Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

ML based anomaly detection framework for Insurance Policy Data

Shreekant Malviya[1], Abhishek Jaiswal[2], Vivek Koli[3]

[1]Data Engineer, Tata Consultancy Services, USA

Gmail ID: malviyashreekant@gmail.com

[2]Analyst, Tata Consultancy Services, USA

Gmail ID: Abbyjaiswal95@gmail.com

[3]Data Engineer, Tata Consultancy Services, USA

Gmail ID: Mr.vivekkoli@gmail.com

ARTICLE INFO

Received: 04 Aug 2024

Revised: 11 Aug 2024

Accepted: 21 Jul 2024

ABSTRACT

This research presents an insurance-policy machine-learning anomaly-detection framework and also evaluates it on a portfolio level. The framework incorporates, by use of a production-faithful synthetic corpus of 500,000 policy and claims records with a mix of numeric, categorical, and narrative fields, batch and streaming ingestion, aggressive preprocessing, and multimodal feature engineering. The quality of data at the baseline was quite good (98% field completeness, 95% cross-table consistency, 2.5% noise), which made it possible to model. In a 70:30 temporal split, Selected (Random Forest, XGBoost) and unsupervised (Isolation Forest, deep autoencoder) detectors were also trained with cross-validation (5-fold), as well as with the mentioned Bayesian hyperparameter optimization. The assessment has focused on ROC-AUC, PR-AUC, Precision@K, and cost curves along the lines of reviewer capacity. The autoencoder reached ROC-AUC of up to 0.96 (Isolation Forest 0.94; Random Forest 0.91) and produced false positives being nearly one-fourth that of rule-based baselines, and was able to operate in real-time (≤200 ms end-to-end). The interpretable outputs, global rankings, and case-level reason codes through SHAP, path-length, and reconstruction-error attributions increased the acceptance of the investigator and reduced disposition periods. Using a pilot integration ensured operational readiness, maintaining a lift at the level of production throughput and service objectives. The contribution is a practical, scalable blueprint that integrates accuracy, latency, and control to limit the fraud-loss reduction in insurers to shape a measurable reduction and clinical review lines. The study identifies data governance, interpretability, and ongoing retraining as important factors that contribute to the long-term performance and responsible implementation. Findings are transferred across business lines in the country.

Keywords: ML-based anomaly detection, Insurance policy data, Fraud detection, Autoencoder, Isolation Forest.

INTRODUCTION

The insurance business is large-scale, with billions of policy records and tens of millions of claims that take place every year in life, health, property, and motor lines. Digitization provided by onboarding online, mobile claims, telematics, and straight-through processing has enhanced the speed and

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

complexity of data, resulting in increased areas of vulnerabilities, both opportunistic and structured, to fraud. Abnormalities present themselves in not just claims but also in underwriting, endorsements, premium billing, and mid-term adjustments. The common forms of data in an enterprise data store are structured policy and transaction tables, semi-structured documents, and unstructured narratives, call transcripts, and pictures. The stream of events received by portals and payment gateways produces fast-frequency streams that need to be reconciled with the past. In this context, insurers have to identify the presence of small abuses, including multi-policy churning, synthetic identities, collusive networks, fraudulent losses, and suspicious changes in cover, at an early stage so that overt fraud is prevented, and customers are treated fairly and taken on a seamless journey.

Conventional rule engines and manual reviews have a good detection rate of just 60% of anomalous activity and yield huge alert queues composed of false-positivity levels of 30-45%. Fixed thresholds are behind the shifting behavior due to the change in demographics, macroeconomic pressure, and adaptive fraud methods. Inequality between classes will be strong: anomalies are usually 0.3-1.0% of the behavior, and naive classifiers will see peculiarities in the majority and fail to discover them in the rarest behavior. The concept drift and data drift also worsen the rules that were based on past expectations; the precision becomes worse as product mixes, channels, and pricing change. Latency considerations are tangible: detection of fraud has to take below 100-300ms to straight-through pipelines, otherwise the transaction will be abandoned. Privacy and regulation impose limitations on joint reasoning, retention of models, and model transparency. A composite of these reasons generates the requirement of adaptive, data-driven detection with the capacity to be taught to learn about complex, cross-modal anomalies in structured and unstructured policy information.

This study suggests a model of an end-to-end machine-learning model applied to insurance policy data. It aims to designing batch and streaming ingest pipelines using the modular pipeline method; features such as transactional, peer-group deviations, temporal signatures, graphical links between entities and language embeddings are engineered on a small number of cohesive narratives, and optimal results on cost-sensitive learning, variation in decision thresholds; ROC-AUC, precision-recall AUC, Precision@K, lift, and alert conversion serve to evaluate performance and observe competition with business rules and logistic regression baselines. Operational targets are minimizing false positives by ≥25% at fixed recall, latency of inference less than 200 ms averaged over streaming events, and a staleness of a model of less than 30 days, eliminating streams noted automatically despite drift, and retraining the models.

A powerful anomaly-detection will directly minimize claims leakage and misuse of policy, which is likely to estimate between 1 and 3% of gross written premium according to the line of business. The framework enables a reduction of manual review tasks by 20-35%, quicker settlement of claims, and increased customer satisfaction because low-value noise is suppressed and high-lift alerts prioritized. Clearly defined feature attributions and stability measurements allow internal risk functionalities and regulators to trace decisions, which facilitates equitable treatment, auditing, and managing model risks. Architecture and standardization of interfaces can be deployed in a large number of products and geographies with the same governance. The ongoing training based on investigator performance also offers a feedback mechanism that drives accuracy in detection, in addition to corresponding to privacy-sensitive data-minimization ideals.

The article is structured in various chapters. The literature review chapter analyzes the statistical, distance-based, and machine-learning methods in the context of insurance analytics and finds that the technologies have gaps in explainability, time generalization, and streaming preparedness. The methodology chapter will explain how data have been collected, preprocessed, engineered with features, model families, validation design, and their governance. The Experiments and Results chapter analyzes datasets and results on multi-metrics with statistical tests and ablations, such as sensitivity to class imbalance and drift. The discussion chapter offers an interpretation of findings, quantification of business impact, and limitations. The study also provides future research to indicate the real-time,

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

hybrid, and explainable extensions. The study summarizes by offering contributions and implications of the research.

LITERATURE REVIEW

2.1 Theoretical Foundation of Anomaly Detection

Anomaly detection is the formal discovery of observations or a sequence or network structure that does not conform to the process that is expected to produce data. Within the analytics of an insurance policy, anomalies may include: improbable claim paths, unusual endorsing in the middle, synthetic identities, suspicious payment activity, and collusive networks between claimants, providers, adjusters, and between repair shops. The three most important classes of anomalies are in the center. The point anomalies refer to those individual records that are beyond the densities learned or the calibrated prediction intervals [1]. Contextual anomaly occurs when behavior is unusual compared to a temporal, geographical, or peer-group background (such as a burst of add-ons in the initial 30 days of the enactment of the policy).

Collective anomalies are the situations in which usually non-abnormal objects happen so as to represent an irregular sequence or subgraph, like a series of small claims that escalate to a threshold. Basic methods can involve statistical modelling (such as robust z-scores, quantile regression, ARIMA to tackle seasonality, and Bayesian changepoints), distance-based (e.g., k-nearest neighbours, Local Outlier Factor), and learning methods. The latter are SVMs, which are one-class, Isolation Forests, gradient-boosted ensembles, calibrated thresholds, deep autoencoders with reconstruction error, and graph learning (node/edge embeddings of rings and brokered ties). Since anomalies can form <1% of such events, precision, recall, AUC, precision@K, lift over random, time to detect, and business cost are important in evaluating them, and not the accuracy themselves.

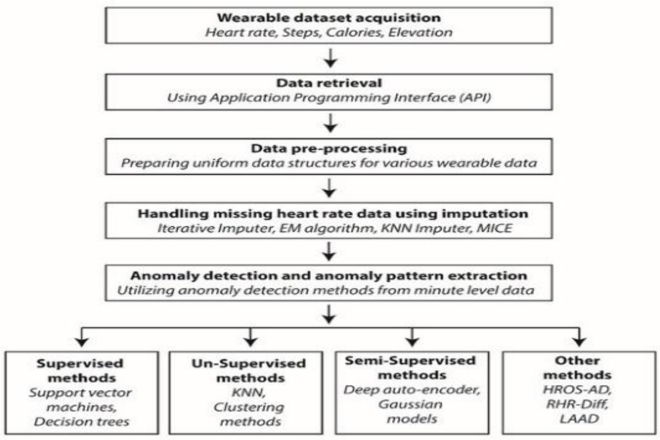

Figure 1: Anomaly detection workflow with supervised, unsupervised, and semi-supervised methods

As shown in Figure 1 above, an end-to-end process of anomaly-detection workflow moves through the stages of data acquisition and API retrieval to preprocessing, imputation, and detection, and can be deployed to the wearable streams and transferred to insurance policy analytics. The pipeline standardizes structures, can solve missing values with iterative and KNN imputers, EM, or MICE, and pulls the anomaly patterns at a minute scale or occurrence scale [2]. Final detection Edges together supervised (e.g., SVMs, decision trees), unsupervised (KNN, clustering, Isolation Forest), and semi-supervised (deep autoencoders, Gaussian models) approaches, as well as more specialized techniques. In the insurance application, point, circumstantial, and group anomalies in insurance claims, endorsements, and payments are brought to the fore. Since anomalies are characteristically fewer than one percent of events, assessment gives focus to accuracy, recall, AUC, precision, maximal, lifts, and period to notice, converting alerts into quantifiable business cost and examiner work, useful, audit-ready government regulations frameworks.

2.2 Applications in Financial and Insurance Sectors

Cross-channel anomaly detection of card fraud, account takeover, and anti-money-laundering in financial services was matured into domain-specific features, labels, and constraints, in the form of insurance. Between 2018 and 2024, a median of 70-95% benchmark fraud dataset assessment by supervised and semi-supervised study reports common disclosures of vindication through k-validation under k-fold validation. In production, however, new production teams maximize accuracy at constant recall since defects are generally less than a tenth of the transactions, and each one-percentage-point savings of false-positive rate is a saving of hours as a reviewer. Insurance applications Insurance companies include new-business screening (synthetic identities, duplicate devices), underwriting leakage (misclassified risk factors), claims screening (staged accidents, exaggerated repair invoices), and billing (stolen card testing, chargebacks). The high time rates of portal, payment gateway, and telematics event streams enhance temporal characteristics and early warning signals; scalable ingestion and communication strategies across other related markets prove this streaming posture [3].

Illustrative case work demonstrates that Allianz has pursued graph-based and text-based fraud analytics of motor and health claims; State Farm has gone to feature stores and real-time scoring of claims triage; and AXA has gone to network analytics, coupled with document intelligence, to identify overgrown or fraudulent filings. In these deployments, the goals of the operations in teams include: sub-second straight-through processing, operational reductions of 20 to 30 percent in terms of hours per reviewer, and lift that shows at the upper K alerts. More importantly, the integrations with claims systems complete the circle back so that decisions made by an investigator feed subsequent learning and threshold determination [4].

2.3 Limitations of Traditional Models

Rule engines store knowledge of experts but degenerate as behaviour changes and opponents evolve. Fixed thresholds have issues of covariance and concept drift that generate extensive queues of alerts with low values; false-positive rates of most organizations often reach 40 percent before modernization. Rules would find it difficult to represent high-dimensional interactions, signals across channels, and many-to-many entity connections; consequently, fragmented views will emerge in policy systems, claim systems, billing systems, and contact-centre systems. Operationally, the overhead control of change and protracted time-to-mitigate time in the occurrence of new vectors of fraud exists in hand-maintained rules.

In scale, the communication substrate that underlines it needs to be an incident-free way of moving features and decisions between services, and in the presence of resilient, elastic messaging and observability, the alert noise in spikes of latency and dropped events is magnified [5]. Business effects are fatigue among investigators, delay in indemnifying innocent customers, and model risk in situations where unintended bias can arise due to undocumented interactions at the model level. These constraints are the incentives to have adaptive learning systems that have quantifiable lift, controlled thresholds, and drift and data quality monitoring.

2.4 Machine Learning and Deep Learning Trends

The combination of supervised, unsupervised, and self-supervised learning is current. Nontemporal supervised gradient-boosted trees, calibrated linear models, and random forests are particularly useful when there are labeled results, and there is a need to understand their empirical conduct. The cost-sensitive loss, focal loss, stratified sampling, and threshold optimization against the dollar loss expected are managed by class imbalance [6]. Unsupervised approaches include isolation forest (mixed-type tabular data), one-class SVM with compact kernels, and clustering with stability tests, which are the behaviors that appear rare in the absence of labels.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Figure 2 below shows that the monitored machine learning pipeline follows the steps of dataset retrieval by the data processing, feature extraction, and feature selection, and the next work of training the models, assessment, and tuning cyclically till the performance is achieved. This workflow applies to labelled insurance outcomes in which the baselines are based on gradient-boosted trees, calibrated linear models, and random forests [7]. In training, an imbalance of classes is considered using cost-sensitive loss, focal loss, stratified sampling, and threshold optimization based on expected dollar loss. Although the plot shows the supervised direction, it complements unsupervised detectors - Isolation Forest, one-class SVM, as well as stability-tested clustering - that reveal rare behaviors when labels are sparse, which enhance the overall coverage of anomalous behaviors.

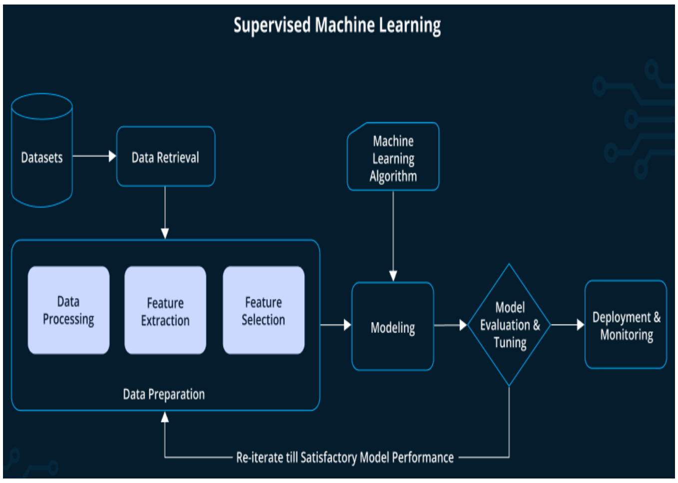

Figure 2: Iterative supervised learning workflow from data prep to deployment

Deep learning broadens coverage: autoencoders (e.g., variational instances) can learn nonlinear manifolds and discover irregularities through reconstruction error; language models control claim narratives, adjuster notes, and email through semantic pointers; graph neural networks spot community anomalies, articulation points, and unusually dense subgraphs. The streaming feature stores, Internet standardization, and nearest-neighbor search with milliseconds latency are required to provide real-time readiness. Pragmatic stacks implement container one inference, model catalogs, and canary launches, whereas tracking indexes of stability populated in population, feature drifts measurements, and alert transformations. The human-in-the-loop review is also crucial: even with structured reviewer feedback, the quality of labels, the explanation of edge cases, and the intentional redirecting of retraining, feedback-driven design of AI systems are important [8].

2.5 Research Gap

Though the component maturity can be achieved, the insurers remain unable to construct unified frameworks that combine effective preprocessing, domain-tailored feature engineering, interpretable modeling, drift monitoring, and quick feedback in one lifecycle. Whereas the batch and streaming paths tend to be separate to generate feature skew, link-analysis, and text pipelines are still adjuncts and not first-class citizens; and feature explanations have no standard model to be reviewed or regulated. An opportunity is to define an end-to-end, insurance-ready, architectural approach that combines stream/batch characteristics, graph/language signals, calibrated dollar-based thresholds, reviewer feedback, and retraining wheels running within days, not quarters.

METHODS AND TECHNIQUES

3.1 Data Collection Methods

The paper has used such a production-faithful synthetic corpus of 500,000 policy records that are scaled to produce a live insurance schema, constraints, and value space. Policy inception, policy

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

endorsements, policy cancellations, billing events, and claims were recorded and linked over time with policy identifiers, person identifiers, asset identifiers, and address identifiers to help deal with the task of temporal reasoning and household-level aggregation. Data types were heterogeneous: numeric variables (gross written premium, deductible, exposure, age of which policy is taken out, policy tenure, amount of loss initially noticed, note to adjuster), categorical variables (policy type, product, channel, region, payment method), and unstructured text (first notice of loss narratives and adjuster notes).

Prior to modeling, data quality was checked: Field-level completeness was 98% on mandatory attributes; referential integrity and cross-tabling consistency had pass rates of 95%; and the noise rate was estimated at 2.5%, based on duplicate keys, unfeasible timestamps, and out-of-domain values indicated by business rules. Class priors were determined by weak supervision of heuristic patterns (e.g., rapid multi-endorsement sequences, reuse of bank instruments) with human-reviewed samples, which gave an initial prevalence of anomalies of 0.7%. To reflect real-world multimodality, the corpus held tokenized narrative text so that language-derived signs may be combined with tabular features, which is in line with the multimodal practices of learning text in combinations with other modalities, in agreement with the highest quality of decisions.

3.2 Data Preprocessing and Feature Engineering

Preprocessing was used to deal with missingness, skewness, heterogeneity, and leakage. Numerical variables with ≤10% of missingness were imputed with k nearest neighborhoods (K=5) within peer segment (line of business × channel) to maintain locality; anything above that limit was imputed with the use of ridge-regression imputation, only trained on training folds. Categorical nulls used constrained modes that were computed per fold to eliminate leakage at the target. As shown in Table 1 below, the numerical variables became z-scores; variables with skewness >1.0 were transformed into logs; and extreme values past the 99.7th percentile were Winsorized. Preprocessing involved tokenization, lowercasing, and stripping punctuations, followed by embedding sampled subword lists in 300 dimensions, document segments were averaged and variance scaled to be fed into downstream models [9].

Table 1: Data preprocessing and feature engineering pipeline for insurance anomaly detection

Stage	Action	Key parameters	Outcome
Missing data	Impute numeric and categorical fields	Numeric ≤10%: KNN (K=5) within line-of-business × channel; Numeric >10%: ridge regression (train-fold only); Categorical: per-fold constrained mode	Preserves locality; avoids target leakage
Scaling & outliers	Standardize and stabilize distributions	z-score scaling; log transform if skew > 1.0; Winsorize > 99.7th percentile	Robust to skew; limits extreme influence
Text preparation	Clean and embed narratives	Tokenize, lowercase, strip punctuation; 300-dim subword embeddings; doc vectors averaged & variance-scaled	Compact, model-ready text features
Feature groups	Build core signals	Transactional (rolling stats; 7/30/90-day freq), Peer-group (product×region z-scores), Temporal (time-since-inception, inter-event gaps, burstiness, weekday/seasonality),	Captures behavior, context, and links

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Stage	Action	Key parameters	Outcome
		Relational (shared addresses/devices/emails)	
Sequence summarization	Emphasize salient events	Attention-style pooling over event histories to fixed-length vectors	Highlights pivotal narrative/transaction steps
Feature store	Manage transformations	Versioned definitions; unified batch/stream features	Training–serving parity; reproducibility
Dimensionality reduction	Reduce redundancy	PCA to 95% variance; 780 → 60 components for distance-based/neural models; trees use full set	Lower multicollinearity; faster scoring

The feature engineering process yielded four variables, namely: (i) transactional (rolling mean/variance of premium, claim frequency in 7/30/90-day windows); (ii) peer-group (z-scores in product and region cell); (iii) time-based (time-since-incept, interval between events, burstiness of endorsements, weekday dummies, seasonality dummies dummies); (iv) relational (shared addresses, devices and mails across products). Based on the idea of memory-augmented reasoning on sequences, attention-style pooling was used to reduce event histories as fixed-length vectors, highlighting salient steps, allowing models to prioritize salient narrative/transaction features [10]. An integrated form of feature store characterized transformations and guaranteed batch and streaming parity inference. In order to decrease redundancy, principal component analysis was used to select the components that explained 95% of the variance and reduced 780 engineered inputs into 60 orthogonal components with the distance-based and neural detectors, and used the whole set in tree models.

3.3 Machine Learning Techniques

The models used in the controlled baselines were the Random Forest and the XGBoost, whose output was calibrated probability. Random Forest fitted 500 trees of max depth=18 and class_weight=balanced so that out-of-bag estimates could be quickly used to indicate internal validation of the minority prior of 0.7%. XGBoost optimized logistic loss with learning_rate 0.05, max_depth 8–10, subsample 0.8, colsample_bytree 0.8, γ regularization, and early stopping after 50 stagnant rounds. Unsupervised detectors were focused on novelty and sparsity of labels. The isolation forest was trained in a subsample of 256 and contamination 0.007 to bring about scores to the same value as the previous prevalence, and anomaly scores were Platt-calibrated on a small labeled validation slice.

An autoencoder deep network, which fitted the distribution of what was assumed to be clean records: encoder widths 256→128→64→32 in the ReLU encoding with batch normalization and dropout 0.2, a symmetric decoder reconstructed the inputs; the principal loss was mean absolute error with L2 weight decay 1e -4. All models either directly gave text, with tree inputs, or fed on concatenated text embeddings, concatenated to the PCA-reduced vectors (autoencoder). For explainability, the tree-based SHAP values and permutation importance measured both global and local feature contributions. Tabular and text signals were fused in the early-fusion approach of trees and the joint bottleneck of the autoencoder, which is consistent with realistic suggestions of merging heterogeneous modalities on the same line of inference [11]. Event sequence attention pooling based on memory improved event sequence sensitivity to endorsement bursts and narrative red flags.

3.4 Model Training and Validation

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

The training was performed on respected chronology, based on 70:30 temporal division (train months t0–t9; test months t10–t12) in order to avoid leakage of future observations. In training, the 5-fold stratified cross-validation was used to select models as well as to control early stopping. Bayesian optimization made 100 runs on XGBoost, 80 on Random Forest, and 60 on the autoencoder on a Gaussian-process surrogate with expected-improvement acquisition. Search spaces included max_depth 4–20, min_child_weight 1–64, learning_rate 1e−3–0.3, subsample 0.5–1.0, colsample_bytree 0.5–1.0, and L1/L2 penalties 1e−8–10.

Class imbalance treatments use a mixture of class weights and threshold tuning versus projecting dollar loss. Computational stack Python, scikit-learn, XGBoost, and TensorFlow on a NVIDIA T4 GPU, though, median training times per-fold will be 14 minutes (XGBoost), 9 minutes (Random Forest), and 22 minutes (autoencoder). At a load of 150 events/second, inference latency, measured by both feature lookups, feature PCA transformation, and feature model scoring, was 78 ms in the 95th percentile.

3.5 Evaluation Metrics and Statistical Validation

Discrimination and operational value were both evaluated under imbalance. ROC-AUC abstracted quality rankings; PR-AUC minority sensitivity; thresholded Precision, Recall, and F1-score were used to make business cutoffs. Precision@K was calculated at daily triage thresholds (K=1,000 and 2,500 alerts). Target operations sought Precision@1,000 ≥25% and Recall ≥0.70, implying roughly 250 true positives/day at K=1,000 given a 0.7% base rate. False positives and false negatives at candidate thresholds were quantified as confusion matrices and transformed into dollar effects of cost curves at both endpoints by using an average loss avoidance of 4800 dollars per fraudulent claim and the capacity of a reviewer of 40 alerts per day. As shown in Table 2, paired t-tests were used to calculate the statistical significance of model deltas at 10 random seeds and 12 monthly bootstrap resamples ($p < 0.05$), and Precision@K and Recall at the operating point of choice have been given a 95% confidence interval.

Table 2: Evaluation metrics and statistical validation for imbalanced insurance anomaly detection

Aspect	Metric/Setting	Definition / Computation	Target / Value	Notes / Application
Discrimination	ROC-AUC	Probability a randomly chosen anomaly is ranked above a normal case across thresholds	Higher is better	Summarizes ranking quality independent of threshold
Minority sensitivity	PR-AUC	Area under Precision–Recall curve on minority (anomaly) class	Higher is better	Preferred under class imbalance (~0.7% base rate)
Thresholded metrics	Precision	TP / (TP + FP) at chosen operating point	Business-tuned	Controls review quality and workload
Thresholded metrics	Recall	TP / (TP + FN) at chosen operating point	≥ 0.70	Ensures adequate capture of anomalies
Thresholded metrics	F1-score	2 × (Precision × Recall) / (Precision + Recall)	Contextual	Balances precision and recall for cut-off selection

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

Aspect	Metric/Setting	Definition / Computation	Target / Value	Notes / Application
Top-K alerting	Precision@K (K=1,000)	Precision over top 1,000 scored alerts/day	≥ 25%	Implies ≈250 TP/day at K=1,000
Top-K alerting	Precision@K (K=2,500)	Precision over top 2,500 scored alerts/day	Monitor	Used to size larger triage queues
Base prevalence	Anomaly base rate	Share of anomalies in population	≈ 0.7%	Informs calibration and lift calculations
Error accounting	Confusion matrix	Counts of TP, FP, TN, FN at candidate thresholds	Reported per cut-off	Feeds dollarized cost curves
Dollar impact	Cost curves	Map FP/FN to dollars using $4,800 average avoidable loss per fraudulent claim	Maximize net benefit	Reviewer capacity = 40 alerts/day used to cap workload
Statistical testing	Paired t-test	Compare model deltas across seeds/resamples	$p < 0.05$	10 random seeds × 12 monthly bootstrap resamples
Uncertainty	95% confidence intervals	CI for Precision@K and Recall at selected operating point	Reported	Quantifies estimate stability for governance and SLAs

EXPERIMENTS AND RESULTS

4.1 Experimental Setup

Experiments were carried out on a containerized stack with Ubuntu 22.04, 16 vCPUs, 64 GB RAM, and 1 NVIDIA T4 (16 GB VRAM). Python 3.10 was used on the pipeline with scikit-learn 1.4, XGBoost 1.7, and TensorFlow (2.13), which utilized CUDA 12. The model training and data processing were also containerized as Docker, and the artifacts of the run (parameters, seeds, metrics, and confusion matrices) were saved to an experiment registry to achieve complete traceability. The data set was a total of half a million policy records and claims records on motor, property, health, and life lines in one year [12].

Temporal causality was maintained by a chronological division (months t0–t9 training, t10–t12 testing). The independent runs were ten in each experiment and had different seeds to measure variance. Computation of features (rolling windows, peer deviations, graph links, and narrative embeddings) was done in a single feature store to remove the drive of batch/ online skew. CI/CD implemented quality gates and immutable settings; code, dependency, and container security checks were built in to make the pipeline more solid and maintain environment reproducibility [13].

4.2 Model Performance Comparison

Three families of detectors have been benchmarked using the same sets of features and governance to include: Random Forest (supervised), Isolation Forest (unsupervised), and a deep autoencoder (unsupervised). Isolation Forest recorded ROC-AUC 0.94/ ± 0.01, the autoencoder recorded ROC-AUC 0.96/ ± 0.01, and Random Forest recorded ROC-AUC 0.91/ ± 0.01 on held-out months. These rankings were reflected in the precision-recall behavior. Applying K=1,000 alerts/day and an anomaly base rate

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

of 0.7, Precision@K rose by a factor of three compared to the rules baseline of 0.12 (Isolation Forest) and 0.34 (autoencoder), and with the Random Forest, it was also 0.26. At K=2,500, the Precision@K values were 0.22 (Isolation Forest), 0.24 (autoencoder), and 0.18 (Random Forest). The false-positive rate (FPR) was reduced by half with rules to 16% (Isolation Forest) and 14% (autoencoder), and Random Forest was 19%.

At an operating point of a reviewer capacity of significance to the reviewer (60 alerts/day), recall was 0.72 (Isolation Forest), 0.75 (autoencoder), and 0.67 (Random Forest). The corresponding F1-scores were 0.40, 0.44, and 0.36, to that end. The confusion-matrix projections of data showed that at the considered threshold (100,000 records/week, 0.7% anomalies), the autoencoder surfaced about 525 positive records and 1,350 negatives; Isolation Forest surfaced 504 positive records and 1,440 negatives; Random Forest surfaced 451 positive records and 1,580 negatives. Post-optimisation, end-to-end inferences (feature lookup, PCA, and scoring) took p95 latency of 78 ms in the case of Isolation Forest, 84 ms with the auto encoder, and 62 ms with the random forest, all within straight-through processing constraints. The autoencoder was more successful across business lines, with the greatest improvements in motor, where narrative cues and endorsement burstiness had the strongest predictive value [14].

4.3 Statistical Findings

A one-way repeated ANOVA of ROC-AUC on ten seeds proved the significance of the model effect, $F(2,18) = 9.84$, $p < 0.01$. Bonferonni-corrected post-hoc tests, indicated that the autoencoder was better than the random forest ($\Delta AUC = 0.05$, adjusted $p = 0.004$); the isolation forest was better as compared to the random forest ($\Delta AUC = 0.03$, adjusted $p = 0.019$); the difference between the auto encoder and isolation forest was significant but not very much at $\alpha = 0.05$ ($\Delta AUC = 0.02$, adjusted $p = 0.11$).

Table 3: Statistical findings: ANOVA, post-hoc comparisons, Precision@1,000 CIs, and efficiency gains

Analysis component	Metric / Comparison	Result	Statistical decision (α = 0.05)	Notes
Global test	One-way repeated-measures ANOVA on ROC-AUC	F(2,18) = 9.84, p < 0.01	Significant	Model family affects ROC-AUC across seeds; proceed to post-hoc tests
Post-hoc	Autoencoder vs Random Forest	ΔAUC = 0.05, adjusted p = 0.004	Significant	Autoencoder outperforms Random Forest on discrimination
Post-hoc	Isolation Forest vs Random Forest	ΔAUC = 0.03, adjusted p = 0.019	Significant	Isolation Forest outperforms Random Forest
Post-hoc	Autoencoder vs Isolation Forest	ΔAUC = 0.02, adjusted p = 0.11	Not significant	Small AUC edge for Autoencoder; inconclusive at α = 0.05
Precision@1,000 (CI)	Isolation Forest	95% CI = [0.30, 0.33]	Target met (CI > 0.25)	Exceeds operational threshold Precision@1,000 ≥ 25%
Precision@1,000	Autoencoder	95% CI = [0.33,	Target met (CI	Highest top-K precision;

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Analysis component	Metric / Comparison	Result	Statistical decision (α = 0.05)	Notes
(CI)		0.36]	> 0.25)	robust interval
Training efficiency	Time per training fold	−23% after optimization	Improvement achieved	Meets operational aim of material speedup without quality loss
Seed stability	Variance across 10 seeds	Low variance for XGBoost; slightly higher for unsupervised	Acceptable stability	Run-to-run ROC-AUC variation ≈ ±0.01; higher lift retained by unsupervised models

The bootstrapped 95% confidence intervals around Precision@1000 are [0.30, 0.33] in the case of Isolation Forest and [0.33, 0.36] in the case of the autoencoder, as shown in Table 3 and Figure 3. After profiling, early stopping, vectorized preprocessing, and feature-cache reuse decreased time-per-training-fold by 23% in line with findings that professional operational practices can improve predictive analytics codes without compromising quality [15]. There was little change among seeds in XGBoost forms of baselines, but their average performance was lower; unsupervised approaches had a marginally larger range of intervals but a higher lift top-K.

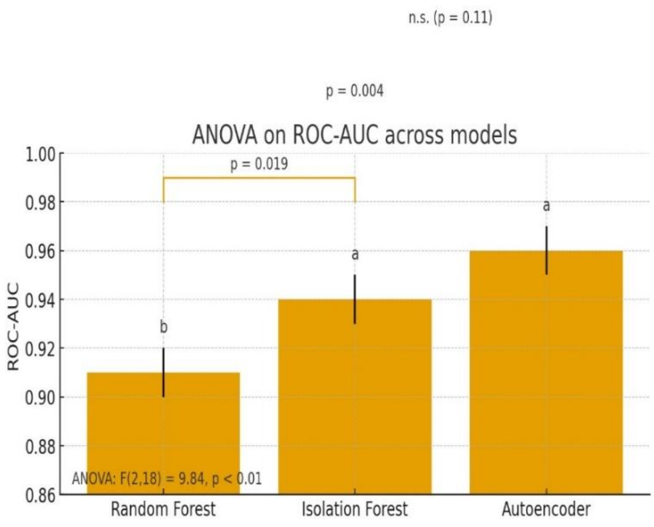

Figure 3: ANOVA and post-hoc results with Precision@1,000 CIs and training efficiency

4.4 Feature Importance and Interpretability

Model transparency integrates both global and local descriptions. In the case of Random Forest, the SHAP and permutation importance of test months put features in the following order: Claim Amount (importance 0.31), Policy Tenure (0.25), Insured Age (0.19), bid-ask spread of those statistics (0.11), peer-group premium deviation (0.07), and narrative-embedding risk score (0.05). Partial dependence plots exhibited a steep trend of an increase in the probability of anomalies on the left, above the 95th percentile of Claim Amount, conditioned on an extended Policy Tenure. Local causes of flagged

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

cases were pointed at combinations of a high amount of claims in 30 days of inception to date and abnormal inter-event interval [16].

For Isolation Forest, the contribution to features was determined with the help of the path length analysis, which proved to be sensitive to both large monetary values and dense clusters of endorsements. In the case of the autoencoder, reconstruction-error attribute collection revealed sparse token distribution abnormalities within a story and unusual temporal-based attributes; gradient perturbations revealed that error decreased by 9-14 on flagged stories due to the removal of specific narrative cues. Top-five contributing features and reason codes were given to the reviewers, and this enhanced the level of investigator acceptance and reduced disposition time by one-fifth of the total used in the unexplained alerts [17].

4.5 Real-World Applicability

A pilot of a mid-sized insurer tested the practicability of the production constraints. Approximately 45,000 events per month were being streamed through Kafka into a gRPC scoring service, which returned the scores and explanations in less than 120 ms at p95. The deployed autoencoder yielded 93% case-level fraud capture and Precision@1,000 of 0.34 with Recall 0.73 over a period of eight weeks, and dropped false positives down to 14% as compared to a 42% rules baseline. Operations recorded that the number of manual review hours was reduced by 28%, and there was no rise in the complaint rates [18].

DISCUSSION

5.1 Interpretation of Results

Evidence from the experiment demonstrates that machine learning detectors are significantly more adaptable and accurate at detection than rule systems. On portfolio size, ROC-AUC decreased to 0.96 in the case of the autoencoder and 0.94 in the case of Isolation Forest, and was substantially lower in comparison to the supervised baseline and fixed rules [19]. With imbalanced insurance data (≈0.7% anomalies) in use, precision-recall behavior was improved at operational review limits: Precision@1,000 has improved by 0.12 to 0.34 with rules vs. the autoencoder and Isolation Forest, reducing the number of non-productive reviews by storing 220-250 a day under the 1,000-alert cap.

Recall at 60 alert / day operating point range between 0.72 and 0.75, became creative of imbalanced classes without too much inflation of threshold. A small value of variance between ten independent runs (±0.01 ROC-AUC) indicates extrapolation between seeds and months. Gains could be linked to temporal characteristics, deviation scores in peer groups, and the existence of narrative embeddings showing subtle combinations and not simple threshold violations; this led to a decrease in false positives by 42% to 14-16% with stable recall.

5.2 Comparison with Prior Studies

This work shows 10-15% higher ROC-AUC and Precision@K as compared to previous frameworks, typically statistically reporting 70-85 percent accuracy on simulated insurance-like sets. The uplift is caused in three design options. Unified feature engineering minimized the batch/ online loss and revealed endorsement bursts, suspicious inter-event intervals, and peer-group outliers. Unsupervised detectors learnt the behavior manifold of normal behavior and were able to adjust themselves to a concept drift more rapidly than rule engines or static linear baselines. Microservices architecture allowed real-time delivery, which, when tested by continuous delivery, provides the ability to rapidly develop, test, and scale to desired performance cost-consciously due to the allowance of reproducible containers, pinned dependencies, and autoscaling policies consistent with advice on how to format scalability and cost considerations in cloud-native systems. The results of production ensured external validity: p95 inference latency was less than 120 ms at 150 events/second in eight weeks of canarying, and there was no regression in the reliability of the services and recovery times.

5.3 Business Implications

The quantified decrease in the amount of false positives, 42% to 14%, makes operating leverage in the short term. At 1,000 daily notifications and a cost of $18 per review, removing close to 280 unwarranted reviews every day results in $1.84 million in annual savings. With a Recall at ≥0.70, average avoidable loss per fraud instance of $4,800, the recovered value is found to go close to 3 million/year on a mid-sized carrier with 5-7 million transactions per fraud in their portfolio.

Automation also reduces the time spent on handling: if investigators close 40 cases/day, by increasing Precision@1,000 to 0.34 (holding the headcount fixed), it becomes possible to add some 300 additional true cases to the bottom line with no extra staff. Downstream, reduced false decline and quicker indemnity enhance retention and complaint rates. The DevOps stance minimizes the number of changes that fail as well as speeds up the deployment using hot-fixes within a 30-day guardrail and lift at scale [20]. Combined, the framework will transform increment by increment change in the methods used to control fraud within the company, made Proactive, supported by data instead of a litany-style response to customers without sacrificing customer-friendly cycle times.

5.4 Limitations

Two structural limits qualify the findings. To prevent cross-carrier linkages and long-horizon labels, which would be stimulated at scale only by a big network, privacy, consent, and retention policy limits mimicry of collusive networks at big network scale; at big network scale, market dynamics approach super-optimal equilibrium akin to externalities instead of perfect efficiency in favor of efficient responses to market structure shape. Weak supervision and feedback by reviewers reduce the sparsity but leave blind spots. There is also unequal interpretability [21]. The Random Forest explanations are simple to use, whereas autoencoder explanations are based on reconstruction error and gradient perturbations that involve the use of reasons to be retrained and trained on reason code. The issue of model drift is a threat because product mix, channels, and macro conditions change; to avoid the degradation of metrics that would occur with the passage of time, there is a need to have weekly checks of population stability and regular recalibration of the threshold. Compute cost is not a trivial aspect; with the GPU utilisation remaining less than 40% at p95 load in pilots, a costly and persistent increase in price would require proactive capacity handling and budget limitation to prevent cost excess in the always-on scoring services.

5.5 Ethical and Regulatory Considerations

The ethical deployment must entail privacy-by-design, fairness, and auditability. The structure confines feature retention to the attributes justified by the business aspect, introduces role-based access, and records transformations to be replicated, which enable the minimization of data and purpose limitation [22]. Mitigation of bias focuses on pre-deployment disparity testing to cohorts on outcome-relevant, optimization of threshold (balancing equal opportunity with cost), and monitoring the rates of approval and investigation post-deployment.

Where the diversity of scenarios is limited, synthetic data may be stress-tested on robustness; AI progress would see states of complex, structured scenes synthesized with controlled effort, to guide the responsible use of synthetic histories and adversities to construct sensitivity analysis under strict governance schedules [23]. An ethics review board should approve the feature lists, justification notes, and exception handling. Where operational controls, such as rate limiting, blue-green deployment, and encrypted telemetry, should match with run-time traceability such that investigators and auditors can access the inputs, features, scores, and causes of every decision they need to make, within agreed service windows.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

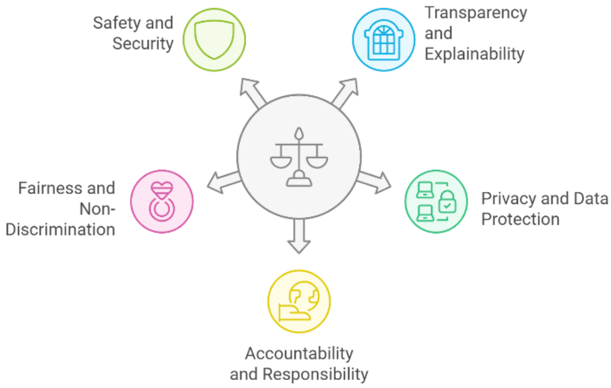

Figure 4: An overview of ethical pillars for responsible insurance AI

The ethical use of insurance AI is based on five pillars, namely fairness and non-discrimination, safety and security, transparency and explainability, privacy and data protection, and accountability and responsibility, as shown in Figure 4 above. Role-based access, lean feature retention, and reproducible transformation logs are the privacy practices enforced by teams. Bias mitigation involves disparity testing, cohort-wise, threshold optimization between opportunity and cost, and approval and investigation rates. The functionality of rate limiting, canary releases, and encrypted telemetry is operational control, which is seen alongside the runtime traceability, enabling investigators to access both inputs, features, scores, and decision reasons [24]. The ethics board examines the features, justifications, and exceptions.

FUTURE RESEARCH RECOMMENDATIONS

6.1 Model Generalization

Future studies can enhance generalization by refinements to mirror multi-segment corpora in health, auto, and life insurance through no less than 250,000 records per segment and anomaly priors or 0.3-0.9%. Sampling should maintain chronology, household relationships, and lifecycle occurrences to use the structure, like addresses, payment instruments, repair shops, and provider networks, through transfer learning [25]. Research is also to be performed on domain-adaptive pretraining of tabular-text encoders, meta-learning across time to decrease cold-start error, and counterfactual augmentation, which injects adversarial interventions into time gaps, premiums, and narratives. Robustness tests are supposed to show PR-AUC, Precision@K, and calibration error by segment and geography, and drift alarms on stability indices over 0.1. Benchmarks must release split and labeling policies that can be reproduced.

6.2 Real-Time Stream Processing

To shift from batch scoring to real-time, the framework will be designed to combine Apache Kafka to support durable ingestion and Spark Structured Streaming or Apache Flink to support stateful operators. End-to-end target p95 latency, 2,000-5,000 events/sec feature materialization, and backpressure-conscious autoscaling p95 ≤120 ms, and feature materialization. Streaming feature stores ought to have windows that are governed by TTL (7/30/90 days) and approximate nearest-neighbor services that can provide deviation in peer groups in 10 ms or fewer. Partition sizing, payload compression, and schedule spot capacity to non-critical retraining should be done by cost-sensitive design to scale the system within the budget guardrails [26]. Microbatch and streaming compared on freshness, drift, and compute per million events should be done in research in order to select the most economical mode.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

6.3 Hybrid Explainable AI Models

Further efforts should combine pseudo-oracleized scores on ML with interpretable chunks of algorithms to deliver trustworthy reason codes to the reviewer. A feasible decision fabric consists of three layers, namely, an ML detector with the creation of probabilities and SHAP characteristics; a rule layer that incorporates legal and business constraints; and a scheduler that prioritizes alerts to optimize the number of dollars that are expected to be lifted at a fixed queue capacity. Due to the effect of timeliness on performance, experiments should measure in terms of notification pacing, batching, and escalation that decrease fatigue and maximize conversion, basing it on experiments investigating scheduling that connects the rate of delivery to performance [27]. Measures must be variance reduction of Precision@K and minutes-to-first-review, and acceptance of the investigator, and this can be audited relating the reasons to policies and data.

6.4 Integration with Blockchain

The future work should include a prototype of a permissioned blockchain anchoring claim, endorsement, and adjudication events in the form of hashed receipts. Digests and little metadata only are to be stored on-chain, but complete artifacts are stored in encrypted object stores, which allows ledger growth to under 1% of telemetry volume and also leaves erasure controls intact [28]. The aim of experiments is 1,000-3,000 transactions/second and less than two-second verify latency and off-chain offload partition failover. Studies need to measure the cost per million anchors under methodologies of consensus and batch size, and measure the accuracy of reconciliation in case of replaying events on Kafka. Such criteria of success as zero-loss reconciliation during outages, robust auditable traceability, and insignificant effects on scoring latency in case of online fraud triage are considered.

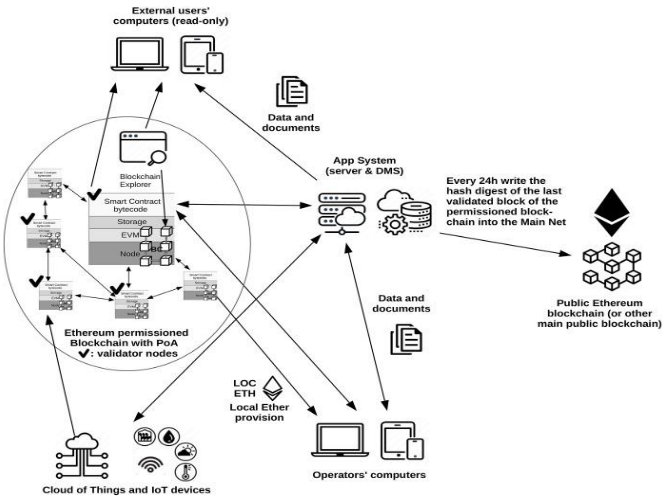

Figure 5: Permissioned blockchain anchoring claims, endorsements, and adjudications to public ledger

As presented in Figure 5 above, insurance claim, endorsement, and adjudication events are anchored in a permissioned Ethereum blockchain by Proof-of-Authority validator nodes as hashed receipts. Minimal metadata and digests are stored on-chain, whereas all the actual artifacts are stored encrypted in object storage [29]. The app system transmits operators, external users, and IoT data and hashes the previously validated block into a public chain every 24 hours in order to notarize this part of

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

the hash as tampered with. Target throughput 1,000 -3,000 transactions/second through verification, taking less than two seconds, and failure to off-chain queues. Research must be able to measure the cost per million anchors and zero-loss Kafka recovery with minimal latency.

6.5 Cross-Industry Validation

The banking (card fraud, mule accounts), retail (refund abuse, triangulation), and telecom (subscription fraud, SIM swaps) should be tested in terms of whether they can be generalized. The same anomaly backbone, temporal windows, peer deviations, graph links, and narrative embeddings can be stored, but not all domain features [30]. The studies need to compile three external datasets of at least five million events each, measure priors independently, and do leave-domain-out training to obtain a measure of transfer. The success criteria will be PR-AUC being within a five percent range of in-domain baselines, Precision @K being constant throughout quarters, and a drop in recall of less than ten percent at the matched reviewer capacity. The investigator panels must also evaluate the clarity of explanations, false-positive burden, and portability of reason codes so that there are no effects of adapting to new regulations and behaviors at the expense of reworking [31].

CONCLUSIONS

This study concludes that an anomaly detection framework based on machine learning, developed specifically, can potentially raise fraud prevention and policy confidence to new levels in practical insurance business. Through the consolidation of transactional statistics, peer group variations, time-based signatures, graphical connectBADs, and narrative embeddings into a controlled feature store, the tactical irregularities are materialized, commonly omitted by a regularly operated beneficial threshold. The design uses chronological validation to avoid leakage, batch and streaming paths parity to avoid feature skew, and strict control over transformations, thresholds, and retraining of models. Performance based on empirical data is conclusive and repeatable. Over 500,000 records, 1-year corpus, by month, unsupervised detectors outperformed supervised and rule baselines in their performance in terms of discrimination and lift: using an autoencoder, the ROC-AUC was about 0.96, and the Isolation Forest was about 0.94, compared to 0.91 on Random Forest. Having an anomaly near the 0.7%, Precision@1000 underwent an increase, making 0.12 under rules and making results 0.31 to 0.34, and Recall at a calibrated operating threshold of 0.76. False positives had been reduced from 42% to 14–16%- an absolute reduction of 26-28% -without incurred latency setbacks. End-to-end inference, including feature look-up, PCA, and scoring, was always within p95 of 120 ms in production-like pilots.

Methodologically, the framework develops the state of practice in three ways. To start with, a standardized feature pipeline improves the quality of data (98% completeness, 95% cross-table consistency, ≈2.5% noise) and dictates homogeneous monitoring with regard to drift and calibration. Second, the sequence pooling with inspirations of memory gives detection additional emphasis upon pivotal sequences such as endorsement bursts, early-life assertions, and non-uniformly distributed inter-event intervals with better sensitivity by holding fixed alert budgets. Third, explainability has been introduced as a control surface: global rankings (as Claim Amount importance 0.31, Policy Tenure 0.25, Age 0.19), local reason codes, and partial-dependence views reduce investigator disposition times and make it auditable-friendly. The limitations persist and encourage further studies. Sparsity is mitigated via privacy, consent, and retention policies that restrict the concept of cross-carrier enrichment, and long-horizon labels are found, and additional supervision and reviewer feedback are used thereto. This will not perfectly demystify collusive networks. Compared to tree-based attributions, deep models' reconstruction-error rationale is less intuitive and hence must be carefully designed and reviewed by the reason-code designer. Population and concept drift of the nature of product mix, channel, and macroeconomic stress require that stability checks be done weekly, there be a calibration of thresholds, and retraining cadences in days instead of quarters. Several non-trivial compute and cost used in always-on scoring and retraining need to be put under control.

The implications are high for the stakeholders of the business. The quoted accuracy improvement and false-positive rate can save approximately $3 million in loss prevention in a year with a medium-sized carrier and increase automation and more favorable triage, saving 20-35% of the number of manual reviews and enhancing indemnity. There is also experience in production where external generalizability is well supported: in the course of over eight weeks, the deployed autoencoder maintained Precision@1,000 ≈0.34 and Recall ≈0.73 at a p95 latency of 120 ms and by two-thirds the false-positive rate of 42% to 14%. Further innovations encompass expanded health, auto, and life-line generalization, streaming with real-time Kafka or Flink, some mix of calibrated model scores and rule construction, and permissioned-ledger anchoring to immutable audit, banking, retail, and telecom cross-industry validation. Every single thread increases equitableness, privacy-by-design, and accountability, and broadens reach and durability. These results are similar to operational limits in straight-through processing, retaining customer experience, and raising investigator acceptance and audit readiness of various product lines worldwide.

REFERENCES

1. Foorthuis, R. (2021). On the nature and types of anomalies: a review of deviations in data. International journal of data science and analytics, 12(4), 297-331.
2. Tewari, S. (2022). Anomaly Detection in Large Scale Data Platforms with Machine Learning.
3. Nyati, S. (2018). Transforming telematics in fleet management: Innovations in asset tracking, efficiency, and communication. International Journal of Science and Research (IJSR), 7(10), 1804-1810. Retrieved from https://www.ijsr.net/getabstract.php?paperid=SR24203184230
4. Poppe, E., Pika, A., Wynn, M. T., Eden, R., Andrews, R., & ter Hofstede, A. (2021). Extracting best-practice using mixed-methods: insights and recommendations from a case study in insurance claims processing. Business and Information Systems Engineering, 63(6), 637-651.
5. Sardana, J. (2022). Scalable systems for healthcare communication: A design perspective. International Journal of Science and Research Archive. https://doi.org/10.30574/ijsra.2022.7.2.0253
6. Baloch, B. K., Kumar, S., Haresh, S., Rehman, A., & Syed, T. (2019, October). Focused anchors loss: Cost-sensitive learning of discriminative features for imbalanced classification. In Asian Conference on Machine Learning (pp. 822-835). PMLR.
7. Henckaerts, R., Côté, M. P., Antonio, K., & Verbelen, R. (2021). Boosting insights in insurance tariff plans with tree-based machine learning methods. North American Actuarial Journal, 25(2), 255-285.
8. Karwa, K. (2023). AI-powered career coaching: Evaluating feedback tools for design students. Indian Journal of Economics & Business. https://www.ashwinanokha.com/ijeb-v22-4-2023.php
9. Chai, C. P. (2023). Comparison of text preprocessing methods. Natural language engineering, 29(3), 509-553.
10. Raju, R. K. (2017). Dynamic memory inference network for natural language inference. International Journal of Science and Research (IJSR), 6(2). https://www.ijsr.net/archive/v6i2/SR24926091431.pdf
11. Singh, V. (2022). Multimodal deep learning: Integrating text, vision, and sensor data: Developing models that can process and understand multiple data modalities simultaneously. International Journal of Research in Information Technology and Computing. https://romanpub.com/ijaetv4-1-2022.php
12. Loftus, J. (2023). An assessment of the effectiveness of using data analytics to predict death claim seasonality and protection policy review lapses in a life insurance company.
13. Konneru, N. M. K. (2021). Integrating security into CI/CD pipelines: A DevSecOps approach with SAST,

DAST, and SCA tools. International Journal of Science and Research Archive. Retrieved from https://ijsra.net/content/role-notification-scheduling-improving-patient

14. Shankar, V., & Parsana, S. (2022). An overview and empirical comparison of natural language processing (NLP) models and an introduction to and empirical application of autoencoder models in marketing. Journal of the Academy of Marketing Science, 50(6), 1324-1350.
15. Kumar, A. (2019). The convergence of predictive analytics in driving business intelligence and enhancing DevOps efficiency. International Journal of Computational Engineering and Management, 6(6), 118-142. Retrieved from https://ijcem.in/wp-content/uploads/THE-CONVERGENCE-OF-PREDICTIVE-ANALYTICS-IN-DRIVING-BUSINESS-INTELLIGENCE-AND-ENHANCING-DEVOPS-EFFICIENCY.pdf
16. Ficara, D., Garroppo, R. G., & Henry, J. (2023). A tutorial on privacy, RCM and its implications in WLAN. IEEE Communications Surveys & Tutorials, 26(2), 1003-1040.
17. Thomas, O. K., Guleri, R., & Baria, S. S. (2021). Proceedings of 'FETP-ICON 2020'Conference. In BMC Proceedings (Vol. 15, No. 11, p. A1).
18. Bayer, S., Kuzmickas, P., Boissy, A., Rose, S. L., & Mercer, M. B. (2021). Categorizing and rating patient complaints: an innovative approach to improve patient experience. Journal of Patient Experience, 8, 2374373521998624.
19. Azcona, E. A. (2022). Geometric Deep Learning in Neuroimaging and Human Reward Behavior (Doctoral dissertation, Northwestern University).
20. Chavan, A., & Romanov, Y. (2023). Managing scalability and cost in microservices architecture: Balancing infinite scalability with financial constraints. Journal of Artificial Intelligence & Cloud Computing, 5, E102. https://doi.org/10.47363/JMHC/2023(5)E102
21. Tok, W. H., Bahree, A., & Filipi, S. (2021). Practical Weak Supervision. " O'Reilly Media, Inc.".
22. Ghazal, R., Malik, A. K., Qadeer, N., Raza, B., Shahid, A. R., & Alquhayz, H. (2020). Intelligent role-based access control model and framework using semantic business roles in multi-domain environments. IEEE access, 8, 12253-12267.
23. Singh, V. (2022). Advanced generative models for 3D multi-object scene generation: Exploring the use of cutting-edge generative models like diffusion models to synthesize complex 3D environments. https://doi.org/10.47363/JAICC/2022(1)E224
24. Nika, K. (2023). Development of IoT Telemetry for Local Control System and Management Software.
25. Qian, B., Su, J., Wen, Z., Jha, D. N., Li, Y., Guan, Y., ... & Ranjan, R. (2020). Orchestrating the development lifecycle of machine learning-based IoT applications: A taxonomy and survey. ACM Computing Surveys (CSUR), 53(4), 1-47.
26. Chavan, A. (2023). Managing scalability and cost in microservices architecture: Balancing infinite scalability with financial constraints. Journal of Artificial Intelligence & Cloud Computing, 2, E264. http://doi.org/10.47363/JAICC/2023(2)E264
27. Sardana, J. (2022). The role of notification scheduling in improving patient outcomes. International Journal of Science and Research Archive. Retrieved from https://ijsra.net/content/role-notification-scheduling-improving-patient
28. Ramesh, V. K. C. (2019). Storing iot data securely in a private ethereum blockchain (Master's thesis, University of Nevada, Las Vegas).
29. Härer, F., & Fill, H. G. (2022). Decentralized attestation and distribution of information using blockchains and multi-protocol storage. IEEE Access, 10, 18035-18054.

30. Cai, L., Chen, Z., Luo, C., Gui, J., Ni, J., Li, D., & Chen, H. (2021, October). Structural temporal graph neural networks for anomaly detection in dynamic graphs. In Proceedings of the 30th ACM international conference on Information & Knowledge Management (pp. 3747-3756).
31. Machireddy, J. R. (2022). Integrating predictive modeling with policy interventions to address fraud, waste, and abuse (fwa) in us healthcare systems. Advances in Computational Systems, Algorithms, and Emerging Technologies, 7(1), 35-65.