Street Art & Urban Creativity | Vol. 11, No. 1, 2025 | ISSN 2183-9956
Street Art & Urban Creativity. Scientific Journal

A Hybrid Rule-Based and Machine Learning Framework for Detecting Data Quality Anomalies in Electronic Health Records

SAINATH REDDY[1]
[1]Research Scientist, Albany, NY, USA

Abstract

Data quality anomalies in electronic health records, including impossible values, unit errors, inconsistent fields, copied-forward results, calibration drift, and duplicates, corrupt research, quality measurement, and the machine learning models built on the data. Rule-based checks are transparent and precise but cover only what has been written down, while learning-based detectors can find the unexpected but are opaque and need labels. This paper presents a hybrid framework in which rules act as high-precision detectors, as features for a learning layer, and as a source of weak labels; an isolation forest and a gradient-boosted classifier cover the gaps that rules leave; a fusion step ranks records under a review budget; and reviewer feedback drives active learning and rule maintenance. We evaluate the framework on a synthetic EHR data set of 30,560 records with eight classes of injected anomaly, using a time-based split. At a 6 percent review budget, rules alone found 38.6 percent of anomalies and an unsupervised isolation forest 51.1 percent, a label-free cascade of the two found 69.5 percent, and a supervised model given the rule flags found 93.0 percent with an F1 of 0.895. Rules caught range, unit, and covered cross-field violations almost perfectly, while learning supplied contextual, batch-drift, and uncovered cases. Weak supervision from rules extended coverage to near-neighbours but not to new classes, and active learning with rule features roughly doubled recall relative to random labelling at 400 labels. The data are synthetic and the absolute figures should not be transferred to real EHRs. We give a deployment and governance guide and an evaluation template for real data.

Keywords: electronic health records; data quality; anomaly detection; rule-based validation; machine learning; weak supervision; active learning; isolation forest.

1. Introduction

Electronic health records were built to support care and billing, and most of the data in them was never designed for analysis. Yet the same records now feed clinical research, quality measurement, population health programmes, regulatory reporting, and, increasingly, machine learning models that influence clinical decisions. The reliability of every one of these uses depends on the quality of the underlying data, and that quality is uneven. Values are entered in the wrong unit, devices malfunction and report impossible readings, interfaces between systems corrupt or duplicate messages, clinicians copy forward yesterday's results, laboratory instruments drift out of calibration, and fields are populated in ways that are internally inconsistent. Some of these errors are glaring, and some are subtle and plausible on their face. The latter are the more dangerous because they pass unnoticed into analyses and models.

Two broad approaches to detecting such anomalies exist, and each has a characteristic strength and weakness. Rule-based approaches encode knowledge as explicit checks: a heart rate must lie within physiologic limits, a systolic pressure must exceed the diastolic, a discharge cannot precede an admission, a male patient cannot be pregnant. Rules are transparent, deterministic, auditable, and precise for the violations they describe, and they are the backbone of data quality tooling. But they cover only what someone has thought to write down, they are brittle at the edges of physiologic plausibility, where a legitimate extreme value is indistinguishable from an error without context, and they cannot express the subtle, contextual, and temporal anomalies that matter most. Machine learning approaches model what normal

data looks like and flag departures from it. They can catch patterns that nobody wrote a rule for, and they can use context, such as the patient's history and conditions, but they are opaque, they need either labels, which are scarce, or assumptions about the proportion of anomalies, and they offer no guarantee on the obvious cases that a simple rule would catch with certainty.

The thesis of this paper is that the two approaches are complements, and that a framework which combines them deliberately can outperform either alone on the criteria that matter in practice: how many true anomalies are found for a fixed amount of human review effort, how reliably the clear-cut violations are caught, how well the system adapts as new anomaly types appear, and how readily its alerts can be explained to the data steward who must act on them. We present a hybrid framework in which rules play three roles, as high-precision detectors, as a source of features for the learning layer, and as a source of weak labels that reduce the need for manual annotation, and in which the learning layer covers the gaps that rules leave. A fusion step ranks records for review under an explicit budget, and a feedback loop turns reviewers' judgements into new labels, rule edits, and updated thresholds.

We make four contributions. First, we specify the framework: the rule engine, the feature builder, the learning layer, the fusion and triage logic, and the feedback loop, with formal definitions of each component. Second, we describe a synthetic EHR generator with eight classes of injected anomaly, including contextual anomalies, copy-forward, batch calibration drift, and near-duplicates, which allows detection to be evaluated against known ground truth by anomaly type. Third, we evaluate rule-only, learning-only, and several hybrid configurations at a fixed review budget, together with a label-efficiency study of active learning. Fourth, we draw implications for the deployment and governance of such a system.

As with other studies in this series, the evidence is synthetic. We did not have access to a real EHR data set, and the generator, anomalies, and rule library are ours. The results therefore show how the components of the framework behave and complement one another under controlled conditions. They are not estimates of performance on any hospital's data, where anomalies are messier, labels are noisier, and the balance between types is different. The framework and the evaluation design are intended as a template for a real evaluation.

The paper is organised as follows. Section 2 gives background on data quality in EHRs. Section 3 reviews related work. Section 4 presents the framework. Section 5 describes the experimental design, and Section 6 reports results. Section 7 discusses deployment and governance. Section 8 states limitations, and Section 9 concludes.

2. LITERATURE REVIEW

2.1 Dimensions of EHR data quality

The literature on EHR data quality describes it along several dimensions. A widely used harmonised terminology distinguishes conformance, whether data comply with specified formats, value sets, and constraints, completeness, whether expected data are present, and plausibility, whether data values are believable, including by comparison with other values and with expectations over time. A review of methods for EHR data quality assessment identified five dimensions that recur in the literature: completeness, correctness, concordance, plausibility, and currency. The broader information-quality literature emphasises that quality is fitness for use by data consumers and includes dimensions such as accuracy, timeliness, and consistency. Our focus is on the detection of anomalies in the conformance, plausibility, and concordance dimensions, that is, values and records that are invalid, implausible, or inconsistent, whether the cause is an error of entry, a device fault, a system fault, or a genuine but rare clinical state.

An important distinction is between an anomaly and an error. An anomaly is a record that deviates from what is expected. An error is a record that does not represent reality. The two overlap but are not the same. A potassium of 7.2 mmol/L is statistically unusual and may be a true value in a patient with renal failure, or it may be a haemolysed sample, or a mistyped entry. The framework therefore produces candidates for review and does not correct or delete data automatically, a design choice with consequences for safety and for the trust of clinical and data teams.

2.2 Sources of anomalies

Anomalies arise at every stage of the data lifecycle. At capture, manual entry produces typographical errors, transposed digits, values in the wrong field, and values in the wrong unit, for example glucose entered in mmol/L in a field defined in mg/dL. Devices produce artefacts,

spurious readings, and stuck values. Interfaces and migrations introduce duplicated records, truncated messages, mismatched identifiers, and unit conversions. Workflow practices such as copying forward earlier values create records that are plausible but stale. Laboratory instruments drift or are recalibrated, producing systematic shifts in all results from a site over a period. Merging of records across systems creates inconsistencies, such as a record with an impossible combination of sex and pregnancy status. Table 1 lists the classes of anomaly used in this paper and their typical sources.

Table 1. Anomaly classes in the synthetic data and their typical sources.

Class	Example	Typical source	Injected (records)
Range	Heart rate 0 or 450; potassium 14	Device fault, entry error	242
Unit	Glucose in mmol/L labelled mg/dL; creatinine in umol/L; temperature in F	Entry or interface unit mix-up	212
Cross-field (rule-covered)	Systolic below diastolic; discharge before admission; male marked pregnant	Field swap, merge error	121
Cross-field (not covered)	Height and weight swapped	Entry error	121
Contextual	Normal creatinine in advanced kidney disease; high haemoglobin in documented anaemia	Wrong-patient entry, merge error	121
Temporal	Implausible jump from previous value; values copied from previous visit	Typo, transposition, copy-forward	242
Batch drift	Sodium +7, potassium +15%, haemoglobin +1.0 across a site-week	Instrument calibration	342 (4 site-weeks)
Duplicate	Repeated encounter, exact or slightly perturbed	Interface or merge fault	212

The classes differ in the evidence needed to detect them. Range violations and unit errors can be detected from a single value, given a rule. Cross-field and temporal violations need two or more fields from the same record. Contextual anomalies need the patient's conditions and expected values. Copy-forward and transposition errors need the patient's previous records. Batch drift is invisible in any individual record and can be seen only in the aggregate for a site and period. Duplicates need comparison across records. A detection system must therefore combine several kinds of evidence, and this is a central reason for a layered design.

3. RELATED WORK

Data quality tooling in industry and in observational research is largely rule based. Systems for large-scale data validation let users declare constraints on columns and relationships, compute metrics, and monitor them over time, and have been adopted in machine learning pipelines to catch schema and distribution errors before they reach models. In health data, open-source tools run libraries of checks against data in a common data model and report conformance, completeness, and plausibility results, and have been used to increase trust in observational data. Such tools provide the kind of rule engine that our framework assumes, and the contribution of this paper is not a new rule engine but the combination of a rule engine with learning.

Outlier and anomaly detection methods have been surveyed extensively. Methods include statistical models, nearest-neighbour and density-based approaches, clustering, tree-based isolation methods, one-class classifiers, and deep autoencoders. Isolation forests detect anomalies as points that are isolated by few random splits and scale well to large tabular data. Comparative evaluations of unsupervised methods on multivariate data show that performance depends strongly on the nature of the anomalies, and that there is no uniformly best method. In clinical data, outlier-based methods have been proposed for detecting unusual patient-management actions in intensive care, and semi-supervised encoding has been used to detect outliers in clinical observations. A line of work on deep semi-supervised anomaly detection shows that a small number of labelled anomalies can improve on purely unsupervised methods.

Two bodies of work are directly relevant to the weak and limited labelling that characterises data quality work. Weak supervision lets users express heuristics, such as rules, as labelling functions whose noisy outputs are combined to produce training labels, so that models can be trained without hand labelling. Active learning chooses the examples whose labels would be most informative, and so reduces the labelling effort needed to reach a given performance. In data cleaning, systems have combined integrity constraints with probabilistic inference and with active labelling to detect and repair errors in relational data. We apply these ideas to EHR anomaly detection, using rules both as detectors and as sources of weak labels and features, and active learning to focus reviewers' attention.

Finally, the clinical informatics literature stresses that secondary use of EHR data is hazardous without attention to data quality, and that data quality problems can bias research findings and degrade models. Our work responds to this call with a practical architecture, and with an evaluation approach that reports detection by anomaly type and at a fixed review budget, which are the quantities that matter to the data stewards who will use it.

4. The Hybrid Framework

Figure 1 shows the architecture, which has six components. The rule engine and the feature builder consume the records. The learning layer consumes features and, optionally, the output of the rule engine. A fusion step ranks the records, a triage queue presents them to reviewers with explanations, and a feedback loop captures reviewers' decisions.

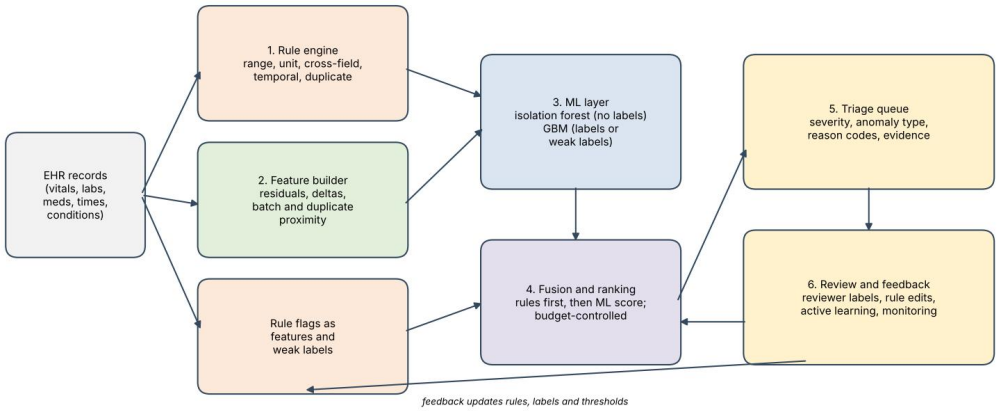

Figure 1. Architecture of the hybrid framework. Rules act as detectors, as features, and as weak labels; the ML layer covers what rules do not; fusion ranks records under a review budget; reviewer feedback updates rules, labels, and thresholds.

4.1 Rule engine

The rule engine is a library of deterministic checks. Each rule j has a predicate that evaluates to one when a record violates it, and each has a severity weight w-j. The rule score of a record is the largest weighted violation,

$$\rho_j(x) = \mathbf{1}[x \text{ violates rule } j], \qquad r(x) = \max_j \; w_j \, \rho_j(x) \tag{1}$$

Table 2 lists the 12 rules used in our experiments, in four families. Range rules test whether a vital sign or laboratory value falls outside limits, and are the most common. Unit-plausibility rules are range rules with narrow limits chosen so that a value that is plausible only in another unit, such as a creatinine of 80 labelled mg/dL, is caught. Cross-field rules test relations among fields of one record, and temporal rules test the ordering of timestamps. A duplicate rule detects exact repeats. Rules have a deliberate limitation that mirrors practice. Some, such as the potassium and heart rate limits, are set to include the extremes that can occur in clinical practice, and so have false positives when a legitimate extreme value occurs. The rule library is also incomplete: it contains no rule for implausible body mass index, no rule on patient context, and no rule for near-duplicates, as will often be true of a real library.

Table 2. The rule library used in the experiments (12 rules).

Family	Rules	Note
Range (8)	Heart rate, SpO2, temperature, potassium, respiratory rate, glucose, creatinine, sodium outside limits	Set to include clinical extremes; some false positives
Cross-field (2)	Systolic not above diastolic; male with pregnancy flag	No BMI or context rules
Temporal (1)	Discharge before admission or lab after discharge	-
Duplicate (1)	Exact repeat of key fields for same patient	Near-duplicates not covered

4.2 Feature builder

The feature builder produces inputs for the learning layer that carry contextual, temporal, and aggregate evidence. Table 3 lists them. Residual features compare a value with the value expected for the patient, given age, sex, and recorded conditions, in standard units. For a quantity c with expected value mu-c and spread sigma-c, the residual is

$$e_c(x) = \frac{\log c - \log \mu_c(\text{age, sex, conditions})}{\sigma_c} \quad (2)$$

These are the features that expose contextual anomalies, such as a normal creatinine in a patient with chronic kidney disease. Temporal features compare a value with the same patient's previous record, in standard units,

$$\Delta_c = \frac{|c_t - c_{t-1}|}{\sigma_c} \quad (3)$$

and count the number of fields that are identical to the previous record, which exposes copy-forward. Batch features summarise, for each site and week, the mean standardised deviation of selected laboratory values, scaled by the square root of the number of records so that large, consistent shifts stand out,

$$b_{s,w} = \bar{z}_{s,w}\sqrt{n_{s,w}} \quad (4)$$

which exposes calibration drift that is invisible in individual records. A duplicate-proximity feature measures the similarity of a record to others from the same patient within a day. None of these features uses labels. Derived quantities such as body mass index and pulse pressure are added, since these make implausible combinations visible to a learner.

Table 3. Features used by the learning layer (38 features; rule flags add 13 in the hybrid models).

Group	Features	Evidence for
Raw	Age, sex, height, weight, conditions, vitals, laboratory values (20)	General context
Derived	BMI, pulse pressure	Implausible combinations
Residual	Creatinine and haemoglobin vs value expected for age, sex, conditions	Contextual anomalies
Temporal	Deltas vs previous visit for 8 quantities; count of identical fields	Jumps, copy-forward
Duplicate proximity	Similarity to other records of same patient within a day	Near-duplicates
Batch	Site-week standardised deviation of sodium, potassium, haemoglobin; combined	Calibration drift

4.3 Learning layer

The learning layer has two modes. The unsupervised mode fits an isolation forest to the feature matrix of historical records, without labels. The isolation forest scores a record by the expected path length needed to isolate it in random trees, normalised by the average path length for the sample size n,

$$s(x, n) = 2^{-\mathbb{E}[h(x)]/c(n)} \quad (5)$$

so that scores near one indicate records that are isolated easily. The supervised mode trains a gradient-boosted tree classifier on reviewer-labelled records, with the features of Table 3 and, in the hybrid configurations, the rule flags as additional inputs. Giving the learner the rule outputs lets it learn which rules are reliable in which contexts, for example that a potassium rule violation in a patient with advanced renal disease is less likely to be an error, and lets it spend its

capacity on what the rules do not cover. A weakly supervised mode trains the same classifier without reviewer labels, using the rules to define pseudo-labels, as described below.

4.4 Fusion and triage

The fusion step turns the outputs of the rule engine and the learning layer into a ranking. A general form is a convex combination of the rule score r and the model score m,

$$\text{score}(x) = \lambda\, r(x) + (1 - \lambda)\, m(x) \tag{6}$$

In the cascade, which we use in unsupervised and weakly supervised settings, the weight on the rule score is large enough that every record with a rule violation precedes every record without one, and the learning score orders the remainder. The ranking is cut at a review budget, a fraction of records that the data stewards can inspect, which is the operational constraint that gives the comparison its meaning. Each flagged record is presented with reason codes, namely the rules violated, the features that contributed most to the model score, and the evidence, such as the previous record or the site-week aggregate. The triage queue groups records by probable anomaly type and by severity, determined by the clinical significance of the affected field, so that, for instance, a likely wrong allergy or medication dose is reviewed before a likely wrong respiratory rate.

4.5 Weak supervision and active learning

Reviewer labels are the scarce resource. Weak supervision uses the rules as a free source of labels. Records that violate a rule are labelled as anomalies, and records that violate no rule and are not unusual according to the unsupervised score are labelled as normal,

$$\tilde{y}(x) = 1 \;\text{ if }\; r(x) > 0; \qquad \tilde{y}(x) = 0 \;\text{ if }\; r(x) = 0 \;\text{ and }\; s_{\text{IF}}(x) < q_{0.90} \tag{7}$$

where s-IF is the isolation forest score and q-0.90 its 90th percentile. A classifier trained on these pseudo-labels, using the features but not the rule flags themselves, can generalise from the rule-detected anomalies to near-neighbours that the rules miss. The labels are noisy, since rule false positives become positive examples and anomalies that no rule covers become negative ones, and the approach will propagate the coverage of the rule library. Active learning then spends the reviewers' effort where it is most useful. A model trained on the current labels computes, for each unlabelled record, the uncertainty of its prediction p-hat,

$$u(x) = 1 - \left|2\,\hat{p}(x) - 1\right| \tag{8}$$

and the records with the highest uncertainty are sent for review. The reviewer's decisions are added to the training set, and they also inform rule maintenance: a pattern of confirmed anomalies that no rule detects is a candidate for a new rule, and a rule whose alerts are consistently rejected is a candidate for revision.

5. Experimental Design

5.1 Synthetic EHR data

We generated a synthetic EHR data set of 30,560 records for about 8000 simulated patients over one year, with about four encounters per patient. Each record contains demographics, height and weight, recorded conditions (chronic kidney disease, anaemia, diabetes, pregnancy), admission and discharge times, a laboratory time, and twelve vital-sign and laboratory values: heart rate, systolic and diastolic pressure, respiratory rate, oxygen saturation, temperature, sodium, potassium, creatinine, glucose, haemoglobin, and white-cell count. Values are drawn from physiologically motivated distributions that depend on age, sex, and conditions, with patient-level persistent effects so that previous records are informative. About 1.4 percent of clean records contain legitimate clinical extremes, such as a true potassium of 6.5 or a heart rate of 160, which are not anomalies but may trigger range rules.

Anomalies were injected into about 5.3 percent of records, 1613 in all, in the eight classes of Table 1: range violations, unit errors, cross-field violations that the rule library covers, cross-field violations that it does not cover, contextual anomalies, temporal anomalies that include implausible jumps and copy-forward from the previous record, batch calibration drift affecting four site-weeks, and duplicates, half of them exact and half near-duplicates. The first 70 percent of the year, 21,362 records, is the training period and the last 30 percent, 9198 records with 511

anomalies, is the test period. The split is by time, as it would be in deployment, and two of the four drift batches fall in each period.

5.2 Methods compared

We compare six methods. Rules only uses the 12 rules of Table 2 and flags every record that violates any of them. The isolation forest is the unsupervised learner, applied to the features of Table 3. The supervised GBM is trained on the training labels with the features of Table 3 and no rule outputs. Three hybrids are evaluated: the cascade of rules followed by the isolation forest, the weakly supervised classifier with the rules as first priority, and the supervised GBM that also receives the rule flags. All learning methods are evaluated at a review budget of 6 percent of the test records, a level slightly above the 5.6 percent prevalence of anomalies in the test period, since a data steward will usually review somewhat more than the true number. Rules only is evaluated at the records it flags, 3.4 percent of the test records.

5.3 Metrics

We report precision and recall at the review budget, and the F1 score, and, for methods that produce a ranking, the average precision. For a ranked list truncated at k records, with TP-k true anomalies among them and P anomalies overall, recall and precision at k are

$$\text{Recall@}k = \frac{\text{TP}_k}{P}, \qquad \text{Precision@}k = \frac{\text{TP}_k}{k} \tag{9}$$

We also report recall by anomaly class, recall as a function of the review budget, and the number of reviewer labels needed by the active learning variant to reach a given recall at the 6 percent budget. For the label-efficiency study, we compare active learning with random sampling of the same number of records, with and without rule features, and average over three repetitions.

6. RESULTS

6.1 Overall performance

Table 4 reports the main results. Rules alone are precise, with a precision of 62.9 percent, but find only 38.6 percent of the anomalies, an F1 of 0.478. Their precision is limited, in this data, by the false positives from legitimate clinical extremes, about 1.5 percent of clean records, and their recall by the classes that they do not cover. The unsupervised isolation forest has a recall of 51.1 percent at the 6 percent budget and a precision of 47.4 percent, an F1 of 0.492, and an average precision of 0.559. It finds different anomalies from the rules: it detects batch drift and contextual anomalies that the rules miss, but it misses many of the clear range and unit errors that the rules catch with certainty.

Table 4. Detection at a 6 percent review budget on the test period (9198 records, 511 anomalies; rules shown at the 3.4 percent of records they flag).

Method	Reviewer labels	Precision (%)	Recall (%)	F1	Average precision
Rules only	None	62.9	38.6	0.478	-
Isolation forest (unsupervised)	None	47.4	51.1	0.492	0.559
GBM supervised (no rules)	Full	82.4	88.8	0.855	0.924
Hybrid: rules then isolation forest	None	64.4	69.5	0.669	0.590
Hybrid: weak supervision from rules	None (rule pseudo-labels)	56.1	60.5	0.582	0.497
Hybrid: GBM with rule features (full labels)	Full	86.2	93.0	0.895	0.953

Combining them in the simplest way, with the rule violations ranked first and the isolation forest filling the rest of the budget, raises recall to 69.5 percent and the F1 to 0.669, substantially better than either component, with no labels at all. The supervised GBM, trained on the labelled training records, does much better than the unsupervised combination, with a recall of 88.8 percent and an F1 of 0.855. This is expected, since it has the benefit of labels, but it is not the end of the story. Providing the GBM with the rule flags as additional features gives the best performance, a precision of 86.2 percent, a recall of 93.0 percent, an F1 of 0.895, and an average precision of 0.953. The improvement over the GBM without rules is modest in aggregate but is

concentrated where it matters, as the per-class results show.

The weakly supervised hybrid, which uses no reviewer labels, reaches a recall of 60.5 percent and an F1 of 0.582, below the simple cascade of rules and isolation forest. This is an instructive negative result. The pseudo-labels reproduce the coverage of the rule library: the classifier learns the rule-detected anomalies well, including the near-duplicates, which it finds more often than the rules do, with a recall of 62 percent against 52 percent, but it inherits the rules' blind spots for contextual and uncovered cross-field anomalies, and the noise from rule false positives. Weak supervision is therefore a way of extending rules to their neighbourhood, not of discovering new classes of anomaly.

6.2 Performance by anomaly class

Figure 2 shows the recall of each method for each class. The pattern is the main finding. Rules detect the range violations, unit errors, and covered cross-field violations essentially perfectly, with recall from 98 to 100 percent, and they detect half of the duplicates, the exact ones, and almost nothing else. The isolation forest detects 97 percent of the batch-drift records, which no rule can see, 66 percent of the uncovered cross-field anomalies, and 55 percent of the contextual anomalies, but only 36 percent of range violations, 53 percent of unit errors, and 29 percent of covered cross-field violations, because values that are extreme in one field are not always unusual in a multivariate sense once the other fields are considered. The supervised GBM without rule features detects all contextual and uncovered cross-field anomalies and 94 percent of duplicates, but only 75 percent of range violations and 74 percent of covered cross-field violations, a weakness that the rules do not have. With the rule features added, the GBM reaches 100 percent on the range, covered cross-field, uncovered cross-field, and contextual classes, 98 percent on unit errors, and 85 percent on temporal anomalies, the hardest class for every method, and retains 88 percent on batch drift and duplicates.

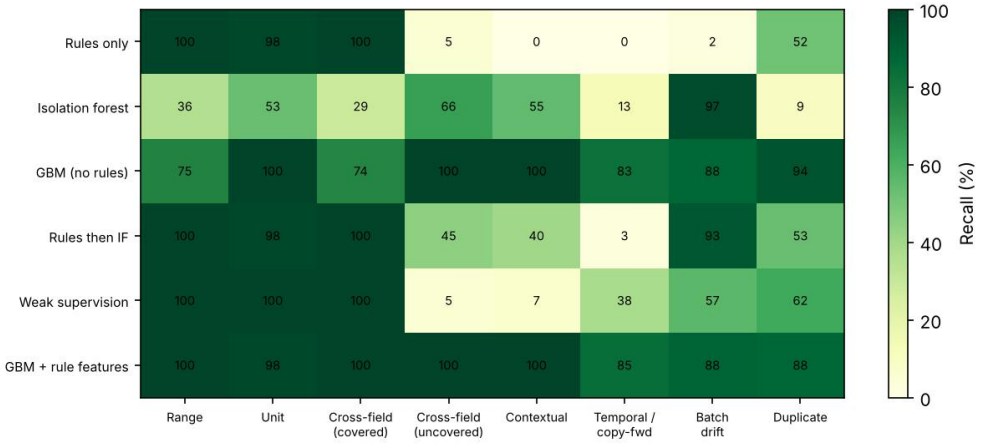

Figure 2. Recall at a 6% review budget by anomaly class and method. Rules are strongest where a rule exists; learning methods cover the gaps; the hybrid with rule features combines both.

These class-level results explain why the hybrid does better than either component and why the improvement is more than a matter of averages. The rule layer provides guaranteed coverage of the clear-cut violations, which is valuable because a failure to flag a heart rate of 450 is hard to defend, however good the model's average performance. The learning layer provides coverage of the contextual, temporal, aggregate, and uncovered cases that a rule library lacks. The temporal class, which contains copy-forward and implausible jumps, remains the hardest, with 85 percent recall in the best configuration and 3 percent or less in the unsupervised hybrid. The batch-drift class is detected by the unsupervised isolation forest, thanks to the site-week features, but only 88 percent by the supervised classifier, since two of the four drift batches fall in the test period and the model has seen only two in training.

6.3 Review budget

Figure 3 shows the recall of each method as a function of the review budget. At a 2 percent budget, which is smaller than the number of anomalies, no method finds more than 36 percent. By 4 percent the supervised methods have found 72 percent, against 49 percent for the unsupervised cascade of rules and isolation forest, 45 percent for the isolation forest, 44 percent

for the weakly supervised hybrid, and 39 percent for rules ranked alone. At a 10 percent budget the hybrid with rule features has found 96 percent of anomalies, the supervised model without rules 93 percent, the unsupervised cascade 78 percent, the weakly supervised hybrid 72 percent, and the isolation forest 62 percent, while rules plateau at about 42 percent because they cannot rank or extend beyond what they encode. A data steward who can inspect 6 percent of records finds 93 percent of the anomalies with the best hybrid, 89 percent with the supervised model alone, 70 percent with the unsupervised cascade, and 51 percent with the isolation forest alone.

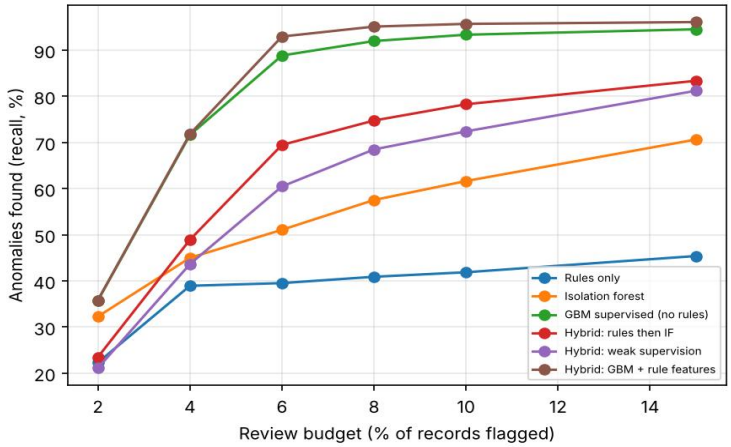

Figure 3. Share of anomalies found (recall) as a function of the review budget, for the ranked methods.

6.4 Label efficiency

Reviewer labels are costly, and Figure 4 compares how quickly different strategies convert labels into detection. With random sampling of labelled records and no rule features, a classifier trained on 400 labels finds 35 percent of anomalies at the 6 percent budget, and on 1600 labels 64 percent. Giving the classifier the rule features raises the latter to 77 percent. Active learning with the rule features does better still once it has enough labels to start: with 400 labels it finds 66 percent of anomalies, nearly twice the random-sampling figure, and with 800 and 1600 labels 76 and 84 percent. The advantage is large in the range of 400 to 800 labels. Active learning performs no better than random sampling, and a little worse, with 50 to 200 labels, which is the familiar cold-start problem: uncertainty sampling from a model trained on very few examples is not informative. The practical implication is to begin with a random or rule-guided seed set of a few hundred labels, for which the weak labels of the rule engine can help, and then to switch to uncertainty sampling.

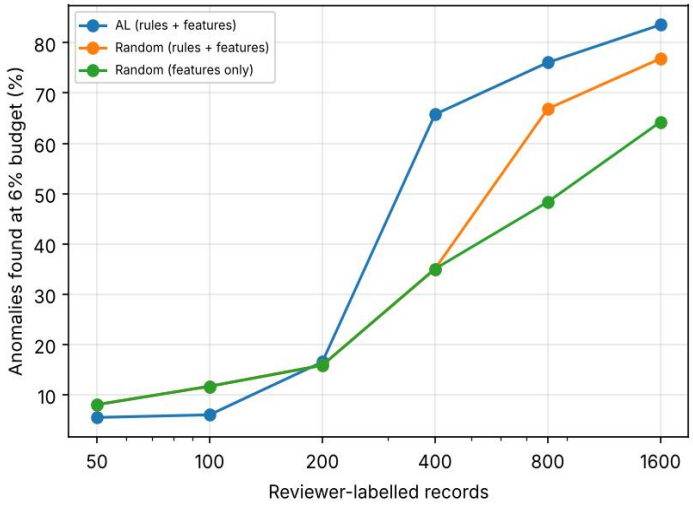

Figure 4. Anomalies found at a 6% review budget as a function of the number of reviewer-labelled records, for active learning with rule features, random sampling with rule features, and random sampling without rule features (mean of three repetitions).

6.5 Error analysis

The rules' false positives are legitimate clinical extremes. In the real world the proportion would depend on how tightly the ranges are set. Setting them wide reduces false positives and misses plausible errors, and setting them narrow does the opposite, and one of the benefits of the hybrid is that the learner can use patient context to discount rule hits that are probably real, such as a high potassium in a patient with kidney disease, so that the thresholds of the rules can be set conservatively without a high cost in precision. The learning methods' false negatives are concentrated in a few classes. Temporal anomalies in which a value is multiplied by a modest factor are difficult to tell from real changes between visits. Near-duplicates with small perturbations are found by the proximity feature, but when the perturbation is larger they become hard to distinguish from two genuine encounters. Batch drift of the size we simulated is detectable only when the site-week has enough records, and smaller shifts or sparser sites would be harder.

7. Deployment and Governance

The framework is designed to be deployable in a data warehouse or a clinical data repository, and several operational questions follow. Table 5 summarises recommendations. The first concerns the rule library. It should be treated as a managed asset, with an owner, version control, documented clinical rationale, severity, and expected false-positive rate for each rule, and with a process for retiring rules that produce alerts that are always dismissed. Rule hits are an unusually clean source of weak labels and of features, so that the library's quality benefits the learning layer as well as the first line of detection.

Table 5. Deployment recommendations.

Area	Recommendation	Why
Rule library	Owner, version control, rationale, severity, tracked false-positive rate	Rules feed detection, features, and weak labels
Budget and thresholds	Set review budget by field importance; calibrate on labelled samples; monitor flag rate	Base rate varies by source and time
Explanation	Reason codes, evidence (previous record, site-week aggregate), grouped alerts	Fast, defensible review
Feedback	Structured outcomes: confirmed error, legitimate value, uncertain	Labels improve the system
Safety	Flag only; corrections through governance with audit trail	Avoid silent loss of real extremes
Maintenance	Refresh supervised model on recent labels; time-split monitoring	New interfaces and anomaly types

The second concerns thresholds and budgets. The review budget is a policy decision that balances the cost of review against the cost of undetected errors, and it should be set separately for fields according to their clinical and analytic importance. Operating points should be calibrated on labelled samples, and, because the base rate of anomalies varies by source system and over time, they should be monitored. A sudden increase in the proportion of flagged records, in a particular site or interface, is itself a data-quality signal, often of an interface or calibration fault.

The third concerns explanation and workflow. Alerts should arrive with reason codes and evidence, so that a steward can decide quickly. For rule hits, the reason is the violated rule. For model hits, the reason is the features that contributed most, for example a deviation from the patient's previous value or from the expected value for the patient's conditions, which can be provided with feature-attribution methods. Alerts should be grouped, so that a batch drift appears as one issue and not as hundreds of records. The review outcome should be recorded in a structured way, confirmed error, legitimate value, or uncertain, with the corrected value where known, because these decisions are the labels that make the system improve.

The fourth concerns safety and scope. The framework flags candidates for review. It does not alter source data, and any correction should follow the organisation's data governance process, with an audit trail. Records flagged as probable errors could be excluded from analyses or model training, or down-weighted, but exclusion also removes real extreme values, which can bias results, and so the decision should be made deliberately and documented. The framework is not a substitute for clinical judgement on real-time decisions: a vital sign that is flagged as

implausible at the bedside should prompt a recheck, not a silent discard. Finally, privacy applies to the labelled data and the features, which are derived from patient records and must be handled under the same controls.

The fifth concerns drift of the system itself. The learning layer is trained on historical data, and new anomaly types, new interfaces, and changes in practice will change the data. The label-free components, the isolation forest and the batch features, continue to work on new types of anomaly, but the supervised classifier needs to be refreshed with recent labels, and its performance should be monitored on a continuing sample of reviewed records. The time-based evaluation of Section 5 is a model for this monitoring: performance is measured on recent data that the model has not seen.

8. Limitations

The principal limitation is that the evaluation is synthetic. The generator is ours, and so are the anomalies, which makes it possible that the learning methods exploit regularities of the generator that real data would not have. In particular, the injected anomalies are in several cases more separable than real ones: the contextual and uncovered cross-field anomalies are detected at or near 100 percent by the supervised classifiers, which would be unlikely in practice. The absolute figures should not be transferred to real data. What we expect to transfer is the qualitative picture: rules and learning have complementary strengths, rule features help the learner, weak supervision from rules extends coverage but does not discover new classes, and active learning pays off after a seed set of labels.

Other limitations should be noted. The rule library is small, twelve rules, and its incompleteness was designed, not observed. A real library is larger and more specific, and the gaps are not known in advance. The set of anomaly types is a subset of those that occur in practice, and omits, among others, missing-data anomalies, text and coding errors, wrong-patient errors, and errors in medication orders. The features rely on patient history, and so they are less informative for patients with few encounters. The comparison uses one model family, gradient-boosted trees, and one unsupervised method, the isolation forest, and other methods, such as autoencoders, might perform differently. The labels in the simulation are perfect, whereas real reviewers' labels are noisy and costly, and the active learning results do not model reviewer error or cost per label. We also do not model the effect of the framework on downstream analyses, which is the ultimate test of its value.

Finally, the distinction between an anomaly and an error remains, in real data, a matter of judgement, and some records flagged as anomalies will be legitimate. Our ground truth treats injected anomalies as errors and clinical extremes as normal, which simplifies a distinction that real reviewers sometimes cannot make.

9. Conclusion

Data quality problems in electronic health records are varied, and no single detection technology covers them. Rule-based checks are transparent and precise for the violations they encode but blind to what they do not encode, and learning-based detectors can find the unexpected but lack guarantees and, in their supervised form, need labels. We presented a hybrid framework in which rules serve as detectors, as features, and as weak labels, a learning layer covers their gaps, a fusion step ranks records under a review budget, and reviewer feedback improves the system.

In a controlled synthetic evaluation, the hybrid configurations outperformed rule-only and learning-only baselines at a fixed review budget. A supervised model given the rule flags found 93 percent of anomalies when 6 percent of records could be reviewed, against 39 percent for the rules alone and 51 percent for an isolation forest alone, and a simple label-free cascade of rules and an isolation forest found 70 percent. The class-level results showed why: rules guarantee coverage of the clear-cut violations and learning supplies the contextual, aggregate, and uncovered cases. Weak supervision from rules extended coverage to near-neighbours but not to new classes, and active learning with rule features roughly doubled the detection achieved by random labelling at 400 labels. These findings come from simulated data and should be confirmed on real EHR data, for which we have given the framework, the evaluation design, and governance guidance as a template. We recommend that organisations building EHR data-quality programmes treat rules and models as parts of one system, and design the feedback loop that lets each improve the other.

References

1. Pawan Kumar Singh, Scaling Gate-All-Around (GAA) Transistor Architectures for Sub-2nm AI Accelerators Using Atomically Thin 2D Materials, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5713
2. Pawan Kumar Singh, Energy-Efficient Task Scheduling in Green Cloud Computing Using Neuromorphic Spiking Neural Networks, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5712
3. A. Mahendra Vardhan and S. Sridhar, "Determining False Positive Analysis of Software Vulnerabilities with Predefined Scan Rules using Random Forest Classifier and Decision Tree Technique," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 622-625, DOI: https://doi.org/10.1109/ICAC3N56670.2022.10074458
4. Ronakkumar shah , Cognitive Process Intelligence: Integrating Generative AI and Automated Workflow Discovery, Vol. 33 No. 08 (2024): JOCAAA, Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5676
5. Ronakkumar shah, research Trends in Smart Workflow Automation: A Bibliometric Analysis of Emerging Themes, Home / Archives / Vol. 33 No. 05 (2024): JOCAAA, Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), https://eudoxuspress.com/index.php/pub/article/view/5675
6. Dip Bharatbhai Pate, Dynamic vs. Static visualizations: Understanding their use cases in big data analysis, International Journal of Science and Research Archive, 2022, 05(02), 392-395., DOI https://doi.org/10.30574/ijsra.2022.5.2.0077
7. Dip Bharatbhai Patel, Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021, DOI: https://doi.org/10.30574/ijsra.2021.2.1.0047
8. Dip Bharatbhai Patel, Understanding the Impact of Text Analytics on Social Media Sentiment Analysis, Vol. 5 No. 5 (2019) ,September-October, DOI: https://doi.org/10.32628/CSEIT1936413
9. Dip Bharatbhai Patel, Real-Time Data Streaming in BI Dashboards: Opportunities and Challenges, Archives / Vol. 5 No. 4 (2019) : July-August, DOI: https://doi.org/10.32628/CSEIT192142
10. Dip Bharatbhai Patel, the Role of Data Visualization in Modern Intelligence, Archives / Vol. 2 No. 2 / Research Article, Published, 2020-11-28 DOI: https://doi.org/10.32996/jcsts.2020.2.2.4
11. A. Srivastava, "Securing PHI in cloud-based healthcare systems: Challenges, frameworks, and future directions," J. Comput. Anal. Appl., vol. 33, no. 2, Aug. 2024. [Online]. Available: https://www.eudoxuspress.com/index.php/pub/article/view/4349/3190
12. A. Srivastava, "AI-assisted cloud migration of healthcare claims data from legacy systems: A metadata-driven framework," Int. J. Commun. Networks Inf. Secur., vol. 14, no. 3, Nov. 2022. [Online]. Available: https://ijcnis.org/index.php/ijcnis/article/view/8682
13. A. Srivastava, "Enhancing provider and claims data accuracy using artificial intelligence on cloud-native data platforms," J. Comput. Anal. Appl., vol. 31, no. 4, Nov. 2023. [Online]. Available: https://eudoxuspress.com/index.php/pub/article/view/4347/3189
14. A. Srivastava, "A unified framework for secure cloud data management: Integrating governance, metadata intelligence, and privacy controls," J. Inf. Syst. Eng. Manage., vol. 9, no. 2, Apr. 2024. [Online]. Available: https://www.jisem-journal.com/download/45_A_Unified_Framework.pdf
15. A. Srivastava, "Optimizing large-scale data migration for cloud-native architectures," J. Comput. Anal. Appl., vol. 31, no. 2, pp. 652–662, May 2023. [Online]. Available: https://eudoxuspress.com/index.php/pub/article/view/4603
16. Abdul Aziz. “Intelligent Health Analytics: Mastering AI in Healthcare Data”. Acta Scientiae, vol. 25, no. 4, Aug. 2024, pp. 544-55, https://www.periodicos.ulbra.org/index.php/acta/article/view/687. https://doi.org/10.5281/zenodo.21699886

17. Event-Aware Multi-Layer Storage Risk Forecasting for Oracle Database Estates Using HAPF, Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 10(4), 2024 ,10.22399/ijcesen.5183
18. Backup Integrity and Recovery Readiness Assessment for High-Availability Databases Raghu Gollapudi, Computer Fraud & Security, Vol. 2023, Issue 8, pp. 35-48, 10.52710/cfs.1031
19. Risk-Controlled Near-Zero-Downtime Oracle Database Migration Using GoldenGate Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 8(3), 2022 ,10.22399/ijcesen.5382
20. An Automated Risk Scoring Framework for SQL Execution Plan Analysis and Performance 56: Manoj Kumar Kagitha, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep),Publication year 2021, Volume 2021, Issue 12, https://computerfraudsecurity.com/index.php/journal/article/view/970
21. Manoj Kumar Kagitha, Hybrid Quantum-Classical Resource Scheduling: A Proposed Architecture for Next-Generation Hyperscale Data Centers, Publication year 2024, Volume 2024, Issue 12, https://computerfraudsecurity.com/index.php/journal/article/view/969
22. Manoj Kumar Kagitha, Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS),Publication year 2023, Volume 2023, Issue 12,https://computerfraudsecurity.com/index.php/journal/article/view/971
23. Manoj Kumar Kagitha, LOAD BALANCING TECHNIQUES IN DISTRIBUTED CLOUD ENVIRONMENTS, (IJESMR - 2017), D.O.I:10.53555/jcr.v6:i1.13380, https://jcreview.com/archives/volume-6/issue-1/13380
24. Manoj Kumar Kagitha, AI-DRIVEN OBSERVABILITY FOR HYPERSCALE COLOCATION DATA CENTERS: A CASE STUDY OF LOG ANALYTICS AND ANOMALY DETECTION PIPELINES, Archives / Vol. 27 No. 7 (2019): JOCAAA,https://www.eudoxuspress.com/index.php/pub/article/view/5053
25. Manoj Kumar Kagitha, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, https://www.eudoxuspress.com/index.php/pub/article/view/5054
26. Bandari, V. V., & Lekkala, H. B. (2024). AI-based operator behavior monitoring and cost optimization using digital traceability in manufacturing systems. Power System Protection and Control, 52(1), 38–45. https://pspac.info/index.php/dlbh/article/view/342
27. Venumadhav Vavilala, Shankar Balla (2024, May). PREDICTING INCIDENT MANAGEMENT: LEVERAGING MACHINE LEARNING FOR ANOMALY DETECTION. Power System Protection and Control Scopus Q1 Journal. PSPC. https://pspac.info/index.php/dlbh/article/view/270
28. Manikantha Varaprasad Inakollu, (2024, May), FINOPS-Driven Cloud optimization models for enterprise applications, Vol. 52 No. 2 (2024): April-June 2024, 99-110, Power System Protection and Control, ISSN-1674-3415, URL: https://pspac.info/index.php/dlbh/article/view/283,, DOI: https://doi.org/10.46121/pspc.52.2.10
29. Naresh Lokiny. (2022). Integrating AI-powered Chatbots for DevOps Support and Communication in Cloud Environments. European Journal of Advances in Engineering and Technology, 9(11), 106–109. https://doi.org/10.5281/zenodo.13325989
30. Naresh Lokiny, (2021), "Disaster Recovery and Business Continuity Planning in DevOps Cloud with AI", International Journal of Science and Research (IJSR), 10(3), 2024-2027. https://dx.doi.org/10.21275/SR24724151733, https://www.ijsr.net/getabstract.php?paperid=SR24724151733
31. Naresh Lokiny, & Ranganath Nandanampati. (2020). DevSecOps: Integrating Security into DevOps with AI in Cloud. Journal of Scientific and Engineering Research, 7(10), 239–242. https://doi.org/10.5281/zenodo.13348695
32. Naresh Lokiny, & Pradip Reddy. (2021). Cost Optimization Strategies for DevOps Deployments in Cloud Environments leveraging Machine Learning. European Journal of Advances in Engineering and Technology, 8(3), 69–72. https://doi.org/10.5281/zenodo.13325845
33. Jayanth Para, 6G Internet Technology Cyber Threat Notification & Alert System, Vol. 52 No. 4 (2024): October-December 2024, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/170 , DOI: https://doi.org/10.46121/pspc.52.4.9

34. Jayanth Para, AI Based Cloud Computation Observational Method & Process, Vol. 51 No. 4 (2023): October-December 2023, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/171 , DOI: https://doi.org/10.46121/pspc.51.4.3
35. Kaleshwar Aryasomayajula, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/294
36. Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/364
37. Kaleshwar Aryasomayajula, Micro services: "A Comparative Analysis of Monolithic vs. Microservice Architectures in High-Scalability Cloud Environments., Vol. 51 No. 2 (2023): April-June 2023 , Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/374
38. Vikas Suresh Bagora, KERNEL-LEVEL PERFORMANCE OPTIMIZATION FOR CARRIER-GRADE BROADBAND AND HIGH-SPEED NETWORKING SYSTEMS, Vol. 47 No. 1 (2019), Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/359
39. Vikas Suresh Bagora, SCALABLE 128G FIBRE CHANNEL AND ETHERNET CONVERGENCE FOR NEXT-GENERATION DATA CENTER NETWORKS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/362
40. Vikas Suresh Bagora, SECURE EMBEDDED NETWORKING ARCHITECTURES USING TRUSTED EXECUTION ENVIRONMENTS IN BROADBAND GATEWAYS, Vol. 52 No. 2 (2024): April-June 2024, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/363
41. Santhosh Chitraju Gopal Varma, The Evolution of Cloud-Native Architectures: Exploring the Synergy between Kubernetes and Microservices, DOI: https://doi.org/10.63282/3050-9246.IJETCSIT-V1I4P104, Vol. 1 No. 4 (2020), https://ijetcsit.org/index.php/ijetcsit/article/view/108
42. Santhosh Chitraju Gopal Varma, Artificial Intelligence in Cloud Computing: Building Intelligent, Distributed, and Fault-Tolerant Systems, Vol. 3 No. 1 (2022), DOI: https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P105, https://ijaibdcms.org/index.php/ijaibdcms/article/view/78
43. Sunil Karmakar, AI driven Identity & Access Management with Oracle Cloud security, Vol. 30 No. 2 (2022): JOCAAA, https://eudoxuspress.com/index.php/pub/article/view/4118
44. Rakshitha Kiran P., Samitha Khaiyum, Aparna R. Palandye, Apoorva S. D., Arjun R., Akshaya S. A., Annapoorna S., Leveraging LLaMA3 and LangChain for Rapid AI Application Development, Vol. 20 No. 10s (2024), https://doi.org/10.52783/jes.5539, https://journal.esrgroups.org/jes/article/view/5539
45. Kancharla Vijayalakshmi, Bethapudi Samuel Anand Andrews, Estimation of Cyproheptadine Hydrochloride and Tricholine Citrate Simultaneously in Syrup Dose by Applying Stability Indicating HPLC Methodology, December 31, 2020, Vol. 11, Issue 4, Pages 1813–1819, DOI: 10.26452/ijrps.v11iSPL4.4383, https://ijrps.com/v11si4/3683
46. Vijayalakshmi Kancharla ,Kumar Abbaraju V D N, Andrews B S A, Nagaraju Dasari, Nagu Korupolu, Melphan as Anticancer and New Insights of Drug as Parenteral Dosage form against SARS-CoV-2 by Using Development and Validation , November 18, 2020, Vol. 11, Issue 1, Pages 1437–1446
DOI: 10.26452/ijrps.v11iSPL1.3686, https://ijrps.com/v11si1/2911
47. GAURAV KUMAR SINHA, Efficient Test-Time Compute Scaling for Resource-Constrained Large Language Models Vol. 32 No. 1 (2024): JOCAAA, DOI: https://doi.org/10.1109/ICAC3N56670.2022.10074458 https://eudoxuspress.com/index.php/pub/article/view/5918

48. GAURAV KUMAR SINHA , The Impact of Artificial Intelligence on Predictive Quality Assurance in Modern Manufacturing, Vol. 33 No. 05 (2024): JOCAAA, https://doi.org/10.1109/ICAC3N56670.2022.10074458 , https://eudoxuspress.com/index.php/pub/article/view/5919
49. GAURAV KUMAR SINHA , Hybrid Quantum-Classical Neural Networks for Combinatorial Optimization and Machine Learning, Vol. 33 No. 08 (2024): JOCAAA, DOI: https://doi.org/10.1109/ICAC3N56670.2022.10074458, https://eudoxuspress.com/index.php/pub/article/view/5921
50. Khan, Irfan Ali, Yu Luo, Weifeng Xu, and Dianxiang Xu. "GNN-based transfer learning and tuning for detecting code vulnerabilities across different programming languages." In 2024 IEEE 24th International Conference on Software Quality, Reliability and Security (QRS), pp. 596-607. IEEE, 2024, DOI : 10.1109/QRS62785.2024.00065, https://ieeexplore.ieee.org/abstract/document/10684618