International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|

DOI: 10.15662/IJSRAT.2024.0706007

AI-Enabled Data Engineering Pipelines for Smart Grid Fault Detection

Nareddy Abhireddy

Independent Researcher, India

nareddy.abhireddy.researcher@gmail.com

ABSTRACT: Research on smart grids has drawn attention due to the increasing reliance on multiple sources of generation, such as solar and wind farms, and the decentralization of decision-making due to the increase in distributed generators. The implementation of new technologies such as machine learning (ML), available within the framework of artificial intelligence (AI), promises to increase reliability and efficiency by anticipating events and acting before they occur. AI-enabled data engineering pipelines integrate data-processing procedures—such as quality assurance, governance, ingestion, transformation, and preparation—with AI modeling techniques addressing applications such as fault detection and predictive maintenance. These pipelines embed advanced data-processing techniques and tools into a reliable management framework. Smart grids monitor power distribution, detect faults, and feed data, including that from phasor measurement units (PMUs), supervisory control and data acquisition (SCADA), and weather-related sensors.

Data flowing through the pipelines is analyzed in real-time using a fixed library of ML fault-detection models. Fixed models, configured in a lawyer-and-create manner, allow some level of exploration during the training phase of the pipeline. Delays in the deployment of fixed models represent a serious concern for many businesses; the situation is more critical for companies with a mature AI practice—the latency is built into the model and processes association—and is nevertheless significant for business pipelines. Stream processing pipelines execute each processing step as soon as new data arrive for that step. However, model evaluation requires more than a supervised test data set; measures are needed to control the false-positive rate in a business pipeline and assess model robustness against rare events such as blackouts or smaller events such as terminal failures.

KEYWORDS: Smart grid · Fault detection · Data engineering · Artificial intelligence · Supervised learning · Unsupervised learning · Semi-supervised learning,AI Use academic tone with objective, evidence-based arguments, and formal structure; present clarity, rigor, and precise terminology throughout the manuscript.

I. INTRODUCTION

Electric utility grids are undergoing a significant transition from traditional architectures toward smarter and more decentralized configurations. Fault detection is one of many essential operations in smart grids that remains an active area of research. It is also one of the most important operations, since any power outages or service interruptions cause discontent among electric consumers and lay significant economic burdens on both the utility organization and the entire economy. Artificial intelligence (AI) methods are making inroads into fault-detection operations, but the task is nontrivial because of the complexity of grids and the variety of failure types. An integral obstacle, however, is that data pipelines for fault detection are seldom designed and engineered from end to end. Data engineering for AI is, in general, underappreciated in many Smart-X domains, where an overwhelming focus is placed on the AI application rather than on the data-source-to-algorithm requirements and resource needs.

Failure to construct pipelines tailored to the input data, output timing, and application of AI leads to data-availability delays and worse potential availability, quality, or freshness of the information inputs needed to train, validate, optimize, and run the algorithms. Addressing such deficiencies usually entails concurrent use of different data-engineering methods built for other Smart-X areas. Such a piecemeal approach adds excessive overhead and latency to preparing data for any consequent modelling. For fault detection specific to electric power grids, AI models based on historical data are increasingly complemented (but not supplanted) by real-time models that persistently and continuously process streams of information.

Failure to design data pipelines that are tailored to the characteristics of input data, required output timing, and the specific needs of AI applications can result in delays in data availability and negatively affect the availability, quality,

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|
DOI: 10.15662/IJSRAT.2024.0706007

and freshness of the information required for training, validating, optimizing, and deploying algorithms. When these deficiencies occur, organizations often rely on multiple data-engineering techniques originally developed for other Smart-X domains, such as smart cities or smart manufacturing. However, this piecemeal integration introduces additional overhead and latency in preparing data for modeling tasks, making the overall process inefficient. In the context of electric power grids, particularly for fault detection, AI models that traditionally rely on historical datasets are increasingly being complemented by real-time models capable of continuously processing live streams of operational data. These streaming-based approaches enhance situational awareness and enable faster detection and response to anomalies, thereby improving the reliability and resilience of modern power systems.

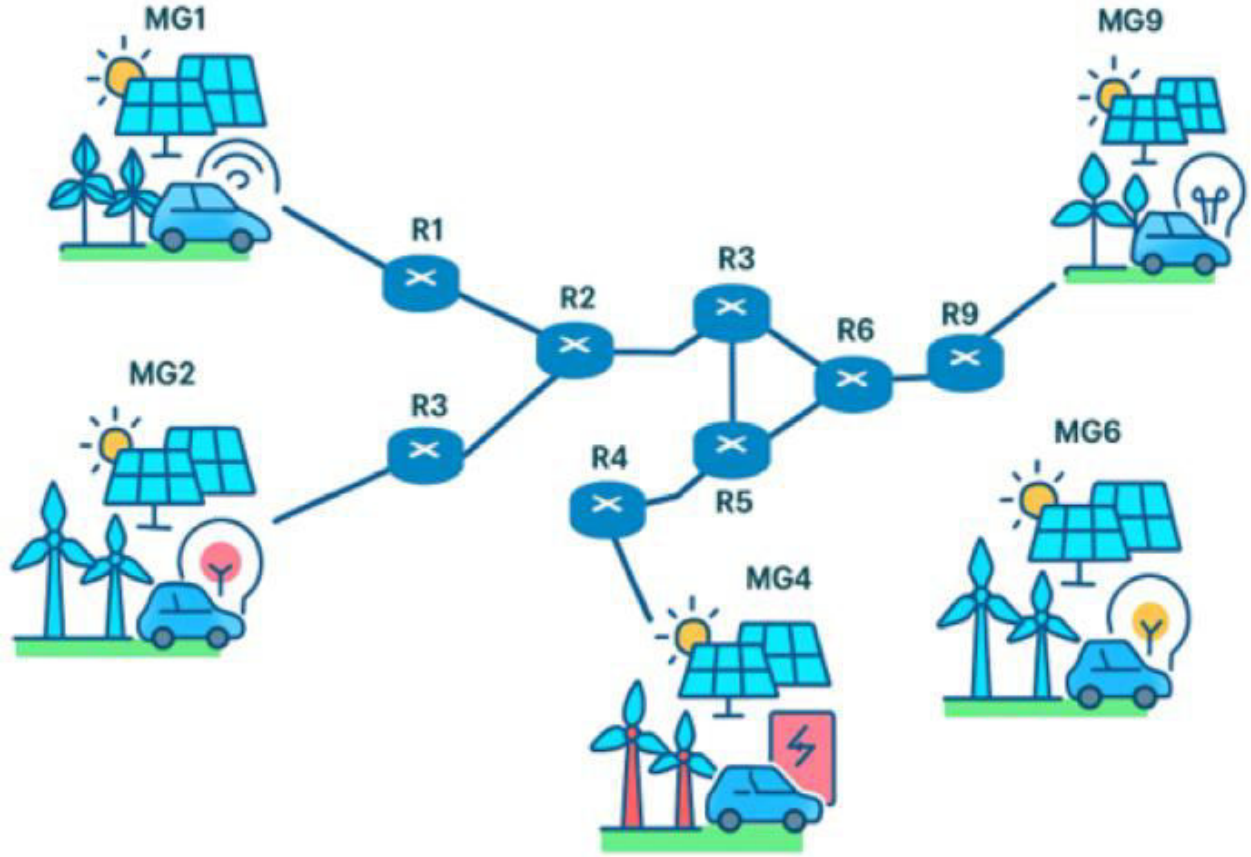


Fig 1: Enhancing Smart Grid Security and Efficiency

1.1. Background and Significance

The smart grid combines electricity supply networks with an integrated ICT layer for end-to-end communication, computation, and control. The flow of information enables data mining-based solutions for a wide range of engineering problems. Many AI-based methods offer innovative solutions for fault detection, but none address the full integration of end-to-end data engineering pipelines devoted to the task. Comprehensive solutions are crucial for real-world deployment, where problems such as data quality and quality assurance cannot be overlooked. Addressing these aspects makes the analysis of the AI modeling technique a secondary matter, and the methods used are not novel.

A strong research trend focuses on the design and implementation of AI-based solutions for specific problems using available sensor data. The proposed work aims to integrate these contributions into end-to-end data engineering pipelines that continuously support fault detection in smart-grid infrastructures using live data. The outcome is expected to improve the reliability and operating cost of the assets by increasing fault-detection precision and coverage. Such systems help avoid catastrophic failures with enormous social costs. The analysis focuses on data-engineering aspects, laying the groundwork for subsequent modelling developments while establishing a dedicated validation framework based on sound principles of evidence-based computing.

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|

DOI: 10.15662/IJSRAT.2024.0706007

Equation 1: Confusion matrix and event counts

Let the ground truth label be:

- $y = 1$: fault
- $y = 0$: healthy

Let the model prediction be:

- $\hat{y} = 1$: predicted fault
- $\hat{y} = 0$: predicted healthy

Define the 4 fundamental counts:

1. **True Positives (TP)**

Fault happened and model predicted fault:

$$TP = \#\{i: y_i = 1 \wedge \hat{y}_i = 1\}$$

2. **False Positives (FP)**

No fault happened but model predicted fault:

$$FP = \#\{i: y_i = 0 \wedge \hat{y}_i = 1\}$$

3. **True Negatives (TN)**

No fault happened and model predicted healthy:

$$TN = \#\{i: y_i = 0 \wedge \hat{y}_i = 0\}$$

4. **False Negatives (FN)**

Fault happened but model predicted healthy:

$$FN = \#\{i: y_i = 1 \wedge \hat{y}_i = 0\}$$

And the total sample size:

$$N = TP + FP + TN + FN$$

II. BACKGROUND AND MOTIVATION

The proposed research seeks to fill this gap by creating Smart Grid-specific Data Engineering Pipelines that provide data preparation and orchestration capabilities tailored to the requirements of AI-powered fault detection and complementary functions. Supervised classification, unsupervised anomaly detection, and self-supervised prediction of available observations for semi-supervised learning are explored in detail. Together with feature engineering, these techniques represent essential components of any fault detection system. The proposed Data Engineering Pipelines integrate all necessary sensors, data sources, and PMUs for streaming or batch fault detection, encompassing Quality, Privacy, Security, and Governance considerations. Careful analysis of the associated end-to-end Big Data and AI Engineering Pipeline combines model training, validation, and Continuous Integration /Continuous Deployment concepts into a DL/CI framework tailored to Smart Grids and beyond.

The primary focus is on specific architectural components and associated Data Architecture aspects of the Data Engineering Pipelines for the family of Power Transmission Grids . These major Capital Assets of any Country are subject to gradually increasing demands of Traffic and Load. The growing complexity and size of the Network and the impossibility of real-time analysis of all information augment the necessity of introducing effective monitoring solutions based on automatic utilities. AI, Data Engineering Pipelines and the increasing penetration of Digital Twins represent the necessary foundation for such evolution. Continuous improvement of the Power transmission Network Reliability and Efficiency is a Sustainable Development Goal of its own in the European Union and World. Modern **power transmission grids**, as critical national capital assets, face steadily rising demands in traffic, load, and operational complexity. As these networks expand, real-time analysis of the vast volumes of operational data becomes increasingly difficult, highlighting the need for advanced monitoring solutions supported by automated systems. **Data engineering pipelines**, combined with **AI-driven analytics** and **digital twin technologies**, form the core architectural foundation for managing, processing, and interpreting grid data efficiently. These technologies enable predictive maintenance, improved reliability, and optimized network performance. Strengthening the **reliability and efficiency of power transmission networks** is therefore a key objective aligned with sustainable development priorities in the **European Union and globally**, ensuring resilient and future-ready energy infrastructure.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|
DOI: 10.15662/IJSRAT.2024.0706007

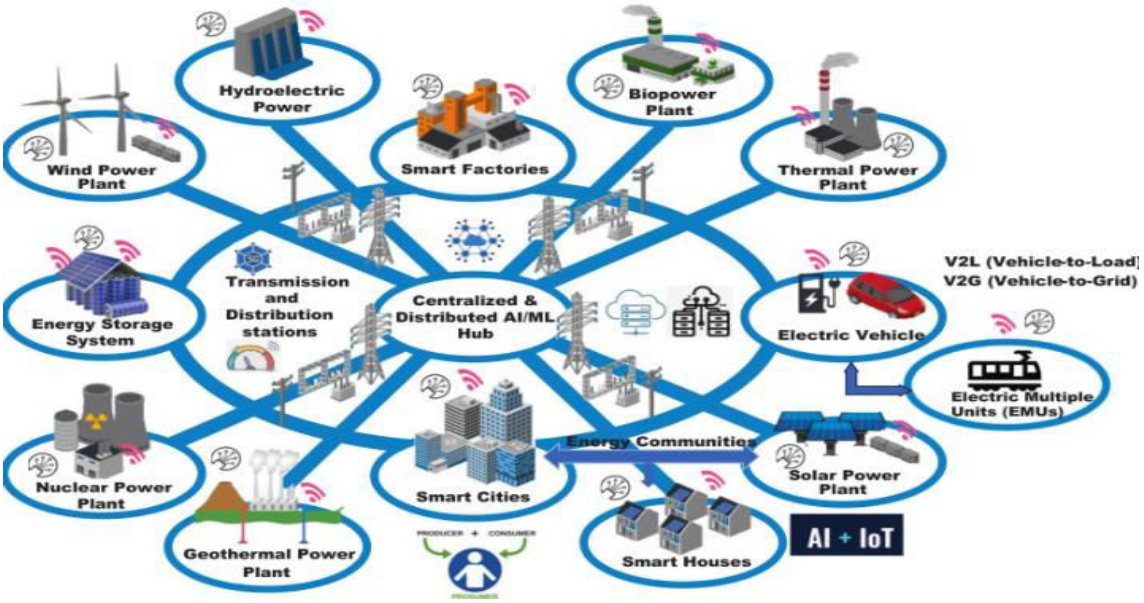


Fig 2: Background and Motivation of AI-Enabled Data Engineering

2.1. Research design

The research hypothesis states that fault conditions in the smart grid environment can be detected using AI techniques during the operational phase of the grid, as indicated by sensor inputs. To validate this hypothesis, several AI models are built with a variety of algorithms (from supervised learning, unsupervised learning, self-supervised learning, and semi-supervised learning). The supervised learning models utilize both historical fault and non-fault data for training, while the remaining models rely on fault-free historical data without any prior labeling of fault conditions. The training of the supervised learning models is performed in a single data ingestion step with cross-validation, and training for the remaining models is combined with subsequent forecasting data when new data arrives (similar to transfer learning)

The input data for these models comprises SCADA, PMU, and condition-monitoring (CM) data. When available, PMU and SCADA data are used together with CM data to build the models, as these sensor types provide different levels of information. In the case of PMU data, where latency constraints associated with communications and computation are critical, a model is built with only CM data and PMU data are supplied subsequently for verification.

Equation 2: Precision (step-by-step)

AI-Enabled Data Engineering Pip…

.

Start from the definition:

- "predicted faults" are all cases where $\hat{y} = 1$, which count is $TP + FP$
- "truly faults among those predicted faults" are TP

So:

$$\text{Precision} = \frac{\text{true predicted faults}}{\text{all predicted faults}} = \frac{TP}{TP + FP}$$

III. DATA ARCHITECTURE FOR SMART GRIDS

Architectural components, the anticipated flow of information through the system, and specific requirements for data collection, storage, and processing are described here for fault-detection pipelines.

The design of fault-detection architectures is based on data sourced from fault-detected or failed remote-control interlocking stations. Thus, four sets of data matter: smart-sensor data (vibration and temperature), Phasor Measurement Unit (PMU) data, SCADA data, and ancillary data from the Indian Meteorological Department and the Indian Space Research Organisation. Data from these sources can be ingested through an online streaming process, making them suitable for decision-making in near real time. All smart-sensor data will be marked with a latency value

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|
DOI: 10.15662/IJSRAT.2024.0706007

and are expected to arrive at the server every second, requiring internal storage of no more than 1,296 records or 21 minutes. User- or application-programming-interface-based SCADA and PMU data are expected to be ingested at least every minute, with a maximum latency of five minutes. The data-dimensionality-explosion problems resulting from the two-dimensional nature of PMU data will be avoided by maintaining 10-sample compression at the ingestion level.

Since ML models presently do not directly accept data from sensory units, a two-pronged approach will be employed. First, only SCADA and PMU data will be ingested, with prediction models triggering the required sensory-predictor models in the background. Second, a batch-pipeline process at its own frequency will dump the required dimensions from the sensory units into a separate space for offline ML model training, performance benchmarking, and failure-action-recommendation model preparation. Online ingestion of ancillary data within a specified lag period will be required for quality assurance. A general-data-quality and governance framework based on data lineage, data-freshness checks, data-privacy requirements, and data-security policies has been defined. Specific checks will be implemented along with system development and deployment.

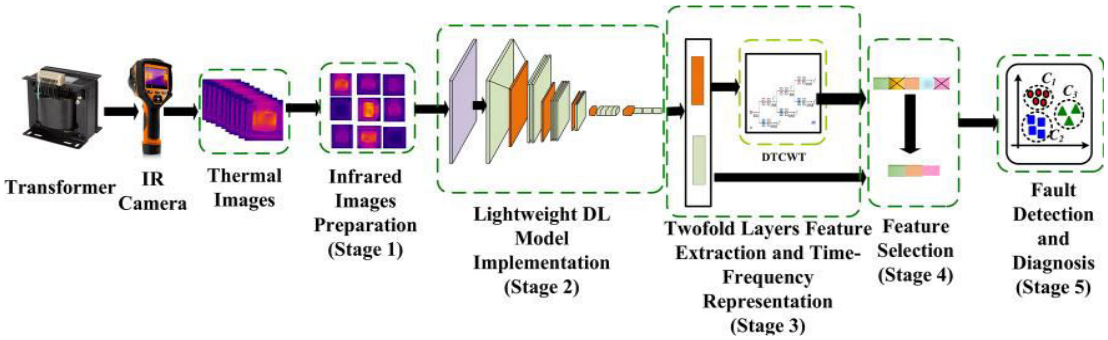


Fig 3: A lightweight deep learning framework for transformer fault diagnosis in smart grids

3.1. Data Sources and Ingestion

Data sources encompass diverse sensor modalities. Electric fault detection demands phasor measurement unit (PMU) data for precise temporal alignment with SCADA measurements. In case of Power and Fast Transfer Oscillation Detection systems the installation of PMUs is mandatory in the vicinity of these systems. Other faults can be detected with simple SCADA data. However, the ideal solution for electrical fault detection utilises PMU measurements from the full grid together with information from the Fast Transfer Oscillation Detection and Power Transfer Oscillation Detection systems. These need to be acquired with a latency of seconds and normalised against one or more values associated with normal operating conditions. Environmental data affect the likelihood of equipment failure but do not necessarily lead to equipment fault indication. As such, they may be considered secondary sources when using SCADA data for fault detection. For these reasons environmental and meteorological data may be sourced from identification of available open-access data sources, retrieved at the maximum temporal resolution available and for the same time period as the electric fault SCADA data.

Data latency and likelihood of encountering missing values also drive other selections, particularly on overhead and buried optical fiber temperatures which are obtained from installations available in Open Data format. Missing values for secondary available data are less critical than for PMU measurements. These are thus filled with methods available in Dask, the library chosen for the ingestion pipelines. Stochastic Transfer function models have been successfully used to fill complete fibres' temperature series with significant improvement of the predictive performance considering that they get close predictive models for neighbouring and prospective systems.

Equation 3: Recall (step-by-step)

AI-Enabled Data Engineering Pip…

.

- "original faults" are all cases where $y = 1$, which count is $TP + FN$
- "faults we correctly detected" are TP

So:

$$\text{Recall} = \frac{\text{correctly detected faults}}{\text{all true faults}} = \frac{TP}{TP + FN}$$

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|

DOI: 10.15662/IJSRAT.2024.0706007

3.2. Data Quality and Governance

Quality assessment relies on the classic dimensions of data quality: accuracy (are values correct?), consistency (are values conflicting?), completeness (is it complete?), conformity (does it follow the schema?), and timeliness (is it up to date?). Based on these attributes, policies for data quality or data governance can be established. For instance, governance may define that data are complete at a given level of completeness or that absence of events is a very rare case. When such conditions are violated, alarms may trigger the evaluation of root causes, and possible corrections may be recommended.

Data governance also covers data lineage, addressing questions such as where data originate, how they are transformed throughout their cycle, and how they are key in an organization. Security and privacy policies also belong within data governance efforts. Sources that deal with personal data are not allowed to expose these identities. In this case, algorithms such as k-anonymity can be applied, and the data will remain viable for many situations. When it comes to sharing data—services or raw data—they must also express their policies for use, which can be defined using specific policies on personal data, such as European General Data Protection Regulation compliance or even defined custom ones.

IV. AI METHODS FOR FAULT DETECTION

A wide range of AI methods and algorithms can be explored to model smart-grid faults. The basic task is to train models that predict faults given the available historical data; thus, the performance is evaluated based on how well the model can predict the future. Artificial neural networks (ANNs) and ensemble techniques (random forests, boosted trees) are the choices for supervised learning because they can capture complex relationships and nonlinearities between features. While other algorithms and ensembles can be considered, these are expected to perform well. The model-training pipeline is represented in. Since labels are crucial for supervised learning, it is important to detail the labeling strategy. The label within each batch is the state of the grid as indicated by binary fault logging (0 – healthy, 1 – fault) for the longest possible time interval, prioritizing ground-truth labels within the window. A separate strategy for preparing historical records from patched OLTC data enables gap-filling, fault-labeling, and feature-selection analyses, relegating supervised techniques to auxiliary support for guiding and validating self-supervised approaches. With each batch associated with its own description, fine-tuning, k-fold validation, and other processes (permutation feature importance, hyper-parameter tuning) become straightforward. Two types of holdout validation provide additional assurance: a strict block-out strategy and a second design that retains part of the most recent history for final validation following thorough earlier scrutiny.

The other principal approaches are unsupervised and semi-supervised methods. During exploration of the data, anomalous windows are highlighted by visualization and may be well-suited for clustering, enabling groups of similar anomalies to be discovered. Anomaly detection exploits the absence of faults to identify atypical behavior through reconstruction-error metrics. Self-supervised workflows leverage multi-view representations learned from recent years without known faults, focusing on efficient encoding. Auxiliary information such as difference-frequency and derivative-time-series representations aids understanding and can further contribute within a semi-supervised learning framework that effectively capitalizes on affirmative labels from known system models.

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|
DOI: 10.15662/IJSRAT.2024.0706007

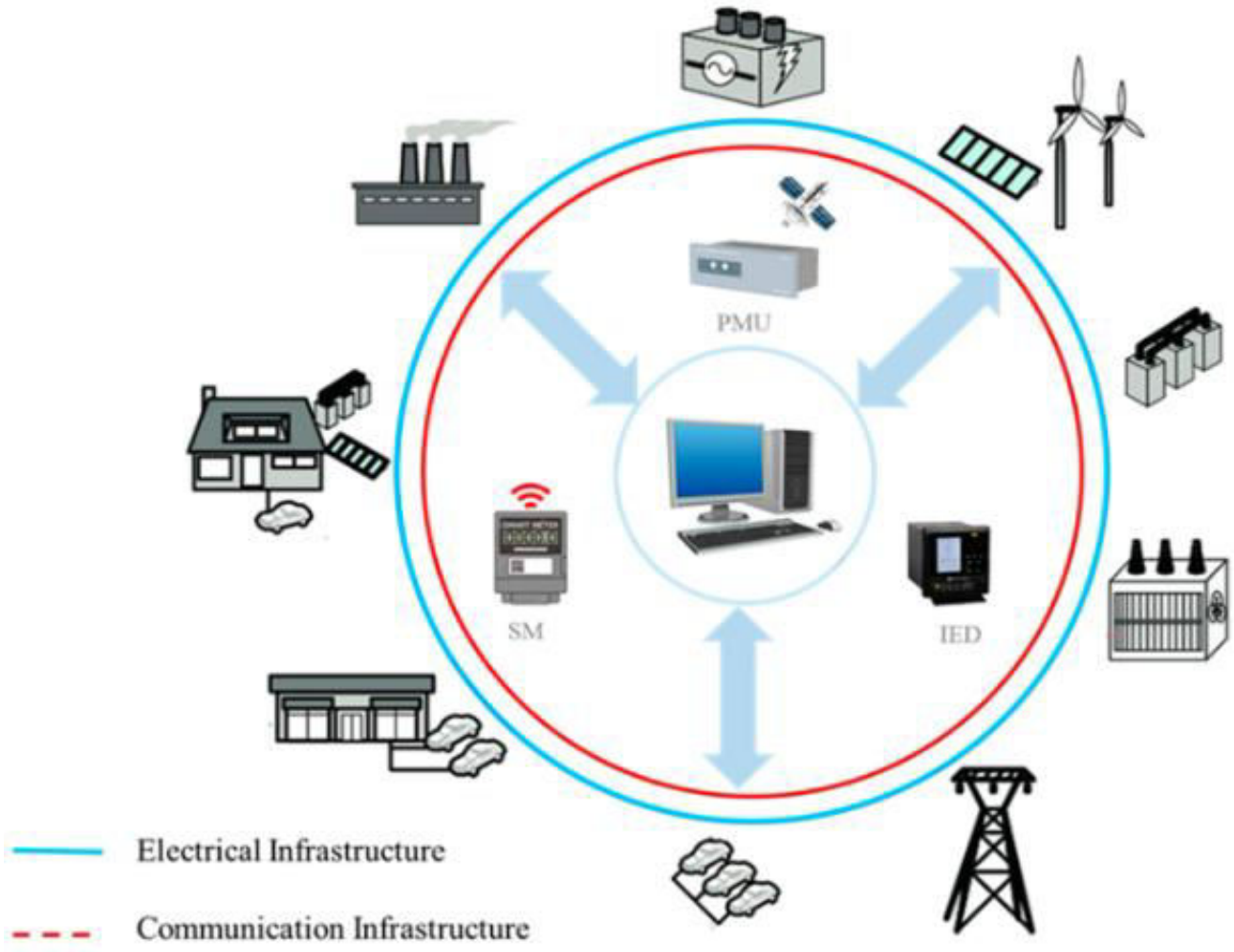


Fig 4: Artificial Intelligence-Based Fault Location Methods in Power Distribution Networks

4.1. Supervised Learning Approaches

Random Forest, Gradient Boosting, and XGBoost models will be created for each fault category. Based on previous analyses of grid currents and voltages, these types of faults correspond to variations in fundamental frequency voltage and current phasors, thus providing an appropriate environment for supervised learning models that require signal label information. The raw data will therefore have to be properly labelled based on the aforementioned approach. Feature engineering will take into account the signal characteristics of the magnitude in the time and frequency domain and the phase in the time domain. Furthermore, in order to ensure the models' good generalisation performance, stratified 10-fold cross-validation will be performed, followed by hyper-parameter optimisation.

Supervised learning requirements are labels and feature engineering. A separate data engineering task will thus be to label the data based on their frequencies until certain thresholds of probability have been reached. These thresholds will be determined after analysing the reliability of the modelling and an in-depth exploration of the data. Once successfully labelled, the MiniPID-L-SCADA data will follow Preisa's methodology, using the different classes as strata. Preisa's feature set will then be applied for model training in the context of clustering. In some cases a Radar plot is better than PCA to explain certain registers.

Equation 4: F1 score (step-by-step)

Start with harmonic mean of two numbers $a$and b:

$$H(a,b) = \frac{2ab}{a+b}$$

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|
DOI: 10.15662/IJSRAT.2024.0706007

Set a = Precision, b = Recall:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Substitute precision and recall using TP/FP/FN if you want a single expression:

$$F1 = \frac{2 \cdot \frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}{\frac{TP}{TP+FP} + \frac{TP}{TP+FN}}$$

Multiply numerator and denominator carefully:
Numerator:

$$2 \cdot \frac{TP^2}{(TP+FP)(TP+FN)}$$

Denominator:

$$\frac{TP(TP+FN)+TP(TP+FP)}{(TP+FP)(TP+FN)} = \frac{TP(2TP+FP+FN)}{(TP+FP)(TP+FN)}$$

Divide numerator by denominator (the $(TP+FP)(TP+FN)$cancels):

$$F1 = \frac{2TP^2}{TP(2TP+FP+FN)} = \frac{2TP}{2TP+FP+FN}$$

So a useful "confusion-matrix form" is:

$$\boxed{F1 = \frac{2TP}{2TP+FP+FN}}$$

4.2. Unsupervised and Semi-Supervised Techniques

Anomaly detection, clustering, self-supervised methods, and semi-supervised learning workflows round out the strategy team tasked with developing the smart grid fault-detection pipeline. Unsupervised approaches play a crucial role, particularly in settings with limited or no labelled fault events, by allowing a more general understanding of normality through normal instance learning and robust clustering. Additionally, unlabeled incoming data can be leveraged via self-supervised strategies. The features extracted through these unsupervised methods serve an important role as auxiliary inputs fed into the various supervised algorithms. The labelled dataset can be complemented through synthetic anomaly generation.

Anomaly detection methods leverage a training set comprising only normal observations, with detection based on the distance of an unseen data point from the learned normality. Clustering approaches group similar instances into a finite number of clusters whose centroids define local normality. For both of these methods, detection can be defined as a binary classification task using the distance from the learned model as a feature. Self-supervision on the incoming data during deployment allows knowledge to be transferred to a semi-supervised model that combines a small number of labelled observations with the larger amount of unlabeled data. Semi-supervised learning can also leverage a supervised classifier fitted on the normal observations to generate pseudo-labels for the incoming data.

V. PIPELINE ENGINEERING AND DEPLOYMENT

Enabling end-to-end pipelines for smart-grid fault detection requires a series of design choices and considerations associated with operational deployment. Streaming pipelines process data in real time, while batch pipelines operate at different frequencies, necessitating consistent cross-domain data freshness. When real-time performance is not essential, the outputs of the batch pipelines can be consumed by the test or secondary-phase training pipelines. Supervised models and their DL implementations are trained periodically, validated before deployment, and continuously monitored for performance degradation. No drift detection is required for self-supervised and unsupervised methods.

Smart_grid_dg.png As shown in the architecture diagram, the different stages and their interactions are orchestrated with Apache NiFi. The Apache Kafka ecosystem, together with Apache Spark, MLflow, and Docker, is used to deliver

International Journal of Science, Research and Technology (IJSRAT)
|Bimonthly | Peer-Reviewed | Scholarly Journal|
|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|
DOI: 10.15662/IJSRAT.2024.0706007

the assets. The end-to-end pipelines are deployed on a completely separated hyperscaler environment and marketed as a service to the participating universities and industries. The goal is to leverage the common data sources, shared codebases, and know-how to reduce costs and increase the frequency of algorithm updates, thus avoiding a tedious and time-consuming phase every time the pipeline needs to be refreshed.

Equation 5: Deriving the "21 minutes" buffer from 1,296 records
Given:

- sample period $T_s = 1$second/record
- max buffered records $N_b = 1296$

Buffered time span:

$$T_b = N_b \cdot T_s = 1296 \cdot 1 \text{ s} = 1296 \text{ s}$$

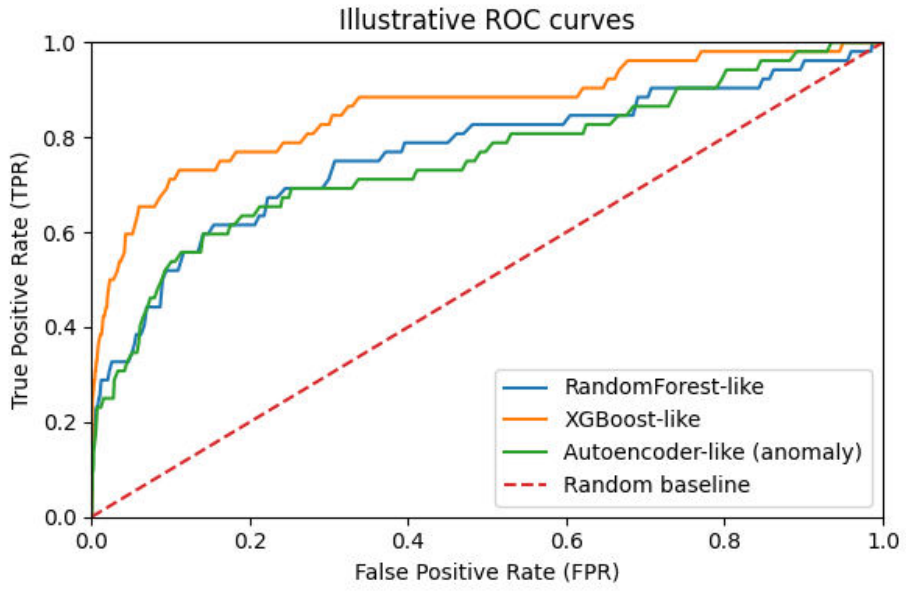
Convert seconds to minutes:

$$1296 \text{ s} \cdot \frac{1 \text{ min}}{60 \text{ s}} = 21.6 \text{ min}$$

5.1. Streaming vs Batch Processing

A key decision point in data pipeline engineering is the choice between a streaming architecture and a batch-processing approach. Smart grids generate high-volume data streams, but AI-enabled pipelines can tolerate some latency and, for some modes, freshness of the data is not a factor. Streaming architectures ingest data in real time and, for fault detection, require data to be delivered with minimal delay, typically a sub-second latency, for automated query and response. However, if a real-time response is not demanded, a batch-processing architecture constitutes a simpler alternative, with the trade-off that—until completion of a batch—the results may be delayed by several seconds. Most deployments will favor a micro-batch approach, where data loads are ingested at regular intervals. Internal sensors send continuous streams of telemetry data, PMU supplies a high-speed signal for stable regions of the grid, and ancillary systems support some level of processing for these self-contained datasets. These features allow failure detection, according to fan-in and fan-out rules, to be performed close to real time but for a latency of perhaps several seconds.

Compared to fault detection in self-contained subsystems, pipelines that monitor for faults in a cascading manner are generally delayed by several seconds, often sub-minute latencies. However, even for SCADA data—the slowest component of the larger fault-detection pipeline—this remains an acceptable range of latency for most operational applications. Latency, therefore, is not a primary concern for the SCADA data-load, nor for the pipelines that rely on the smart-meter data-load from the ancillary-class system, where arrival at the processing point occurs many seconds after any reported incident. For the majority of these auxiliary datasets, fresh real-time data is not strictly required; information posted after the fact is sufficient for validation of supervisory monitoring and internal authorization.



International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|

DOI: 10.15662/IJSRAT.2024.0706007

5.2. Model Training, Validation, and DL/CI

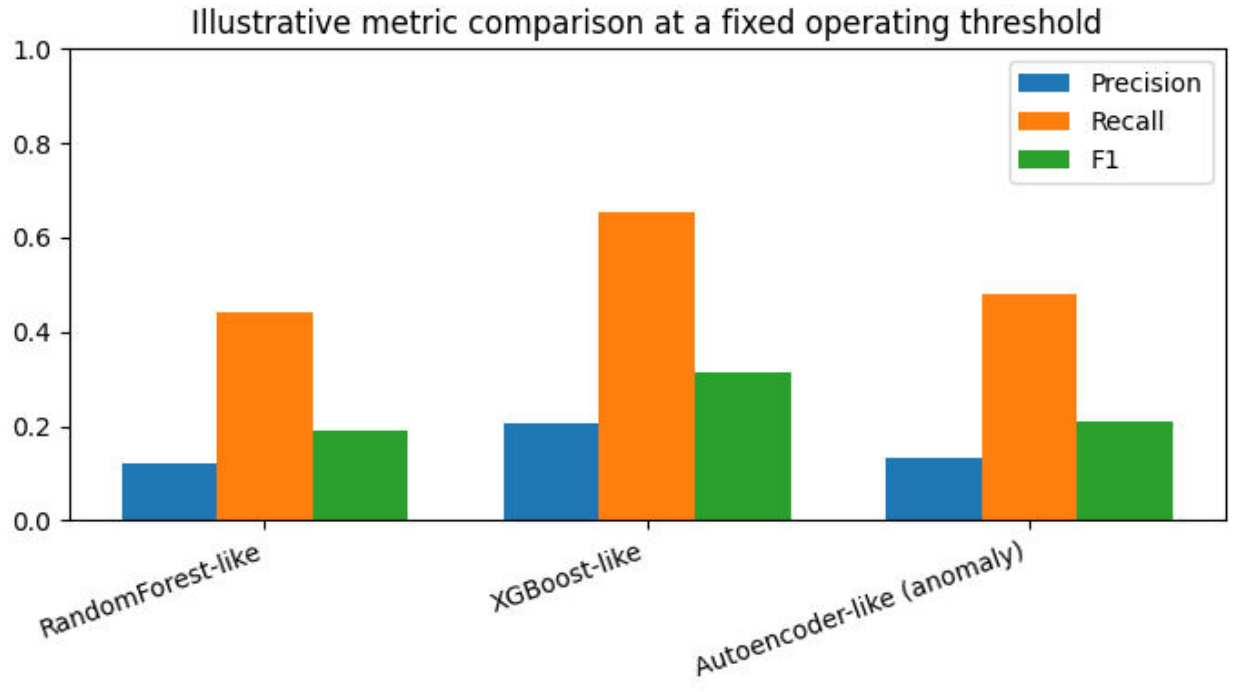
The training pipeline for supervised models incorporates an approval layer for cross-validation. Baseline supervised models provide the foundation for employing more complex and resource-intensive algorithms. Supervised learning generally excels when the label set is sufficiently large, balanced, and accurately reflects the targeted behavior. Regular monitoring of labeling drift, population drift, and feature drift is crucial. Continuous Integration (CI) pipelines enable sophisticated engineering of Continuous Delivery (CD) workflows that deploy the cross-validating models. A heterogeneous model community is established and nourished through these channels.

Validation of unsupervised and semi-supervised approaches should be done using clear metric definitions and real-world, production-like grid scenarios. The principle of using production-like conditions for accuracy also applies to monitoring anomaly and detection, clustering, and semi-supervised methods to prevent collapse in unbalanced situations. Self-supervised models are held separately. As discussed previously, the main goal is to deliver an operational model with other auxiliary models covering the unbalanced populations.

VI. EVALUATION AND VALIDATION

To assess whether the end-to-end data engineering pipelines have sufficient quality for practical deployment within a smart grid context, multiple validation strategies are applied and formal validation metrics specifically designed for fault-detection problems are defined. Several recent studies employed AI methods for smart grid fault detection, but almost none of these works considered the complete data life cycle. Reviews of AI methods for smart grid applications pointed out this shortcoming. The absence of working data engineering pipelines adds uncertainties about the reliability of practical system deployment. Addressing these gaps is critical, especially for data-driven applications with limited domain knowledge. Machines learning approaches learn directly from data, making them especially sensitive to data quality and resource availability. Pipelines addressing these aspects need to be developed and incorporated into real-time systems.

A formal set of empirical metrics specifically tailored to quantify critical aspects of the fault-detection functionality evolves from similar definitions in the field of information retrieval. For fault detection, the most important indicators are precision – the fraction of predicted faults that truly represent a fault in the original data set; recall – the fraction of detected original faults that are predicted by the AI method; and the corresponding F1 score, balancing precision and recall by combining them into a single metric. Other indicators are numeric, indicate the classifier's overall discriminative power (ROC-AUC), and consider performance aspects beyond detection accuracy, such as real-time operational acceptance (latency) and robustness under uncertain traffic conditions.



International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|

DOI: 10.15662/IJSRAT.2024.0706007

6.1. Metrics for Fault Detection

Robustness, generalization capability, and the ability to produce timely predictions are of primary importance for classification models aimed at detecting faults on real-world smart grids. Consequently, evaluation metrics must fulfill these practical demands in accordance with the definition of a fault. Principle accuracy-related metrics, namely precision, recall, and the F1 score, should be improved as far as possible, with a special focus on either precision or recall according to the traffic condition of interest.

A substantial fraction of the monitored conditions must be in the normal class without sacrificing the ability to detect faults of all kinds. This requirement has two consequences: a high value of the area under the receiver operating characteristic curve and a specific-case evaluation of the model based on the true negative rate. Time-related metrics, namely latency and jitter, should be as low as possible without significantly compromising other accuracy-related metrics. The models are also expected to perform satisfactorily when evaluated on subsamples with varied features, classes, or any other relevant properties for which ground-truth labels are available. A family of conditional precision/recall/F1 plots provides a generic procedure for assessing this ability.

6.2. Case Studies and Benchmarking

The full range of smart grid components is considered through a set of case studies covering key failure types and fault events. Testing is performed on various publicly available datasets adapted to the pipelines. The end-to-end deployment phase includes pipeline orchestration through airflow, along with full operational deployment. A data engineering solution that accommodates data from various sources enables fault detection during normal operations of the smart grid with a latency requirement of less than ten seconds.

The proposed architectures take into account a comparison between batch and streaming processes in both training and prediction phases. The latency requirements and data freshness needs are at the fore of the discussion during model training. Considering the common challenges faced by distribution grids, these pipelines are tested using a range of datasets. Each of the AI-based detection categories is tested under its respective condition, whether that be unlabelled or a small set of labels for semi-supervised learning.

VII. CONCLUSION

An integrated approach to data engineering for smart grid fault detection pipelines is presented. Requirements of the pipeline architecture serve as consistent and coherent guidelines to develop data ingestion and quality monitoring, AI methods, and model training and deployment. Addressing the multi-faceted data engineering aspects surrounding the pipeline enables operational AI that supports reliable fault detection without status degradation in fall detection or hyperparameterization overhead. In addition, this streamlined data engineering solution employs multiple end-to-end data pipelines that underpin fault detection and reliability analysis.

Automated detection of smart grid faults provides vital operational information to appropriate authorities or specialists. While longitudinal data streams can be interrupted by network intensities, CAT, event trigger, or PMU systems, fault identification remains resilient owing to non-homogeneity and full cycle consideration in function/structure-based strategy sets adopted. Nevertheless, ensuring model viability through continuous validation and development under changing conditions remains vital. Thus, specialized supervised, semi-supervised, and unsupervised modeling strategies cater to the variable learning environments linked to different fault types.

Stream / Source	Typical cadence / requirement	Notes
Smart-sensor telemetry (vibration/temperature)	1 second	Every second; internal buffer ≤ 1,296 records (~21.6 min)
SCADA	≥ 1 minute	Max latency 5 minutes
PMU	≥ 1 minute	Max latency 5 minutes; 10-sample compression at ingestion

Table: Pipeline timing summary (from paper + derived)

International Journal of Science, Research and Technology (IJSRAT)

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|

DOI: 10.15662/IJSRAT.2024.0706007

7.1. Future Trends

Several trends are emerging that will shape the future of building AI-enabled data engineering pipelines for fault detection in electrical grids. First, the growing demand for low-latency and high-throughput detection capabilities is driving a shift toward stream processing. Streaming pipelines allow AI models to be continuously updated with the latest data while facilitating real-time operation and the delivery of newly detected faults to operators. Second, the need for continual development and monitoring of AI models is leading to the incorporation of S-DLC and CI/CD practices into end-to-end pipelines. Such processes automate model retraining and quality checks as part of a central development environment, reducing the workload associated with deploying AI in operational settings. The ongoing digitalization of electrical grids is further broadening the spectrum of available data sources. Data from Advanced Metering Infrastructure and Distribution Automation, as well as other smart devices in the grid environment, are becoming increasingly accessible. These sources are expected to supplement the existing data pool and help mitigate issues caused by insufficiently informative datasets, thus improving model reliability.

In addition to widening the data availability horizon, the integration of data engineering methods with AI for fault detection is likely to gain further attention. Possible application areas include the development of dedicated databases, data governance mechanisms, high-quality data generation through simulation, and tools for manual labeling of fault occurrences. In fact, the continuing evolution of the AI implementation landscape in electrical grids is expected to further drive the integration of data engineering pipelines in additional domains, improving the implemented detection capabilities and the reliability of resulting outcomes.

REFERENCES

[1] Davuluri, P. S. L. N. (2024). AI-Driven Data Governance Frameworks for Automated Regulatory Reporting and Audit Readiness. Metallurgical and Materials Engineering, 30(4), 996–1010. Retrieved from https://metall-mater-eng.com/index.php/home/article/view/1936

[2] Guntupalli, R. (2024). Enhancing Cloud Security with AI: A Deep Learning Approach to Identify and Prevent Cyberattacks in Multi-Tenant Environments. Available at SSRN 5329132.

[3] Anderson, T. (Ed.). (2008). The theory and practice of online learning (2nd ed.). AU Press.

[4] Singireddy, J. (2024). Ai-enhanced tax preparation and filing: Automating complex regulatory compliance. European Data Science Journal (EDSJ) p-ISSN, 3050-9572.

[5] Anikina, Z., & Yakimenko, O. (2015). EdTech and digital learning environments in higher education: Trends and outcomes. Procedia - Social and Behavioral Sciences, 206, 432–436.

[6] Inala, R. Revolutionizing Customer Master Data in Insurance Technology Platforms: An AI and MDM Architecture Perspective.

[7] Baker, R. S. J. d. (2010). Data mining for education. In B. McGaw, E. Baker, & P. Peterson (Eds.), International encyclopedia of education (3rd ed., pp. 112–118). Elsevier.

[8] Paleti, S. (2024). Transforming Financial Risk Management with AI and Data Engineering in the Modern Banking Sector. American Journal of Analytics and Artificial Intelligence (ajaai) with ISSN 3067-283X, 2(1).

[9] Bandura, A. (1986). Social foundations of thought and action: A social cognitive theory. Prentice Hall.

[10] Kummari, D. N., & Burugulla, J. K. R. (2023). Decision Support Systems for Government Auditing: The Role of AI in Ensuring Transparency and Compliance. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 493-532.

[11] Bichsel, J. (2013). The state of e-learning in higher education: An eye toward growth and increased access. EDUCAUSE Center for Applied Research.

[12] Sheelam, G. K., & Koppolu, H. K. R. (2024). From Transistors to Intelligence: Semiconductor Architectures Empowering Agentic AI in 5G and Beyond. Journal of Computational Analy- sis and Applications(JoCAAA), 33(08), 4518-4537.

[13] Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher, 13(6), 4–16.

[14] Meda, R. (2024). Agentic AI in Multi-Tiered Paint Supply Chains: A Case Study on Efficiency and Responsiveness. Journal of Compu-tational Analysis and Applications (JoCAAA), 33(08), 3994-4015.

[15] Brinton, C. G., Rill, R., Ha, S., Chiang, M., Smith, R., & Ju, W. (2014). Individualization for education at scale: Improving online learning via automated student modeling. IEEE Transactions on Learning Technologies, 7(4), 347–363.

[16] Singireddy, S. (2024). The Integration of AI and Machine Learning in Transforming Underwriting and Risk Assessment Across Personal and Commercial Insurance Lines. Journal of Computational Analy- sis and Applications(JoCAAA), 33(08), 3966-3991.

[17] Challa, K. (2024). Neural Networks in Inclusive Financial Systems: Generative AI for Bridging the Gap Between Technology and Socioeconomic Equity. MSW Management Journal, 34(2), 749-763.
[18] A Scalable Web Platform for AI-Augmented Software Deployment in Automotive Edge Devices via Cloud Services. (2024).
[19] Campos, P., Desmarais, M., & S. d. Baker, R. (2011). Mining constraint-based tutor data for personalization. Educational Data Mining, 3(1), 13–26.
[20] Garapati, R. S. (2023). Optimizing Energy Consumption in Smart Build-ings Through Web-Integrated AI and Cloud-Driven Control Systems.
[21] Nandan, B. P., & Chitta, S. S. (2023). Machine Learning Driven Metrology and Defect Detection in Extreme Ultraviolet (EUV) Lithography: A Paradigm Shift in Semiconductor Manufacturing. Educational Administration: Theory and Practice, 29(4), 4555-4568.
[22] Christensen, C. M., Horn, M. B., & Johnson, C. W. (2008). Disrupting class: How disruptive innovation will change the way the world learns. McGraw-Hill.
[23] Singireddy, J. (2024). AI-Driven Payroll Systems: Ensuring Compliance and Reducing Human Error. American Data Science Journal for Advanced Computations (ADSJAC) ISSN, 3067-4166.
[24] Clow, D. (2013). An overview of learning analytics. Teaching in Higher Education, 18(6), 683–695.
[25] Challa, K. (2024). Artificial Intelligence and Generative Neural Systems: Creating Smarter Customer Support Models for Digital Financial Services. Journal of Computational Analysis & Applications, 33(8).
[26] Emerging Role of Agentic AI in Designing Autonomous Data Products for Retirement and Group Insurance Platforms. (2024). MSW Management Journal, 34(2), 1464-1474.
[27] Drachsler, H., & Greller, W. (2016). Privacy and analytics: It's a DELICATE issue. In Proceedings of the 6th International Conference on Learning Analytics & Knowledge (pp. 89–98). ACM.
[28] Singireddy, S. (2024). Predictive Modeling for Auto Insurance Risk Assessment Using Machine Learning Algorithms. European Advanced Journal for Emerging Technologies (EAJET)-p-ISSN, 3050-9734.
[29] Inala, R. AI-Powered Investment Decision Support Systems: Building Smart Data Products with Embedded Governance Controls
[30] Ellis, R. A., & Goodyear, P. (2010). Students' experiences of e-learning in higher education: The ecology of sustainable innovation. Routledge.
[31] Sriram, H. K., Challa, S. R., Challa, K., & ADUSUPALLI, B. (2024). Strategic Financial Growth: Strengthening Investment Management, Secure Transactions, and Risk Protection in the Digital Era. Secure Transactions, and Risk Protection in the Digital Era (November 10, 2024).
[32] Nandan, B. P. (2024). Semiconductor Process Innovation: Leveraging Big Data for Real-Time Decision-Making. Journal of Computational Analysis and Applications (JoCAAA), 33(08), 4038-4053.
[33] Gašević, D., Dawson, S., & Siemens, G. (2015). Let's not forget: Learning analytics are about learning. TechTrends, 59(1), 64–71.
[34] Aitha, A. R. (2024). Generative AI-Powered Fraud Detection in Workers' Compensation: A DevOps-Based Multi-Cloud Architecture Leveraging, Deep Learning, and Explainable AI. Deep Learning, and Explainable AI (July 26, 2024).
[35] Gobert, J. D., Baker, R. S., & Wixon, M. (2015). Operationalizing affective constructs in educational software. In Proceedings of EDM 2015 (pp. 1–8). International Educational Data Mining Society.
[36] Graesser, A. C., Conley, M., & Olney, A. (2012). Intelligent tutoring systems. In K. R. Harris, S. Graham, & T. Urdan (Eds.), APA educational psychology handbook (pp. 451–473). American Psychological Association.
[37] Guntupalli, R. (2023). AI-Driven Threat Detection and Mitigation in Cloud Infrastructure: Enhancing Security through Machine Learning and Anomaly Detection. Available at SSRN 5329158.
[38] Greller, W., & Drachsler, H. (2012). Translating learning into numbers: A generic framework for learning analytics. Educational Technology & Society, 15(3), 42–57.
[39] Griffin, P., McGaw, B., & Care, E. (Eds.). (2012). Assessment and teaching of 21st century skills. Springer.
[40] Halawa, S., Greene, D., & Mitchell, J. (2014). Dropout prediction in MOOCs using learner activity features. In Proceedings of the 2nd European MOOC Stakeholder Summit (pp. 58–65).
[41] Guntupalli, R. (2024). AI-Powered Infrastructure Management in Cloud Computing: Automating Security Compliance and Performance Monitoring. Available at SSRN 5329147.
[42] Hattie, J. (2009). Visible learning: A synthesis of over 800 meta-analyses relating to achievement. Routledge.
[43] Vardhan Kumar Bandi, V. D. (2024). Automated Feature Engineering Systems in Large-Scale Healthcare Data Environments. Journal of Neonatal Surgery, 13(1), 2127-2141.
[44] Nandan, B. P. (2024). Revolutionizing Semiconductor Chip Design through Generative AI and Reinforcement Learning: A Novel Approach to Mask Patterning and Resolution Enhancement. International Journal of Medical Toxicology and Legal Medicine, 27(5), 759-772.

[45] Ionita, A., & Dede, C. (2016). Personalized learning and learning analytics in higher education. Journal of Learning Analytics, 3(1), 1–12.
[46] Singireddy, S. (2024). Applying deep learning to mobile home and flood insurance risk evaluation. American Advanced Journal for Emerging Disciplinaries (AAJED) ISSN, 3067-4190.
[47] Kizilcec, R. F., Piech, C., & Schneider, E. (2013). Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In Proceedings of the 3rd International Conference on Learning Analytics and Knowledge (pp. 170–179). ACM.
[48] Kolla, S. H. (2024). RETRIEVAL-AUGMENTED GENERATION WITH SMALL LLMS FOR KNOWLEDGE-DRIVEN DECISION AUTOMATION IN ENTERPRISE SERVICE PLATFORMS. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 15(3), 476–486
[49] Koper, R. (2014). Conditions for effective smart learning environments. Smart Learning Environments, 1(1), 1–17.
[50] Kushvanth Chowdary Nagabhyru. (2023). Accelerating Digital Transformation with AI Driven Data Engineering: Industry Case Studies from Cloud and IoT Domains. Educational Administration: Theory and Practice, 29(4), 5898–5910. https://doi.org/10.53555/kuey.v29i4.10932.
[51] Kuh, G. D. (2009). The national survey of student engagement: Conceptual and empirical foundations. New Directions for Institutional Research, 141, 5–20.
[52] Mashetty, S., Challa, S. R., ADUSUPALLI, B., Singireddy, J., & Paleti, S. (2024). Intelligent Technologies for Modern Financial Ecosystems: Transforming Housing Finance, Risk Management, and Advisory Services Through Advanced Analytics and Secure Cloud Solutions. Risk Management, and Advisory Services Through Advanced Analytics and Secure Cloud Solutions (December 12, 2024).
[53] Lahari Pandiri, "AI-Powered Fraud Detection Systems in Professional and Contractors Insurance Claims," International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE), DOI 10.17148/IJIREEICE.2024.121206.
[54] Long, P., & Siemens, G. (2011). Penetrating the fog: Analytics in learning and education. EDUCAUSE Review, 46(5), 31–40.
[55] Singireddy, S. (2024). Leveraging Artificial Intelligence and Agentic AI Models for Personalized Risk Assessment and Policy Customization in the Modern Insurance Industry: A Case Study on Customer-Centric Service Innovations. Journal of Computational Analysis and Applications (JoCAAA), 33(08), 2532-2545.
[56] Aitha, A. R. (2023). CloudBased Microservices Architecture for Seamless Insurance Policy Administration. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 607-632.
[57] Mayer, R. E. (2009). Multimedia learning (2nd ed.). Cambridge University Press.
[58] Davuluri, P. N. Integrating Artificial Intelligence into Event-Driven Financial Crime Compliance Platforms.
[59] Milligan, C., & Littlejohn, A. (2017). Why study on a MOOC? The motives of students and professionals. International Review of Research in Open and Distributed Learning, 18(2), 92–102.
[60] Reddy Segireddy, A. (2024). Federated Cloud Approaches for Multi-Regional Payment Messaging Systems. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 15(2), 442–450. https://doi.org/10.61841/turcomat.v15i2.15464.
[61] Challa, K. (2024). Enhancing credit risk assessment using AI and big data in modern finance. American Data Science Journal for Advanced Computations (ADSJAC) ISSN, 3067-4166.
[62] Nelson, K. J., Creagh, T. A., & Clarke, J. A. (2012). Engagement and retention in the first year: Strategies for support. Higher Education Research & Development, 31(2), 217–229.
[63] Meda, R. (2023). Intelligent Infrastructure for Real-Time Inventory and Logistics in Retail Supply Chains. Educational Administration: Theory and Practice.
[64] Pardo, A., & Siemens, G. (2014). Ethical and privacy principles for learning analytics. British Journal of Educational Technology, 45(3), 438–450.
[65] Pardo, A., Mirriahi, N., Martinez-Maldonado, R., Jovanovic, J., Dawson, S., & Gašević, D. (2019). A systematic review of learning analytics dashboards for higher education. Educational Technology Research and Development, 67, 1323–1346.
[66] Papamitsiou, Z., & Economides, A. A. (2014). Learning analytics and educational data mining in practice: A systematic literature review. Educational Technology & Society, 17(4), 49–64.
[67] Segireddy, A. R. (2024). Machine Learning-Driven Anomaly Detection in CI/CD Pipelines for Financial Applications. Journal of Computational Analysis and Applications, 33(8).
[68] Picciano, A. G. (2012). The evolution of big data and learning analytics in education. Journal of Asynchronous Learning Networks, 16(3), 9–20.
[69] Bandi, V. D. V. K. (2024). Intelligent Data Platforms For Personalized Retail Analytics At Scale. Metallurgical and Materials Engineering, 30 (4), 1011–1027.

|Bimonthly | Peer-Reviewed | Scholarly Journal|

|ISSN 2454-3853|www.ijsrat.com|editor@ijsrat.com|Vol.7, Issue 6, November-December 2024|

DOI: 10.15662/IJSRAT.2024.0706007

[70] Reigeluth, C. M., & Carr-Chellman, A. A. (Eds.). (2009). Instructional-design theories and models: Building a common knowledge base (Vol. 3). Routledge.
[71] Kaulwar, P. K. (2024). Agentic Tax Intelligence: Designing Autonomous AI Advisors for Real-Time Tax Consulting and Compliance. Journal of Computational Analysis and Applications (JoCAAA), 33(08), 2757-2775.
[72] Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(3), e1355.
[73] Rose, C. P., & Siemens, G. (2014). Shared tasks for learning analytics. Journal of Learning Analytics, 1(1), 1–5.
[74] Rongali, S. K. (2024). Federated and Generative AI Models for Secure, Cross-Institutional Healthcare Data Interoperability. Journal of Neonatal Surgery, 13(1), 1683-1694.
[75] Sadler, D. R. (1989). Formative assessment and the design of instructional systems. Instructional Science, 18(2), 119–144.
[76] Salomon, G., & Perkins, D. N. (1998). Individual and social aspects of learning. Review of Research in Education, 23, 1–24.
[77] Nagabhyru, K. C. (2024). Data Engineering in the Age of Large Language Models: Transforming Data Access, Curation, and Enterprise Interpretation. Computer Fraud and Security. [78] Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. EDUCAUSE Review, 46(5), 30–40.
[79] Simpson, O. (2012). Supporting students in online, open and distance learning (2nd ed.). Routledge.
[80] Slade, S., & Prinsloo, P. (2013). Learning analytics: Ethical issues and dilemmas. American Behavioral Scientist, 57(10), 1510–1529.
[81] Nagubandi, A. R. (2023). Advanced Multi-Agent AI Systems for Autonomous Reconciliation Across Enterprise Multi-Counterparty Derivatives, Collateral, and Accounting Platforms. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 653-674.