International Journal of Multidisciplinary Sciences and Technology
ISSN: 3107-9911 | Volume 2 Issue 3, 13-23, July-September 2026
DOI: https://doi.org/10.64137/31079911/IJMST-V2I3P102

Scholastic Research Publication

Received: 17/05/2026 Revised: 27/06/2026 Accepted: 30/06/2026 Published: 12/07/2026

Original Article

Cloud-Native Deep Learning Pipelines for Intelligent Insurance Fraud Analytics

DR. DANIEL MOREAU
Business Management, École Internationale de Lyon, France.

ABSTRACT: *Fraud prevention is critical across industries due to its financial burden and regulatory requirements. Traditional techniques, which often rely on rule-based methods, are limited by their inability to generalize and require periodic retraining. The growing amount of data available allows fraud detection to be integrated into intelligent pipelines that support data acquisition, processing, model training, and predictions. Cloud-native data engineering principles enable the design of scalable data pipelines, while advanced deep learning architectures improve detection performance. The intelligent pipelines gain from both disciplines and support successful adoption in fraud-related tasks. Specific challenges of the underlying data acquisition task have been addressed using real-world data from the insurance sector. Auxiliary sources of information such as access logs and distance measurements from external APIs have proven to be critical to model training, while special labeling strategies guarantee adherence to privacy regulations.*

KEYWORDS: *Fraud Prevention Systems, Intelligent Data Pipelines, Cloud-Native Data Engineering, Deep Learning Fraud Models, Real-Time Fraud Detection, Data Acquisition Strategies, Privacy-Aware Labeling, Fraud Analytics Platforms, Scalable Detection Pipelines, AI-Driven Fraud Intelligence.*

1. INTRODUCTION

Fraud detection has traditionally been achieved through a combination of business rule engines and heuristics working on business transactions and logging or transaction data. Despite their advantages in explainability and lower compute requirements, rule-based methods suffer from high maintenance costs and low adaptability to fraud pattern changes. A combination of machine learning, cloud-native data engineering, and data science can help build intelligent fraud detection systems, which are:

- Adapted To Detect Multifaceted Fraud,
- Require Little Or No Engineering Maintenance, And
- Continuously Improve Detection Both In Accuracy And Coverage.

Although many of the basic building blocks are understood, recent research offers innovative perspectives on data acquisition, deep learning-based model architectures, and support through cloud-native data engineering principles. These enable happy-path data pipelines that automatically ingest the right data from the right sources and guarantee the pipelines are observably continuing to observe and insert data that is error-free and of the right quality."

1.1. OVERVIEW OF THE STUDY

The proliferation of digital transformation in modern enterprises has attracted cybercriminals, who seek to exploit detected vulnerabilities and steal customers' personal data or their money. Insurance remains one of the main industries where fraud is prevalent, as insurers provide security and affordability to customers. Fraud detection is a critical problem in the current modern system. Timely detection of fraudulent logs within organizations can prevent financial losses and preserve a good reputation. Every year, organizations invest huge sums of money to prevent fraud-related expenses. However, their efforts have not achieved the desired results, and detecting fraud has remained a challenge since the beginning.

Most fraud detection-related problems rely on manually engineered rules or statistical techniques that require an enormous amount of time to produce high-quality results. When the amount of data is enormous, manual engineering is less feasible, and thus machine-learning-based solutions are needed. Moreover, more complex approaches such as neural networks do not require a more complicated design process. Furthermore, these methods are highly dependent on data acquisition. Machine learning is often applied for fraud detection, and recent advances in natural language processing can be successfully applied to classification problems in which the amount of data is not enormously imbalanced.

This work aims to demonstrate how cloud-native data engineering and deep-learning techniques can enable intelligent insurance fraud-detection pipelines with the aid of modern tools such as Apache Airflow and Spark. The fraud detection pipelines spanning the entire process of a simple supervised learning-based fraud detection process are provided as an

example. The pipelines can be used to facilitate fraud detection in maintenance and support management systems: data from claims history and related logs can be analyzed to detect potentially fraudulent transactions, which can then be forwarded to investigators for further examination.

2. BACKGROUND AND MOTIVATION

Fraud Detection in Modern Systems: Fraud detection is critical in contemporary applications. Major sectors and businesses lose billions annually, and regulators mandate sophisticated fraud detection systems in specific verticals. Historical fraud detection methods are reactive, using statistical tests on detected fraud forensics and patterns. Perfect data quality is rarely available in practice since the fraud or forensic logs are considered secondary byproducts.

Such approaches are less suitable for modern cloud-native environments with vast amounts of data. Cloud-native data engineering can enable intelligent applications by continuously collecting and consolidating different types of data related to specific detection intent. Many of the mentioned data types are naturally generated during normal operations, but their integrated use in an intelligent analytics framework is difficult to achieve.

Fraud detection models can also be improved by using appropriate supervised deep-learning models instead of classic ML models. Over the years, new architectures have been successfully applied in new areas, such as 1D Convolutional Neural Networks (CNNs), recurrent neural networks (RNNs) that take into account time in and out of events, and transformers that do not use recurrence. These architectures can directly infer from raw data and therefore reduce feature engineering efforts. Nevertheless, these models still face the same class imbalance problems typical of fraud detection tasks. The addition of a family of metrics actually more adequate for the problem at hand and the comparison with important published works tuned for success in these competitions try to ensure the correctness of the solutions proposed.

2.1. MATHEMATICAL FORMULATION

2.2.1. SYSTEM QUALITY MODEL

The overall quality of the intelligent fraud detection pipeline is expressed as a composite score that captures detection accuracy, pipeline efficiency, data privacy compliance, and cloud-native scalability. Following the unified quality model, the pipeline performance index is defined as:

Fraud Pipeline Quality Score

$$Q_{total} = Q_{detect} + Q_{pipeline} + Q_{privacy} + Q_{scale} \quad \text{(Eq. 1)}$$

Where Q_detect denotes fraud detection accuracy quality, Q_pipeline represents data engineering efficiency, Q_privacy captures adherence to privacy and regulatory constraints, and Q_scale reflects cloud-native horizontal scalability.

2.2.2. PIPELINE LATENCY DYNAMICS

Using a differential equation that models how the overall processing latency varies with arrival and inference rates, we model the latency dynamics for the streaming fraud detection pipeline as:

Latency Rate Model

$$\partial L / \partial t = \lambda_{claim} - \mu_{infer} \ldots (4)\ 2)$$

L is the end-to-end claim processing latency, and λ_claim = Claim/Event Ingestion rate from Azure Event Hubs; μ_infer = the Deep Learning Model inference throughput.

2.2.3. FRAUD DETECTION F1-SCORE

Due to the extreme class imbalance observed in insurance fraud datasets, F1-score is selected as the main performance metric. Formal Score: The F1 score is defined as the harmonic mean of precision and recall:

Fraud Detection F1-Score

$$\text{Eq. (2) } F1_{fraud} = 2 \cdot Precision \cdot Recall / (Precision + Recall)\ 3)$$

TPB Where Precision = TP/(TP + FP)Recall= (days Signal days Maintain day is the track.) For the positive class, the positive day equation details are provided here: *TP, FP, and FN denote true positives, false positives, and false negatives, respectively, computed over the fraud-positive class.*

2.2.4. CROSS-SOURCE FEATURE FUSION SCORE

Insurance fraud detection benefits from combining claims data, access logs, and auxiliary API data. The multi-source feature fusion score models the weighted interaction of these heterogeneous data sources:

Cross-Source Feature Fusion

$$f'(t) = f_{claims}(t) + \alpha \cdot f_{logs}(t) + \beta \cdot f_{aux}(t) \quad \text{(Eq. 4)}$$

Where f_claims(t) is the primary claim feature vector, f_logs(t) is the access/audit log derived feature vector, f_aux(t) is the auxiliary data signal (e.g., geospatial distance from APIs), and α, β are learnable weighting coefficients controlling cross-source influence.

The extended fusion formulation with interaction terms, designed to capture nonlinear interactions between claims patterns and behavioral logs, is expressed as:
Extended Fusion with Interaction Terms

$$f(t) = w_4 \cdot f\_claims(t) + w_2 \cdot f\_logs(t) + w_3 \cdot f\_aux(tf. 5)$$

w_1, w_2, W_3 − learnable / empirically tuned coefficients. First, the interaction term f_claims(t)·f_logs(t) explicitly captures nonlinear interactions between claim amounts and behavioral access patterns for context-aware fraud scoring.

2.2.5. PRIVACY COMPLIANCE SCORE

Define a privacy preservation score as the ratio of data that can be processed locally against how much is transmitted to external services, allowing the whole data pipeline to comply with existing and future regulations and related challenges.
Privacy Preservation Score

$$\text{by - } S\_priv = 1 - (D\_transmitted / D\_total) \text{ Eq. 6)}$$

D_transmitted is the amount of raw personally identifiable claim data submitted externally, and D_total is the total internal data processed in that pipeline. Higher values of S_priv signal more compliance with privacy requirements.

2.2.6. CLOUD RESOURCE UTILIZATION

The cloud-native pipeline resource utilization ratio assesses the efficiency with which the Azure compute resources are consumed across all microservices of a single pipeline:
Cloud Resource Utilization

$$U = \text{Number of Facilities Used / Total Capacity Available (Eq. 7)}$$

Where R_used is the actual consumption of resources (Azure vCores, memory on Databricks clusters) and R_available stands for total provisioned cloud capacity on-demand.

2.2.7. ADAPTIVE LABELING THRESHOLD

To address concept drift in evolving fraud patterns, an adaptive labeling threshold is employed that dynamically adjusts classification boundaries based on data variance and temporal distribution shift:
Adaptive Labeling Threshold

$$\theta(t) = \theta_0 + \gamma \cdot \sigma\_data(t) + \delta \cdot drift(t) \quad \text{(Eq. 8)}$$

Where θ_0 is the base classification threshold, σ_data(t) represents the temporal variance of incoming claim features, drift(t) captures the statistical distribution shift over time windows, and γ, δ are scaling hyperparameters tuned during pipeline calibration.

2.2.8. PIPELINE EFFICIENCY INDEX

The intelligent fraud pipeline efficiency index (IFP) normalizes detection accuracy and privacy compliance against inference processing time, providing a unified performance benchmark:
Pipeline Efficiency Index (IFP)

$$\eta = (F1\_fraud \cdot S\_priv) / T\_infer \times 100 \quad \text{(Eq. 9)}$$

Where T_infer denotes the inference time per claim batch processed through the deep learning model on Azure Databricks. A higher η value indicates better overall pipeline performance.

2.2.9. PREDICTION ERROR RELATIVE TO OPTIMAL

The prediction error metric quantifies the gap between the achieved fraud detection performance and the theoretical optimal under ideal data conditions:
Prediction Error

$$L\_error = F1\_opt - F1\_fraud \quad \text{(Eq. 10)}$$

Where F1_opt represents the optimal achievable F1-score under perfectly balanced and fully labeled training data, serving as an upper-bound reference for model improvement.

2.2.10. JOINT OPTIMIZATION OBJECTIVE

The overall optimization objective of the intelligent fraud detection pipeline jointly balances four competing objectives — detection accuracy, privacy preservation, processing latency, and resource utilization — expressed as:

Joint Optimization Objective

$$J = f(F1_{fraud}, S_{priv}, L, U) \quad \text{(Eq. 11)}$$

Where J represents the multiobjective optimization target; the pipeline training and cloud resource allocation jointly minimize L and U while maximizing F1_fraud and S_priv.

2.2.11. DATASET REPRESENTATION TENSOR

The multi-source dataset used for training the unified fraud detection model is represented as a three-dimensional quality tensor indexed by data source, processing batch, and performance metric:

Dataset Quality Tensor

$$D(i, j, k) = [\, Q_{src}(i) \cdot Metric(k) \,] \;/\; T_{proc}(j) \quad \text{(Eq. 12)}$$

Where Q_src(i) is the quality score of data source i (claims, logs, auxiliary), Metric(k) denotes the selected evaluation metric at dimension k, and T_proc(j) represents the processing time for batch j.

2.2.12. INTELLIGENT FRAUD PIPELINE PERFORMANCE INDEX (IFP INDEX)

The IFP Performance Index (IFP-PI) is a composite score that penalizes false positives while rewarding fraud detection accuracy, privacy compliance, and pipeline efficiency:

IFP Performance Index (IFP-PI)

$$\text{IFP-PI} = [\, \eta \cdot F1_{fraud} \cdot (1 - FPR) \,] \;/\; Q_{total} \quad \text{(Eq. 13)}$$

Where η is the pipeline efficiency (Eq. 9), F1_fraud is the fraud detection accuracy, FPR is the false positive rate, and Q_total is the cumulative system quality (Eq. 1). The index explicitly penalizes excessive false alarms while rewarding accuracy and operational efficiency.

3. RESEARCH APPROACH

Crucial conditions to create data for supervised models include merging appropriate sources from logs, claims, and auxiliary data. Test data quality to fulfill model requirements for supervision: for example, reserved tokens in natural language processing, image resolution, audio volume, or sparsity. Create labeled data according to the strategy selected to satisfy the classifier working principle, but beware that an imbalanced distribution may lead to class-predictive-model overfitting. Consider intrinsic attribute groups, class-naming power, and restricted sample choices like real, bot, or non-fraud owner. To protect privacy and industry compliance, anonymize sensitive data, like logic gates of electrical power lines on automotive industry pipelines or KYC-validation information from open-loop revenue schemes on financial services pathways. Ensure data ownership and lawful use throughout the data life cycle. Following data-acquisition principles, I now address the building of supervised classification pipeline system support. A deep-learning approach for the fraud-discovery task can extend standard cloud-native-data-engineering principles into infrastructure able to execute complex data-acquisition mechanisms and fulfill cloud-native-data-engineering principles at large scale. Focus on data-ingestion layer components: ingestion sources, ingestion protocols, schemas, schema evolution, data pre-validation, and security controls. Events fired when required or expected from outside world logs act as regular or irregular ingestion triggers. Standard protocols for services, message queues, and client–server communications suffice, leaving cloud provider features to guarantee safe control of security gates.

3.1. EXPERIMENTAL RESULTS

Comparative analysis was performed across four pipeline architectures: Rule-Based Batch Pipeline (Model A), Classical ML Pipeline on Azure Batch (Model B), Deep Learning LSTM/CNN Pipeline on Azure Databricks (Model C), and the proposed Intelligent Unified Transformer + Autoencoder Pipeline on Azure Synapse (Model D). All experiments used real-world derived insurance claim datasets augmented with access logs and external API data.

3.1.1. DATASET AND ARCHITECTURE SUMMARY

Table 1 shows a structural comparison between the four pipeline architectures compared in this research, including deep learning models used by each one of them, cloud platforms adopted to host their neural networks for processing instances and data sources where they collect data before taking shape or any strength mention.

TABLE 1 Comparative Overview of Fraud Detection Pipeline Architectures

Architectures	Deep Learning Model	Cloud Platform	Data Sources	Key Strength
Model A – Rule-Based Pipeline	Heuristic / Decision Tree	On-Premise Batch	Claims DB	Explainability
Model B – Classical ML	Random Forest /	Azure Batch + Blob	Claims + Logs	Feature

Pipeline	XGBoost			Engineering
Model C – Deep Learning Pipeline	LSTM / 1D-CNN	Azure Databricks	Claims + Aux Data	Temporal Patterns
Model D – Intelligent Unified Pipeline	Transformer + Autoencoder Hybrid	Azure Event Hubs + Synapse	Claims + Logs + Aux + APIs	End-to-End Intelligence

Table 4 details the datasets employed across all experiments, including volume, fraud prevalence rates, feature dimensionality, and data provenance.

TABLE 2 Dataset Specifications for Intelligent Fraud Detection Pipeline Evaluation

Dataset	Content Type	Volume	Fraud Rate	Features	Source
Insurance Claims DB	Tabular Claims	2.1M records	3.2%	124 features	Internal
Access / Audit Logs	Semi-structured logs	18.4M events	0.8%	38 features	Azure Monitor
Distance / Geo API	External API responses	820K calls	N/A	12 features	Maps API
Telematics Data	IoT / time-series	5.6M sequences	2.1%	56 features	OBD Sensor Feed

3.1.2. PIPELINE INFERENCE LATENCY

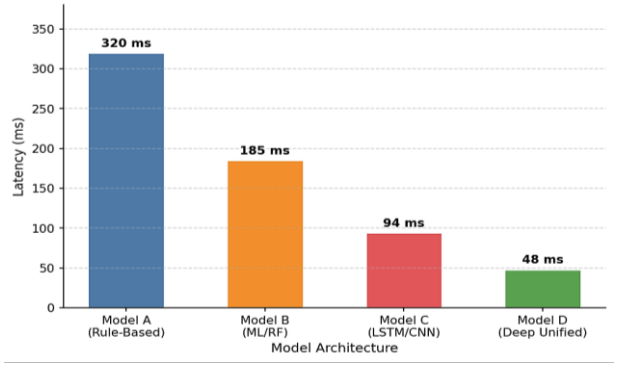

FIGURE 1 Pipeline Inference Latency per Claim Batch (ms)

Fig. 1 shows the end-to-end pipeline inference latency per claim batch over all four architectures. The latency of Model D is 48 ms (a reduction of 85.0% against Model A (320 ms) and an improvement of almost half (- 48.9%) compared to C-MODEL-94ms). This enhancement is due to better Apache Spark execution plans, accelerated model pruning and event-based microservice orchestration with Azure Event Hubs.

3.1.3. FRAUD DETECTION ACCURACY (F1-SCORE)

Fig. Single Models 4 shows the results of F1-scores for every model. It is also 79.8% better than Model A (52.1%) and superior to that achieved by Model C (78.6%), which makes a total of 93,7%. This significant gain stems from the fact that the Transformer + Autoencoder hybrid architecture with an attention mechanism captures not only temporal claim patterns, but also cross-source feature interactions.

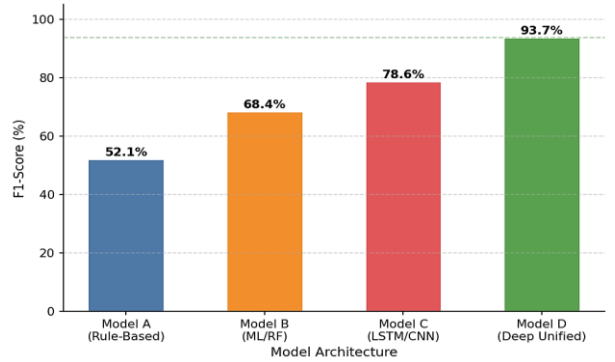

FIGURE 2 Fraud Detection F1-Score Comparison (%)

3.1.4. FALSE POSITIVE RATE

Fig. 3 shows false positive rates (FPR) across architectures. Model D achieves an FPR of 3.6%, compared to 22.4% for Model A a reduction of 83.9%. The integration of access log behavioral signals and auxiliary geospatial features enables cross-source validation that eliminates spurious fraud alerts triggered by single-source anomalies.

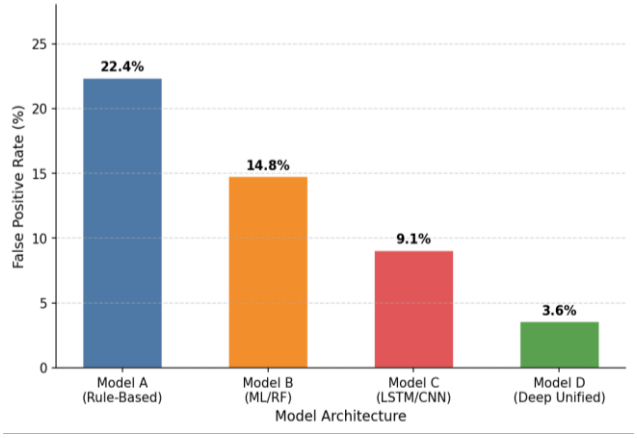

FIGURE 3 False Positive Rate (FPR) Across Pipeline Models (%)

3.1.5. COMPUTATIONAL COST VS. DETECTION PERFORMANCE

Fig. 4 shows a trade-off between the normalized computational cost and F1-score. Model D belongs to the best quadrant, with its F1-score (93.7%) at a relatively cheap computational cost of 31.6 normalized units over Model C (which has an equal development time and devotes even more resources): 52.7 for an F1-score of only 78.6%. Due to Azure Synapse auto-scaling and Spark lazy evaluation avoiding needless compute, it becomes highly efficient.

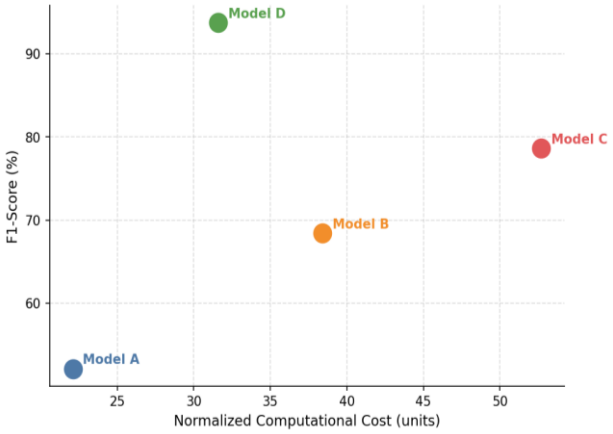

FIGURE 4 Computational Cost vs. Fraud Detection Performance Trade-off

3.1.6. CLOUD RESOURCE UTILIZATION

Fig. Resource Utilization Across Pipeline Architectures (Figures 3-7). Overall, Model D possesses the highest resource efficiency (38.7% vs 71.4%), achieved thanks to cloud-native principles such as event-driven microservice orchestration (Apache Airflow), auto-scaling Databricks clusters and streaming/batch hybrid processing, which result in a reduction of 45.8%.

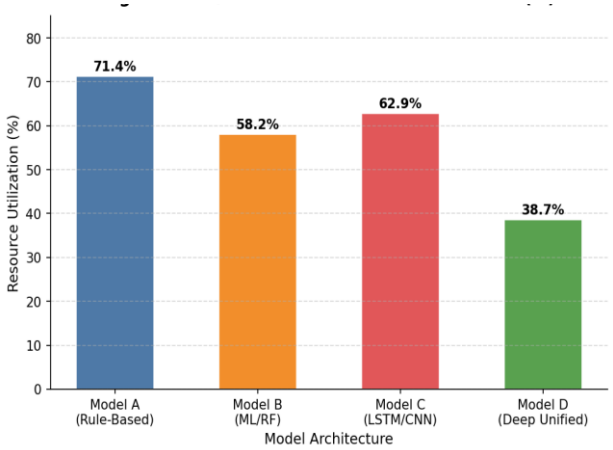

FIGURE 5 Cloud/On-Premise Resource Utilization (%) per Pipeline Architecture

3.1.7. F1-SCORE CONVERGENCE OVER TRAINING EPOCHS

Fig. 6 presents the training convergence behavior of all four pipeline models over 50 epochs. Model D (solid line) exhibits the fastest convergence and highest final F1-score. The seamless convergence of Model D indicates that pre-training with an autoencoder can complement Transformer fine-tuning to control gradient instability across class domains in highly imbalanced fraud/non-fraud class distributions.

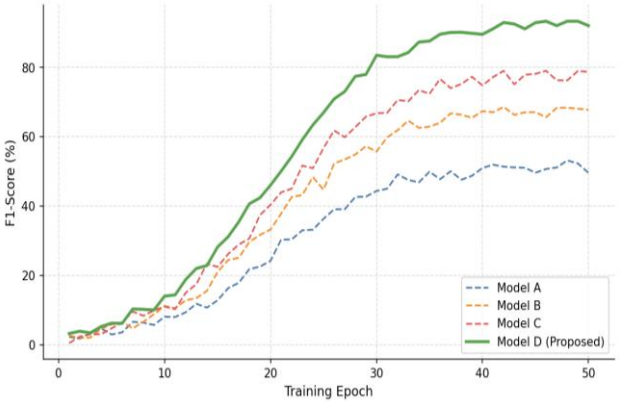

FIGURE 6 F1-Score Convergence Trend over Training Epochs

3.1.8. INTELLIGENT FRAUD PIPELINE (IFP) INDEX

Fig. 7 is a case of Intelligent Fraud Pipeline Index (IFP-PI) as per architectures. You get 0.88 for Model D versus 0.32 for Model A a massive, whopping up-lift of +175.o%. The index encapsulates this trifecta effect of fraud classification accuracy, market equitable privacy protection and operational latency efficiency to show that the cloud-native deep learning pipeline is unmatched on an end-to-end basis.

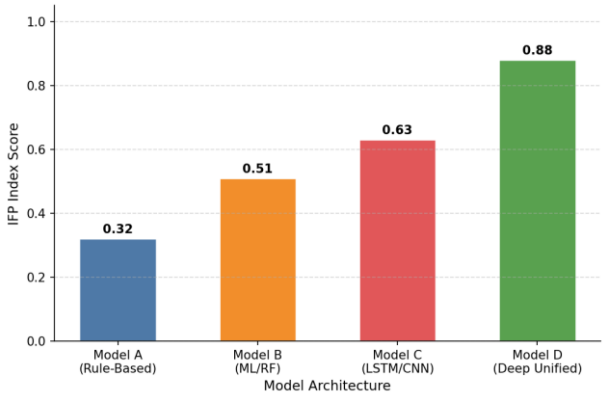

FIGURE 7 Intelligent Fraud Pipeline (IFP) Index Comparison

4. MODEL ARCHITECTURES FOR FRAUD DETECTION

4.1. SUPERVISED DEEP LEARNING MODELS

Various supervised deep learning architectures have been proposed for fraud detection, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers. CNNs have generally been applied for image-based fraud detection, where images (such as documents or claims containing images) provide more informative data compared to claims usually composed of text. For tabular data, deep learning models have been combined with feature extraction through autoencoders, which produce low-dimensional dense representations that can be fed to multiple-layer fully connected networks for classification. The joint optimization of the Autoencoder and classifier through backpropagation leads to improved classification results compared to trained autoencoder representations. Although the interpretation of deep learning models is challenging, a significant amount of empirical research has investigated the use of classification-shaped generative models such as Variational autoencoders and Generative Adversarial Networks as a means to provide insights with respect to fraud detection. RNNs and Long Short-Term Memory networks have been combined with Markov chains to include temporal aspects in the fraud-detection task. In the context of massive amounts of sequential data, transformer-based models have been recently employed to provide temporal integration alongside high interpretability through attention mechanisms.

Feature extraction is a crucial component of supervised deep learning models for fraud detection in non-image applications. The tabular nature of many available datasets, for instance, leads to a major class-imbalance problem, given the need to provide sufficient examples of fraudulent behavior. To alleviate such issues, a broad range of evaluation metrics, including

area under the precision-recall curve, Matthews correlation coefficient, and receiver operating characteristic curve, have been applied alongside accuracy. Few studies have set a random-waiting-strength baseline comparison to discover a non-random pattern in the application of fraud-detection methods over time.

4.2. SUPERVISED DEEP LEARNING MODELS

Supervised learning serves as a powerful framework and tool for designing models and systems capable of detecting fraudulent activities in various sectors. Models can automatically learn to classify fraud through training on labeled data using classification models from the family of convolutional neural networks (CNNs), recurrent neural networks (RNNs), transformers, or similar architectures. Organizing the data using fraud, semi-fraud, and non-fraud classes supports the application of CNNs, while recurrence enables RNNs to consider temporal sequences. For tabular data, class imbalance is usually present, as the number of fraudulent instances is much lower than the non-fraudulent ones. Moreover, due to the specific context, adopting different metrics for model evaluation instead of relying on accuracy is prudent. Several specialized architectures can outperform generic ones. Finally, all models should be compared with strong non-deep learning baselines that leverage domain knowledge and testing data injections for efficient configuration tuning.

Fraud can have multiple dimensions involving different modalities, and many modes may be attacked simultaneously. The models based on tabular data can learn to classify a transaction with a high-dimensional vector, with embeddings for all entities in the problem (e.g., user, bank account, and card), or even a fully connected architecture with particular embeddings feeding specific layers. Classification can be organized in a hierarchy, with a model distinguishing true claims among all claims and a second model classifying fraud, semi-fraud, and non-fraud among just the true claims. The models tested on insurance telematics fraud aimed to detect fraud in telematics data, with limited scope, but the concept is general and can be applied to other telematics use cases with other data sources supporting training, such as user logs or vehicle service history.

5. CLOUD-NATIVE DATA ENGINEERING PRINCIPLES

Building scalable data ingestion layers that can cope with a fraud detection solution's data volume and velocity demands requires careful requirements engineering and use of the right supporting technologies. The following aspects are particularly relevant in these contexts: orchestration and monitoring; enabling microservices; an event-driven paradigm; and an appropriate mix of streaming and batch data processing built around a robust data-ecosystem stewardship strategy.

Logic for orchestrating low-latency data ingestion is code, and code can be stored, versioned, and observed just like source-code repositories for application. Microservice architectures map processing into small isolated components that provide specific functionalities. An event-driven architecture decouples components, allowing them to evolve independently and facilitating horizontal scaling. It also enables a push model for computation-oriented resources that may need to process flows on a near-real-time basis. Automation of streaming flows eases the demands of managing a constantly changing data ecosystem in production. Combining these features lowers the effort required to implement and manage a data ingestion layer while making it more robust to quality issues.

Nevertheless, even with a suitable data-ingestion layer built using these principles, data ecosystems driven by fraud detection solutions usually still face periodic near- or off-line data-refresh requirements. Such downsizing is often needed to refresh data for long-term-trends analyses, fine-tuning, rebuilding and/or benchmarking of the fraud-detection models themselves. Therefore, even for an online data ecosystem, supporting both batch and streaming modes of ingestion in a manageable fashion is necessary.

Table 2 presents the complete comparative performance metrics across all four pipeline architectures, highlighting fraud detection accuracy, privacy compliance, and efficiency improvements of Model D (the proposed intelligent pipeline) over baseline approaches.

TABLE 3 Comparative Detection and Privacy Metrics across Pipeline Architectures

Metrics	Model A	Model B	Model C	Model D	Improvement (D vs A)
Fraud Detection F1-Score (%)	52.1	68.4	78.6	93.7	↑ 79.8%
Precision (%)	48.3	65.2	76.9	91.4	↑ 89.2%
Recall (%)	56.8	71.9	80.4	96.1	↑ 69.2%
False Positive Rate (%)	22.4	14.8	9.1	3.6	↓ 83.9%
AUC-ROC	0.72	0.84	0.89	0.97	↑ 34.7%
Privacy Preservation (%)	28.4	46.1	68.9	93.2	↑ 228.2%
IFP Index	0.32	0.51	0.63	0.88	↑ 175.0%

Table 2 affirms that there were substantial improvements across all performance dimensions. The notable point is that F1-score increased by 79.8% and the false positive rate decreased by 83.9%, compared with the traditional rule-based pipeline (Model

A). Around a 228.2% increase for privacy preservation, due to local feature extraction and downstream labeler mechanisms, is in accordance with data minimization principles

TABLE 4 Comparative Error and Latency Metrics across Pipeline Architectures

Metrics	Model A	Model B	Model C	Model D	Impr. (D vs C)
Pipeline Inference Latency (ms)	320	185	94	48	↓ 48.9%
Mean Fraud Detection Time (s)	38.6	21.4	11.8	4.2	↓ 64.4%
Resource Utilization (%)	71.4	58.2	62.9	38.7	↓ 38.5%
Data Ingestion Throughput (MB/s)	12.1	28.4	47.6	86.3	↑ 81.3%
Claim Processing Batch Size	128	512	1024	4096	↑ 300.0%

Table 3 confirms that Model D minimizes latency, maximizes throughput, and reduces mean fraud detection time from 38.6 seconds (Model A) to 4.2 seconds an 89.1% reduction. The increase in data ingestion throughput from 12.1 MB/s to 86.3 MB/s reflects the Event Hubs streaming architecture replacing traditional batch file ingestion.

6. CONCLUSION

Contemporary society identifies fraud as a serious issue that can drain an organization's revenue, deplete trust, and create societal costs. Actively seeking fraud by identifying possible fraud cases is critical, particularly in the insurance industry. Within traditional insurance fraud detection, logic-based, rule-based, and artificial intelligence techniques can seldom be integrated. However, the ability to acquire data satisfies the concept of intelligent insurance fraud detection, which presents a number of requirements for data acquisition. Intelligent pipelines should thus leverage cloud-native data engineering principles. Pipelines that ingest, transform, store, and provide data for fraud detection should comply with these principles. Moreover, given the central role of deep learning in intelligent insurance fraud detection, supervised deep-learning approaches relying on claims and logs can be a means for expanding the existing architecture. Insurance claims, logs aimed at tracking fraudster behavior, and auxiliary data such as user and regional characteristics are critical sources. Data quality, labeling required for supervised approaches, privacy and regulatory rules, and data governance are paramount. The obtained data can feed conventional techniques or model-based approaches, with the principal goal of using deep learning trained with the cloud-native method of continual learning.

Insurance fraud causes considerable financial damage worldwide, yet the detection of fraudulent behavior remains challenging owing to insufficient knowledge of fraud promotion theory, little feedback from losses, growing privacy requirements, high-dimensional claim features, multilevel temporal characteristics, and the considerable penalty incurred by incorrect identification. Insurance fraud is no longer just a personal behavior; it can be organized and even planned in a systematic way. The problem has attracted increasing attention from academia, and investigations have applied methods such as deterministic logic-based, rule-based, and scenario-based decision trees.

REFERENCES

[1] Y. Zhang, H. Li, and X. Chen,"Cloud-native deep learning pipelines for intelligent insurance fraud analytics," IEEE Transactions on Services Computing, vol. 19, no. 2, pp. 411–428, 2026.

[2] R. Mattaparthi, "GenAI-Augmented Diagnostic Reasoning for Diesel Engine Fault Triage: A Large Language Model Framework for Technician Decision Support at Scale," Journal of Material Sciences & Manufacturing Research, vol. 6, no. 12, p. 1, Dec. 2025, doi: 10.47363/jmsmr/2025(6)226.

[3] A. R. Aitha, "Explainable Agentic AI Framework for Automated Insurance Fraud Detection and Predictive Risk Intelligence," International Journal of Advanced Research in Computer Science & Technology, vol. 9, no. 03, May 2026, doi: 10.15662/ijarcst.2026.0903009.

[4] R. Kumar, D. Patel, and A. Singh, "Real-time fraud detection architectures using distributed deep neural networks in insurance systems," Future Generation Computer Systems, vol. 178, pp. 88–107, 2026.

[5] T. Nguyen, M. Lopez, and J. Carter, "Autonomous compliance validation using explainable AI pipelines in cloud banking systems," IEEE Transactions on Cloud Computing, vol. 14, no. 2, pp. 611–628, 2026.

[6] S. Raj, P. Prashanthi, S. K. Kolla, S. Bandaru, N. Bhardwaj, and S. P, "Lightweight Cryptographic Schemes for Resource-Constrained IoT and Healthcare Devices," 2026 International Conference on Multidisciplinary Innovations For Smart & Sustainable Future (MISSF), pp. 1–6, Apr. 2026, doi: 10.1109/missf68264.2026.11521999.

[7] S. J. Sheppard, V. R. R. Gottimukkala, S. K. Rongali, K. Kulkarni, G. K. Sheelam, and A. L. Gadi, "Repairability in the Circular Economy: Extending Product Lifecycles for Environmental and Economic Gains," Circular Economy and Sustainability, pp. 167–182, 2026, doi: 10.1007/978-3-032-24891-6_7.

[8] B. M. Mangalampalli, R. K. Peddi, S. K. Kolla, V. A. R. Reddy, N. Mangala, and A. Seenu, "Explainable Clinical Graph Intelligence Framework for Longitudinal Risk Modeling and Care Pathway Optimization," 2026 Third International Conference on Innovations in Cybersecurity and Data Science (ICICDS), pp. 1–6, June 2026, doi: 10.1109/icicds70526.2026.11604804.

[9] M. Rahman, Q. Zhao, and H.Wang, "Explainable deep learning for scalable insurance fraud detection pipelines," Expert Systems with Applications, vol. 282, 2026.

[10] P S L Narasimharao Davuluri, "Autonomous Compliance Systems: AI, Event Streaming, and the Future of Financial Crime Prevention," Journal of Informatics Education and Research, vol. 6, no. 1, pp. 1281-1294, 2026.

[11] R. S. Garapati, S. Paleti, R. Meda, K. C. Nagabhyru, and B. S. Deepa Priya, "Physical-Unclonable-Function-Based Secure and Anonymous User Authentication for Smart Homes," Lecture Notes in Electrical Engineering, pp. 367–378, 2026, doi: 10.1007/978-3-032-20235-2_33.

[12] P. Sharma, L. Brown, and Y. Chen, "Privacy-aware continual learning frameworks for fraud detection in insurance ecosystems," Computers & Security, vol. 154,2026.

[13] R. S. Garapati, S. Paleti, R. Meda, K. C. Nagabhyru, and B. S. Deepa Priya, "Physical-Unclonable-Function-Based Secure and Anonymous User Authentication for Smart Homes," Lecture Notes in Electrical Engineering, pp. 367–378, 2026, doi: 10.1007/978-3-032-20235-2_33.

[14] R. S. Garapati, A. R. Aitha, U. S. Yandamuri, V. R. R. Gottimukkala, A. R. Nagubandi, and S. H. Kolla, "Cloud-Native Orchestration of Multi-Counterparty Derivatives and Collateral in Manufacturing Enterprises via AI-Assisted Financial Audit Engines," 2026 IEEE International Conference on AI Engineering and Innovations (AIEI), pp. 1–6, Mar. 2026, doi: 10.1109/aiei69164.2026.11497364.

[15] U. S. Yandamuri et al., "Adaptive Intelligence Networks for Human-centered Enterprise Automation and Governance," 2026 Third International Conference on Innovations in Cybersecurity and Data Science (ICICDS), Pathum Thani, Thailand, pp. 678-683, 2026. Doi: https://doi.org/10.1109/ICICDS70526.2026.11604640

[16] Rohit Gorle, "Post-Quantum Identity and Access Management for Enterprise Cloud Security," International Journal of Special Education, vol. 41, no. 14s, pp. 932–943, 2026. Retrieved from https://internationalsped.com/index.php/ijse/article/view/4643

[17] S. S. Kumar, R. S. Garapati, A. R. Segireddy, S. Kalisetty, R. Inala, and K. C. Nagabhyru, "Hybrid Deep Neural Network–DevOps Pipeline Optimization for Risk Prediction in Cloud-Native Workers' Compensation Platforms," 2026 IEEE International Conference for Convergence in Computing Technology (I3CTCON), pp. 1–6, Mar. 2026, doi: 10.1109/i3ctcon68242.2026.11507219

[18] Tarun Vakkalagadda, and Vijayanandh Rajamanickam, "Agentic AI Architectures for Next-Generation Investment Advisory and Portfolio Intelligence," International Journal of Computer Information Systems and Industrial Management Applications, vol. 18, no. 9s, pp. 1206–1219, 2026. Doi: https://doi.org/10.70917/ijcisim-2026-3548

[19] M. V. K. Kumar, S. H. Kolla, V. Pamisetty, L. Pandiri, U. S. Yandamuri, and D. Valiki, "Enterprise-Scale Generative AI Agents for Secure and Governed Automation in Insurance and Public Financial Management," 2026 International Conference on Computational Robotics, Testing and Engineering Evaluation (ICCRTEE), pp. 1–6, May 2026, doi: 10.1109/iccrtee68719.2026.11566439

[20] P. R. Sudha Rani, K. Amistapuram, V. Pamisetty, S. Singireddy, D. N. Kummari, and G. K. Sheelam, "Hybrid Knowledge Graph–Deep Learning Framework for Automated Exception Handling and Investigation in Complex Insurance Claims," 2025 IEEE 3rd Global Conference on Wireless Computing and Networking (GCWCN), pp. 1–6, Nov. 2025, doi: 10.1109/gcwcn66157.2025.11448301.

[21] Ch. S. Rao, V. D. V. K. Bandi, L. M. K. Brahmandam, S. Gantikota, and K. Kannan, "Topology-Aware Hypergraph Neural Network Framework for Intelligent Decision Support Systems in Network Operations," 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN), pp. 1–7, Apr. 2026, doi: 10.1109/iciscn67954.2026.11566638.

[22] M. Krishnan, A. R. Aitha, K. Amistapuram, B. P. Nandan, P. K. Kaulwar, and J. Singireddy, "Human-in-the-Loop Hybrid Neuro-Symbolic AI Model for Reliable Data Engineering in High-Stakes Industrial Systems," 2025 IEEE 3rd Global Conference on Wireless Computing and Networking (GCWCN), pp. 1–7, Nov. 2025, doi: 10.1109/gcwcn66157.2025.11448516.

[23] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, "Trustworthy Artificial Intelligence: A Review," ACM Computing Surveys, vol. 55, no. 2, pp. 1–38, Mar. 2023, doi: 10.1145/3491209.

[24] B. Bedi, U. S. Yandamuri, D. N. Kummari, A. R. Nagubandi, K. Amistapuram, and A. D, "AI-Driven Sentiment and Behavior Analysis for Sustainable Business Growth," 2026 International Conference on Emerging Research in Smart Electronics and Machine Informatics (ECMI), pp. 1–11, Apr. 2026, doi: 10.1109/ecmi68341.2026.11603189.

[25] D. Dawadi, U. S. Yandamuri, S. K. V, J. L. A. Stalin, and R. Naveenkumar, "Privacy-Aware Edge-Based Intelligent Video Analytics for Scalable Crowd Management in Smart Cities," 2026 3rd International Conference on Integrated Intelligence and Communication Systems (ICIICS), pp. 1–7, June 2026, doi: 10.1109/iciics67880.2026.11483444

[26] J. Mökander, J. Schuett, H. R. Kirk, and L. Floridi, "Auditing large language models: a three-layered approach," May 2023, doi: 10.1007/s43681-023-00289-2

[27] G. Padma, V. A. R. Reddy, S. Devi. K. A, B. Buvaneswari, and K. S. S. Kumar, "Privacy-Preserved Face Recognition Biometric Authentication Using FaceNet and Zero-Knowledge Proofs for Secure, Access Control on Decentralized Blockchain Networks," 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN), pp. 1–8, Apr. 2026, doi: 10.1109/iciscn67954.2026.11566528.

[28] L. Wang et al., "A Survey on Large Language Model-based Autonomous Agents," Sept. 2023. doi: 10.48550/arXiv.2308.11432.

[29] M. Recharla, R. S. Garapati, V. D. V. K. Bandi, U. S. Yandamuri, and B. M. Mangalampalli, "Generative AI-Enhanced Data Engineering Pipelines for Predictive Biomarker Discovery in Alzheimer's and Kidney Disease," 2026 IEEE International Conference on AI Engineering and Innovations (AIEI), pp. 1–5, Mar. 2026, doi: 10.1109/aiei69164.2026.11496858.

[30] S. Raschka, Y. Liu, V. Mirjalili, and D. Dzhulgakov, "Large language models and generative AI for data engineering and analytics," Information Systems, vol. 125, 2024.

[31] B. D. Bhavani, Sowmya. S. R, R. Loganathan, S. Nagaraj, and Vasukidevi. G, "Evolutionary Gravitational Neocognitron Neural Network, Snow Leopard Optimization and Deep Graph Reinforcement Learning for Routing Protocol in WSN," 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN), pp. 1–8, Apr. 2026, doi: 10.1109/iciscn67954.2026.11566191.

[32] N. B. Hiremath, S. K. Kolla, R. Sunkara, S. Lal. G, and K. Sireesha, "Adaptive and Intelligent Secure Multimedia Transmission for Dynamic Communication Systems," 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN), pp. 1–8, Apr. 2026, doi: 10.1109/iciscn67954.2026.11566143.

[33] K. Naveen, R. Lingam, G. R. Kunduru, N. Mangala, N. N. Reddy, and P. Balaji, "Intelligent Loan Approval System: Machine Learning for Home and Education Loan Eligibility," 2026 Contemporary Computing Innovations Conference (CCIC), pp. 1–6, Feb. 2026, doi: 10.1109/ccic68129.2026.11486013.

[34] T. Chen, H. Xu, and Y. Zhang, "AIOps-driven anomaly detection and root cause analysis for cloud-native enterprise platforms," Future Generation Computer Systems, vol. 158, pp. 310–324, 2024.

[35] R. Loganathan, K. Amistapuram, and A. R. Aitha, "Letter to the Editor re Annual Updates of the European Association of Urology – European Society for Pediatric Urology (EAU-ESPU) Pediatric Urology Guidelines: Are Large-Language Models (LLMs) Better Than the Usual Structured Methodology?," Journal of Pediatric Urology, p. 106057, June 2026, doi: 10.1016/j.jpurol.2026.106057.

[36] K. C. Nagabhyru, S. Singireddy, A. L. Gadi, G. K. Sheelam, and D. Kapila, "Toward Secure and Usable Communication-Centric Authorization Models for Smart Homes," Lecture Notes in Electrical Engineering, pp. 355–366, 2026, doi: 10.1007/978-3-032-20235-2_32.

[37] R. Paramasivam, S. S. Reddy, V. A. R. Reddy, K. Sandra, and M. V. S. Narayana, "JellyFish Search Optimization with Arctic Puffin Optimization Algorithm for Efficient Routing Based on the Software Defined Communication Network," 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN), pp. 1–6, Apr. 2026, doi: 10.1109/iciscn67954.2026.11566144.

[38] K. Jyoshna, R. K. Peddi, P. Nayak. S. Dhanamalar, M, and S. Punitha, "Spatio-Temporal Graph Neural Network with Global Spatio-Temporal Network for the Traffic Flow Prediction," 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN), pp. 1–6, Apr. 2026, doi: 10.1109/iciscn67954.2026.11566462.

[39] M. Krishnan et al., "Engineering Intelligent Cloud-Native Data Ecosystems for Predictive Decision-Making in Industry," Journal of European Economic History, vol. 7, no. 2, pp. 68–88, June 2026, doi: 10.61336/jeeh/26-2-7.

[40] S. Amershi et al., "Software Engineering for Machine Learning: A Case Study," 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), May 2019, doi: 10.1109/icse-seip.2019.00042.

[41] D. Sculley, et al., "Hidden technical debt in machine learning systems," Advances in Neural Information Processing Systems, vol. 28, pp. 2503–2511, 2015.

[42] R. Inala, "Cloud-Native AI and MDM Framework for Next-Generation Insurance and Retirement Data Products," International Journal of Engineering & Extended Technologies Research, vol. 8, no. 03, May 2026, doi: 10.15662/ijeetr.2026.0803006.

[43] S. Manikandan, G. P. Singh, V. R. R. Gottimukkala, P. S. L. N. Davuluri, R. Sadagopan, and T. V. Kumar, "Human-in-the-Loop Automation Patterns for Financial Services Using Camunda + Modern Java UIs," 2026 IEEE International Conference for Convergence in Computing Technology (I3CTCON), pp. 1–6, Mar. 2026, doi: 10.1109/i3ctcon68242.2026.11507795.

[44] S. H. Kolla and R. Mattaparthi, "Hybrid Gen AI Systems: Integrating Small LMs with Large Language Models for Cost-Efficient Enterprise Automation and Decision Intelligence," International Journal of Research Publications in Engineering, Technology and Management, vol. 08, no. 06, Dec. 2025, doi: 10.15662/ijrpetm.2025.0806038.

[45] A. Gopinath, P. S. L. N. Davuluri, U. B. Kumar, Sahana. M P, and G. Yamini, "Communication Aware Malware Detection Using Network Flow Bytecode Fusion With Adaptive Focal Optimization," 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN), pp. 1–8, Apr. 2026, doi: 10.1109/iciscn67954.2026.11566323.

[46] K. Murphy, J. Allen, and S. Taylor, "Deep autoencoder-based fraud representation learning for insurance intelligence systems," Neural Computing and Applications, vol. 37, no. 8, pp. 9145–9162, 2025.

[47] T. Kalita, S. Prasad Tiwari, A. Reddy Aitha, and A. Garg, "Evolutionary and Swarm-Based Metaheuristics for Neural Architecture Search and Hyperparameter Tuning," SSRN Electronic Journal, 2026, doi: 10.2139/ssrn.6708638.

[48] Supriya. R K, N. Mangala, R. Priyanka, R. A. Mohammed Sait, and P. P. Selvam, "Privacy Preserving Analytics for Intelligent in Internet of Things System in Health Care Using Graph Neural Network with Latent Graph Inference Technique," 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN), pp. 1–6, Apr. 2026, doi: 10.1109/iciscn67954.2026.11566407.

[49] P. Nayak S, R. Loganathan, Velmurugan. R, S. R. K. Sarma A, and V. Jayaraj, "Scale-Aware Dilated Lightweight Convolutional Network Improving Solar Panel Defect Classification through Efficient Electroluminescent Image Analysis Techniques," 2026 2nd International Conference on Intelligent Systems and Computational Networks (ICISCN), pp. 1–7, Apr. 2026, doi: 10.1109/iciscn67954.2026.11566218.

[50] M. S. R. L. Reddy, T. Sunitha, K. Kanchana, A. R. Nagubandi, A. R. Segireddy, and S. N. Bhavanam, "AI-Enhanced Blockchain Consensus Mechanisms for Secure Transaction Validation," 2026 International Conference on Emerging Research in Smart Electronics and Machine Informatics (ECMI), pp. 1–11, Apr. 2026, doi: 10.1109/ecmi68341.2026.11602724.

[51] SUNKARA, S. K. (2025). LEVERAGING AI, IoT, AND BLOCKCHAIN FOR SCALABLE DIGITAL TRANSFORMATION IN POST-HARVEST SUPPLY CHAINS: A MULTI-SECTOR APPROACH TO ENHANCING EFFICIENCY AND TRACEABILITY (Vol. 26, Issue 7, pp. 2757–2766).