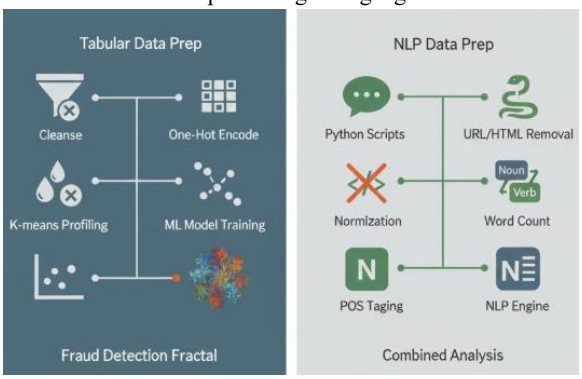


Fig 3: Bimodal Data Engineering for Insurance Fraud Detection: Integrating Fractal-Encoded Numerical Processing and NLP-Based Linguistic Structuring

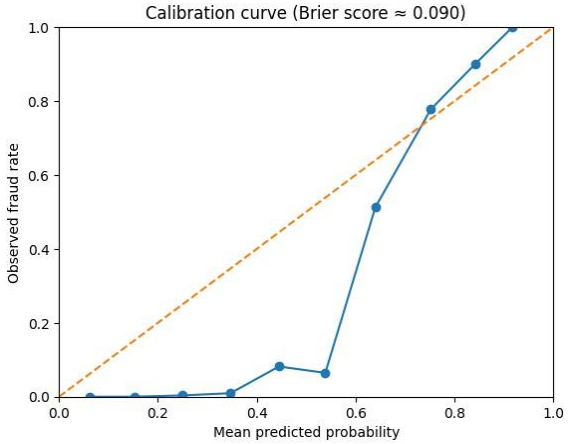
6. Model Development and Evaluation

The model development for attack detection is conducted in three high-level stages. The first stage involves the exploration of various deep learning architectures and the selection of promising candidate models suitable for classifying fraud. The second stage covers the definition of training protocols for the identified candidates, an assessment of potential overfitting issues, and the identification of appropriate evaluation metrics as well as a validation plan. The third stage includes a concrete plan for the deployment of the best-performing model into continuous operation.

While deep learning discriminative classifiers are the core candidate architectures, some foundational supervised models have been added and will be evaluated to better

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 22 ACCEPTED: MARCH 02 PUBLISHED: MARCH 27

understand the data and to provide performance benchmarks. In addition to convolutional networks for images, long short-term memory models for sequences, and spatiotemporal architectures (convolutional long short-term memory networks), it is also planned to explore less common unsupervised / self-supervised / adversarial approaches: generalised autoencoders, GANs and VAEs for both image and time series data anomalies. Predictive deep-fake models and GANs in particular might also indicate impression-manipulating sophistication or socially-aware generation capability. Other less-well-explored strategies like knowledge-distilled models – operationally efficient classifiers derived from knowledge-embodied and heavy transformers – will also be trialled on dataset subsets. Evaluation metrics will include standard classification measures (precision, recall, F1-score, ROC-AUC, Brier score) but with added sensitivity to the criticality of false negatives. For instance, a relatively high false-positive-to-false-negative cost ratio could justify maximising the false-negative-and-missed-class rate instead of minimising them. Models will also be cross-validated using multiple techniques: standard k-fold, hold-out retroactive sets and time-respecting splits. Attention-based architectures will undergo attention-visualisation techniques to uncover which parts of the data are being used for classification. Sensitivity and explainability of supervised algorithms will be studied to understand what features they are using to identify fraud rapidly and which are “distracting” them during inference.



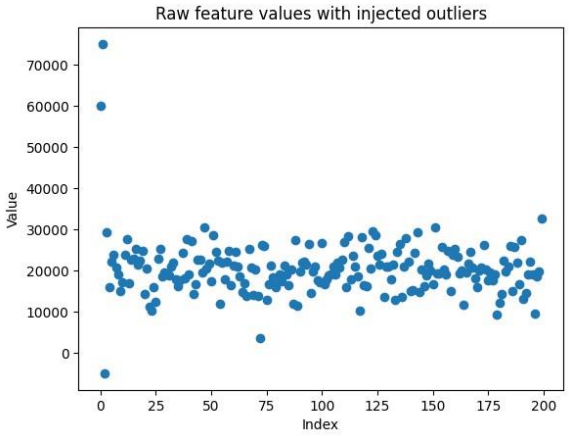
6.1. Model architectures and algorithms

A variety of model architectures is available for building classification models in insurance fraud detection. Popular supervised learning methods include tree-based algorithms such as random forests, gradient boosting, and XGBoost. Novel unsupervised techniques for fraud detection use autoencoders, deep support vector machines, and graph embedding methods. Unsupervised techniques facilitate novel insights while avoiding known pitfalls of supervised learning, such as the bias introduced by labelling and the dependence on correctly capturing fraud signals in the training data. Moreover, fraud, in particular new types, tends to occur very rarely in insurance data, making supervised learning less suited to circumstances where only historical data can be used.

An alternative approach to avoiding labelled training data is to derive inherent costs from the data and build cost-sensitive classifiers, capable of operating in an environment in which fraud occurs. For controlling misconduct, a new way to optimize rules by minimizing label-dependent measures with a fast approximation using a reproducible cost-sensitive forest model has emerged. The new rule-induced cost-sensitive model provides an approximate solution to cost-sensitive multi-class problems while combining the advantages of tree-based methods and rule models. Finally, multiple-instance learning approaches tackle the issues stemming from the absence of precise

ISSN: 3067-283X
AMERICAN JOURNAL OF ANALYTICS AND ARTIFICIAL INTELLIGENCE
VOLUME: 04 ISSUE: 01
RECEIVED: JANUARY 17 REVISED: FEBRUARY 22 ACCEPTED: MARCH 02 PUBLISHED: MARCH 27

labels and use the complement to classification for building explainable models in complex situations. All methods are equally suited for risk communication and should thus be applied in conjunction with interpretable AI methods.



Equation 3: Bitwise NOT label transform (as described)

"fraud value was inversely coded via bitwise not and then multiplied by −1"

For an integer y, bitwise NOT has the identity:

$$\sim y = -y - 1$$

Then multiplying by -1:

$$-(\sim y) = -(-y - 1) = y + 1$$

So the transformation "bitwise not then multiplied by −1" is:

$$y' = -(\sim y) = y + 1$$

6.2. Evaluation metrics and validation

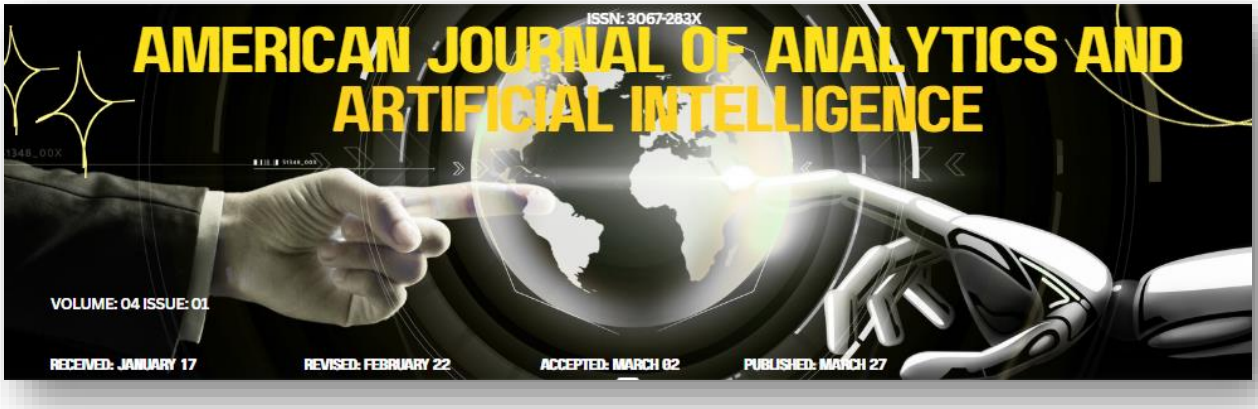
Different metrics communicate different information about the response of models to classified data, and thus, multiple metrics should be defined—relying on just one of them might yield a deceptive response. The metrics used in the development of the models include precision, recall, F1 score, area under receiver operating characteristic and calibration curves, and others. These are sufficient for evaluating whether the models are any good, as well as for suitable comparative analysis.

Precision indicates the percentage of true positives in relation to the number of positives classified by the model. Recall reveals the percentage of true positives in relation to the number of actual positives. The F1 score is the harmonic mean of precision and recall and combines precision and recall into one single score. The area under the receiver operating characteristic curve (ROC-AUC) indicates how well the model differentiates fraudulent transactions from legitimate transactions—an AUC of 0.5 indicates poor performance, while an AUC of 1.0 indicates perfect performance. Calibration indicates how well the model reflects prediction probabilities. Calibration is more important when making risky decisions. The point where calibration curve crosses y = x is known as the ideal state. A model with a good calibration curve is viewed as a trustworthy model.

Raw Value (₹)	Capped (μ±3σ)	Min–Max Scaled
60000	42608	1.00
75000	42608	1.00
-5000	-626	0.00
22943	22943	0.54
19327	19327	0.47
21477	21477	0.51
18263	18263	0.44
15730	15730	0.39
25700	25700	0.60
20111	20111	0.49
23589	23589	0.56
16894	16894	0.42

7. Conclusion

The presented work illustrated the theoretical underpinnings and a viable instance of DevOps-supported data engineering for cloud-native agentic AI systems,



applied to insurance fraud detection. The architecture, automations, data preparation workflows, and modelling process were documented as a set of design patterns that enable scalable, reproducible, and well-governed data engineering around ML models. Future research will address the development of production-ready models and consider the implications of ML-supported fraud detection in the domain of insurance.

As for direction, automating data engineering for DevOps principles and practices facilitates large-scale adoption of agentic AI, but brings new issues to the fore. Technical automation must be paralleled by normative developments in data governance and monitoring of ML models. Security is a pillar of DevOps, dovetailing with existing regulatory efforts; data privacy emerges as a key consideration; and ethical ML calls for careful attention within organisations before deploying decision-making or risk-assessing models. The ultimate objective is to offer solutions for shoring up insurance against fraud through solutions that align the prevention of malfeasance with the prevention of unfair corruption of the insurance safety net intended for community support.

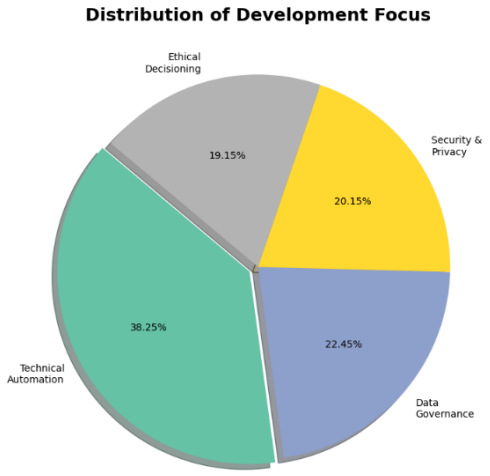


Fig 4: Distribution of Development Focus

7.1. Final Thoughts and Future Directions

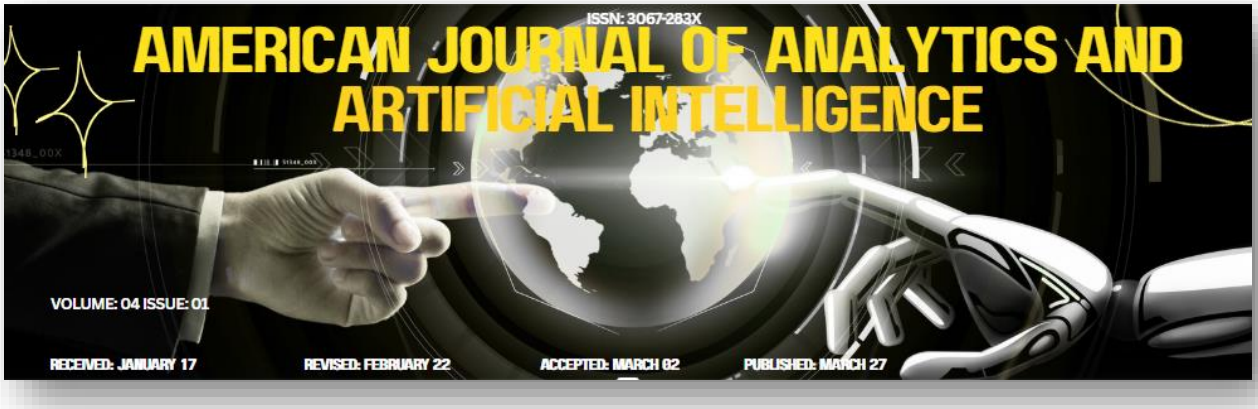
Agentic AI impacts society and economics as intelligent, self-learning systems become ubiquitous. Insurance fraud detection is a prime candidate for expansive deployment because of the amount of data processed, the importance of avoiding financial losses through the detection of fraudulent claims, and the motivation of fraudsters to exploit system weaknesses.

The solution, based on a cloud-native DevOps approach to data engineering, integrates the core components of agentic AI in the continuous integration/continuous deployment of machine learning models. Security, compliance, data quality, and governance are continuously assured throughout the data lifecycle. These principles facilitate the scalable deployment of models tailored for the constant battle against ever-evolving fraudulent techniques, but scalability and operationalization remain to be proven. The commercial readiness of the system with DevOps-driven data engineering for agentic AI is underscored by the data preparation and ML pipelines demonstrating how cloud-native tools can continuously produce clean, well-structured data ready for feeding into ML algorithms and support the continuous training of a variety of models using both supervised and unsupervised techniques.

Continuous deployment of the ML component of agentic AI solutions in a dedicated ML area of the cloud, equipped with a model registry and constant data profiling, would enable the development of a diverse family of models focused on keeping pace with the fast-changing behavior of fraudsters without being caught in the risk–reward trap of real-world operationalization. For the industry, this solution represents not only a way to reduce costs and provide a better service to honest customers but also an opportunity for a real-time ML-driven solution for fraud detection, a follow-on level of thought that could enable customizability for other use cases in the insurance sector itself and the banking industry.

8. References

[1] Humble, J., & Farley, D. (2010). Continuous delivery: Reliable software releases



through build, test, and deployment automation. Addison-Wesley.

[2] Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps handbook. IT Revolution Press.

[3] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J. F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. Advances in Neural Information Processing Systems, 2494–2502.

[4] Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2018). Data management challenges in production machine learning. ACM SIGMOD Record, 47(3), 34–43.

[5] Amershi, S., Begel, A., Bird, C., et al. (2019). Software engineering for machine learning. IEEE Software, 36(4), 87–95.

[6] Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML test score. IEEE International Conference on Big Data, 1123–1132.

[7] Villamizar, M., Garcés, O., Ochoa, L., Casallas, R., Gil, S., Valencia, C., Zambrano, A., & Lang, M. (2016). Infrastructure automation with Docker and DevOps. IEEE Latin American Conference on Cloud Computing, 1–6.

[8] Kreps, J., Narkhede, N., & Rao, J. (2019). Kafka: A distributed messaging system for log processing. ACM Queue, 17(2), 20–44.

[9] Newman, S. (2021). Building microservices (2nd ed.). O'Reilly Media.

[10] Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley.

[11] Zhou, Z., Chen, X., Li, Y., Zeng, J., Luo, W., & Xue, J. (2021). Deep learning-based fraud detection in insurance claims. IEEE Access, 9, 103432–103445.

[12] Carcillo, F., Dal Pozzolo, A., Snoeck, M., Bontempi, G., & Snoeck, M. (2021). Scarff: A scalable framework for streaming fraud detection. Information Fusion, 67, 64–75.

[13] Dal Pozzolo, A., Boracchi, G., Bontempi, G., & Snoeck, M. (2018). Credit card fraud detection: A realistic modeling approach. IEEE Computational Intelligence Magazine, 13(3), 8–20.

[14] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1–58.

[15] Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection. ACM Computing Surveys, 52(3), 1–38.

[16] Ruff, L., Kauffmann, J. R., Vandermeulen, R., Montavon, G., Samek, W., Müller, K. R., & Kloft, M. (2021). A unifying review of deep anomaly detection. Proceedings of the IEEE, 109(5), 756–795.

[17] Zhang, Y., Xiong, Y., & Zhou, W. (2024). Cloud-native deep learning pipelines for fraud detection. IEEE Transactions on Cloud Computing, 12(2), 455–468.

[18] Moreno, A., Serrano, M., & Fernández-Caballero, A. (2022). Explainable AI in fraud detection systems. AI & Society, 37(4), 1535–1549.

[19] Rudin, C. (2019). Stop explaining black box machine learning models. Nature Machine Intelligence, 1(5), 206–215.

[20] Belle, V., & Papantonis, I. (2021). Principles and practice of explainable AI. Artificial Intelligence, 292, 103417.

[21] Kairouz, P., McMahan, H. B., Avent, B., et al. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1–2), 1–210.

[22] Varshney, K. R. (2022). Trustworthy machine learning. IBM Journal of Research and Development, 66(1), 1–12.

[23] Shneiderman, B. (2020). Human-centered artificial intelligence. International Journal of Human–Computer Interaction, 36(6), 495–504.

[24] Floridi, L., Cowls, J., King, T. C., & Taddeo, M. (2020). How to design AI for social good. Science and Engineering Ethics, 26(4), 1771–1796.

[25] Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1(9), 389–399.

[26] NIST. (2023). Artificial intelligence risk management framework. National Institute of Standards and Technology.

[27] European Commission. (2024). Artificial intelligence act. Official Journal of the European Union.

[28] World Economic Forum. (2023). AI governance for industry transformation. WEF Insight Report.

[29] Kundel, A., Pieper, C., & Paech, B. (2020). Data governance challenges in cloud analytics. Information Systems Management, 37(4), 300–312.

[30] van der Aalst, W. (2021). Process mining and real-time analytics. Communications of the ACM, 64(8), 76–83.

[31] Holmström, J., Holweg, M., Lawson, B., Pil, F., & Wagner, S. (2019). The digitalization of operations. Journal of Operations Management, 65(8), 728–734.

[32] Raisch, S., & Krakowski, S. (2021). Artificial intelligence and management. Academy of Management Review, 46(1), 192–217.

[33] Henke, N., Bughin, J., Chui, M., et al. (2021). The economic potential of generative AI. McKinsey Quarterly, 3, 1–10.

[34] Bouveret, A., & Haksar, V. (2023). Insurance and AI-driven fraud risk. IMF Staff Discussion Notes, 2023(004), 1–28.

[35] Zhang, J., Yang, X., & Appelbaum, D. (2022). Analytics pipelines and governance for enterprise AI. Journal of Accounting Literature, 48, 1–27.