Kern Public : Pages:1–19

Original Research

Collaborative Anomaly Detection for Revenue Recovery Using Ensembles of Heterogeneous Data Agents

Pham Ngoc Hai[1] and Le Thi Bao Chau[2]

[1] Da Nang Institute of Technology and Management, Management and Computer Science Department, Nguyen Van Linh Street, Da Nang, Vietnam.
[2] Southern Mekong University of Applied Sciences, Management and Computer Science Department, Truong Chinh Road, Can Tho, Vietnam.

Abstract

Revenue management systems in digital platforms increasingly depend on fine-grained telemetry to ensure that contracted value is actually collected. Operational failures, instrumentation gaps, and adversarial behavior can introduce subtle discrepancies between reported and billable activity, giving rise to revenue leakage. Manual investigation and rule-based alarms often struggle to cover the breadth of heterogeneous data sources involved in modern billing pipelines. This paper examines collaborative anomaly detection for revenue recovery, in which multiple specialized data agents share signals to prioritize candidate losses. Each agent operates near a distinct telemetry stream, maintains its own detection model, and publishes anomaly scores that reflect local evidence of under-reported revenue. The central question is how to combine these partially overlapping, noisy views into actionable recommendations that align with downstream investigation capacity. The proposed framework models local anomaly scores as heterogeneous features, learns ensemble weights linked to historical recovery outcomes, and incorporates structural constraints derived from business rules. A linear decision layer maps aggregated scores to a ranking over candidate anomalies, while an optimization module selects subsets consistent with operational budgets. The study explores agent reliability modeling, cross-agent calibration, and robustness to missing or delayed signals in streaming settings. Empirical evaluation on synthetic and production-inspired datasets compares collaborative ensembles with isolated detectors and monolithic models trained on centralized logs. Under the examined scenarios, collaborative data agents allocate investigative effort toward events with higher estimated financial impact and expose anomalies that remain hidden to single-view methods.

1. Introduction

Revenue leakage arises when the value created by a product or service is not fully captured as realized income in accounting systems [1]. In digital businesses such as advertising platforms, subscription services, and marketplaces, the mapping from user activity to revenue involves multiple technical layers, including client instrumentation, network delivery, logging pipelines, attribution logic, and billing engines. Anomalies in any of these layers can break alignment between observed usage and expected charges, yet they may not manifest as easily observable defects. Small discrepancies in counters, skewed distributions for certain segments, or systematic under-reporting of specific event types can accumulate over time into nontrivial losses. Because these discrepancies are often entangled with normal variability in traffic and customer behavior, they are difficult to detect early with manual inspection alone [2].

Anomaly detection offers a way to surface unusual patterns in data streams that may signal such misalignments. Classical approaches rely on global thresholds on aggregated metrics, control charts, or univariate time series models. More recent work employs multivariate statistical methods and machine learning models to identify observations that deviate from learned baselines. While these techniques can be applied to revenue-related metrics, the complexity of modern telemetry makes it unlikely that a

single model, operating on a single view of the data, captures all relevant evidence [3]. Different systems observe different slices of the process, often with distinct definitions, sampling rates, and latency profiles. Errors can therefore appear as inconsistencies across views rather than as extreme values in any single metric.

Operationally, organizations respond to this complexity by creating specialized monitors close to particular data sources. Teams responsible for logging infrastructure, billing logic, or attribution signals maintain their own dashboards, alerts, and anomaly detection configurations. This specialization increases local sensitivity but does not automatically translate into a coherent, revenue-centric view [4]. Local anomalies may have no financial impact, while subtle cross-system inconsistencies with substantial monetary consequences might not trigger any single local alarm. Moreover, investigation resources such as analyst time and engineering capacity are limited, so it is not feasible to treat every alarm with equal priority. There is a need for mechanisms that coordinate heterogeneous detectors and direct attention toward events with higher expected revenue impact.

This paper examines a collaborative perspective in which each monitoring component is abstracted as a data agent [5]. A data agent is defined by the telemetry it can access, the transformations it performs, and the anomaly scores it emits. Agents may rely on distinct modeling techniques, ranging from simple statistical checks to complex learned models, and may operate at different granularities such as user, session, or invoice level. Rather than attempting to standardize all data into a single schema, the proposed view preserves local autonomy and expresses collaboration at the level of exchanged scores and low-dimensional summaries. An ensemble layer aggregates agent outputs into a global anomaly score that reflects an estimate of revenue risk for each candidate event or segment [6].

A central design objective is to link anomaly detection more directly to revenue recovery outcomes. To that end, the ensemble is not treated purely as an unsupervised aggregation of deviations. Instead, it is parameterized and trained using historical data in which a subset of anomalies has been investigated and labeled with estimated recovered revenue. This supervision remains partial and noisy, since only a fraction of issues are ever discovered and quantified, yet it provides a signal about which combinations of agent activations tend to correspond to financially meaningful incidents. The resulting model can prioritize anomalies whose joint signatures resemble past high-impact cases, even if individual agent scores are not extreme [7] [8].

Another consideration is the operational constraint that only a limited number of anomalies can be investigated over a given period. The detection system should therefore not only assign anomaly scores, but also propose a selection of candidates that respects these constraints and allows for diverse coverage. An optimization layer, formulated using linear models of expected recovery and capacity constraints, produces a set of recommended investigations. The overall system therefore comprises three conceptual layers: local detection within agents, collaborative ensembling that produces revenue-oriented anomaly scores, and decision optimization that maps scores to concrete actions [9].

The remainder of the paper develops this framing in more detail. It formalizes the revenue leakage setting and defines a mathematical representation of heterogeneous data agents. It then introduces a collaborative ensemble model that operates on agent-level anomaly scores and auxiliary descriptors, with an emphasis on linear formulations amenable to efficient optimization and interpretation. Revenue impact modeling and investigation selection are formulated as linear or convex optimization problems, enabling explicit treatment of operational budgets and risk tolerances [10]. Finally, the paper outlines an experimental design for assessing such systems under realistic constraints and discusses qualitative observations about the behavior of collaborative ensembles in this context.

2. Problem Formulation and Revenue Leakage Setting

The problem setting considered in this paper starts from a stream of revenue-bearing events. Each event corresponds to an atomic unit that, under ideal behavior of the system, would contribute a deterministic amount of revenue after the application of the business logic. For a digital advertising platform, an event may correspond to a delivered impression, a click, or a conversion attributed to a campaign. For a

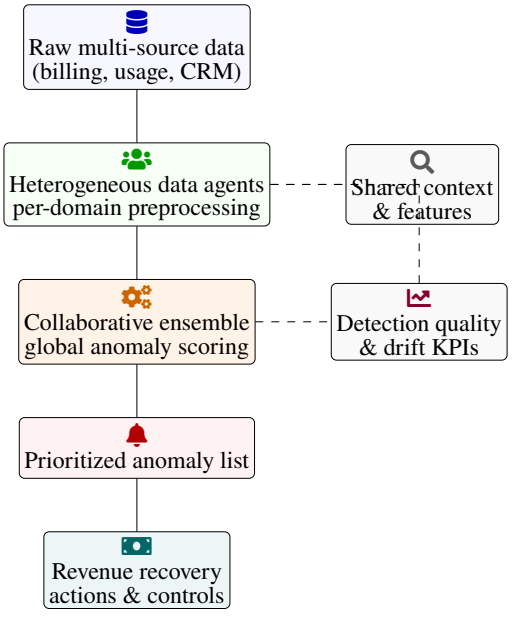

Figure 1: End-to-end collaborative anomaly detection architecture for revenue recovery. Heterogeneous data streams are transformed by domain-specific agents, combined into a collaborative ensemble for anomaly scoring, and consumed by a revenue recovery engine with monitoring and shared context.

Table 1: Summary of datasets used for collaborative anomaly detection.

Dataset	Domain	#Records (M)	#Features
Billing Logs	Postpaid	120	48
Network CDRs	Mobile Traffic	350	32
CRM Events	Customer Care	18	27
Web Analytics	Self-care Portal	95	22
Meter Readings	Fixed Access	62	19

Table 2: Heterogeneous data agents and their primary characteristics.

Agent Type	Data Source	Model Family	Output Granularity
Usage Agent	Network CDRs	Gradient Boosting	Session-level score
Billing Agent	Billing Logs	Temporal Autoencoder	Invoice-level score
Customer Agent	CRM Events	Sequence Classifier	Ticket-level score
Channel Agent	Web Analytics	Graph-based Model	Session-level score
Access Agent	Meter Readings	HMM + Clustering	Line-level score

subscription service, an event may be a billing cycle for a subscriber account [11]. For a marketplace, an event may be a completed transaction between buyer and seller. These events are identified by composite keys comprising attributes such as user identifiers, product identifiers, region, device type, or experiment assignment.

For each event index i, there is a notion of reference revenue r_i^* that would be realized if instrumentation, logging, and billing were perfectly aligned. In practice this reference value is latent. Observable signals include a billed amount r_i, telemetry counters from various systems, and derived features [12].

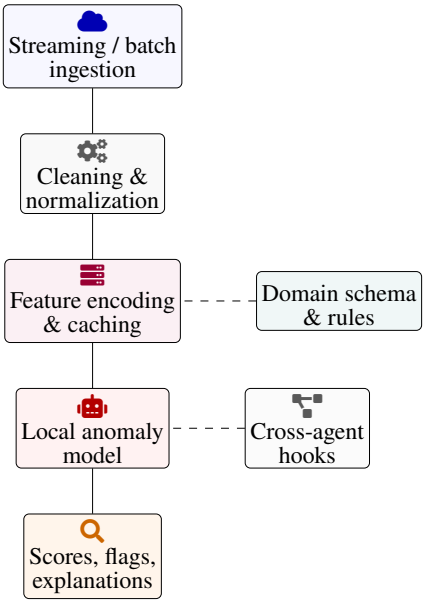

Figure 2: Internal structure of a heterogeneous data agent. Each agent handles ingestion, cleaning, feature encoding, and local anomaly modeling, exposing scores and explanations while integrating domain schemas and lightweight cross-agent hooks.

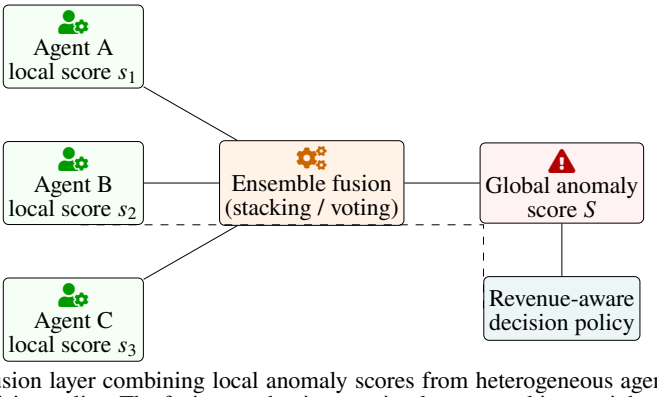

Figure 3: Ensemble fusion layer combining local anomaly scores from heterogeneous agents into a global score and revenue-aware decision policy. The fusion mechanism can implement stacking, weighted voting, or calibrated aggregation with feedback from downstream decisions.

Revenue leakage occurs when the accumulated discrepancy between reference and billed revenue over a set of events is positive. Let S denote a subset of events, and define the leakage over S as

$$L(S) = \sum_{i \in S} \left(r_i^* - r_i\right).$$

In reality r_i^* is unknown, so leakage is not directly observable. Instead, data teams rely on anomalies in related metrics as proxies. The aim of collaborative anomaly detection for revenue recovery is to identify subsets S that are likely to have positive leakage and for which investigation can lead to partial recovery through correction of configuration, data repair, or adjustments [13].

The volume of events is typically high, so analysis operates at aggregated granularities. Let each candidate item for anomaly detection be an index t representing a bucket of events grouped along keys

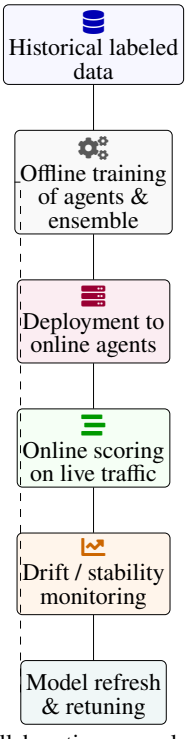

Figure 4: Training and deployment loop for collaborative anomaly detection. Historical data drive offline training across agents and the ensemble, with models deployed to online scoring, monitored for drift, and periodically refreshed in a closed loop.

Table 3: Primary anomaly categories relevant for revenue recovery.

Category	Example Pattern	Typical Impact
Leakage	Usage present in CDRs but missing in billing records	Direct revenue loss
Misconfiguration	Incorrect tariff mapping or tax application	Under-charging
Fraudulent Activity	SIM cloning, arbitrage, or bypass routes	High-margin loss
Process Breakdowns	Failed rating, mediation, or invoicing jobs	Delayed revenue
Data Quality Issues	Truncated sessions, duplicate events, inconsistent IDs	Hidden leakage

relevant for analysis, such as advertiser, campaign, region, or feature combinations. For bucket t, aggregated observable revenue is denoted R_t, and an unobserved reference revenue R_t^* would correspond to leakage $E_t = R_t^* - R_t$. The goal is to estimate, for each bucket t, a quantity related to the expected value of E_t and to select buckets for investigation according to their estimated contribution to total leakage and operational constraints.

Heterogeneous telemetry complicates this estimation. Distinct logging pipelines and systems produce multiple views of each bucket [14]. Some views may be closer to user behavior, while others are closer to billing logic. Denote by K the number of data agents, where each agent k is associated with one or more telemetry streams and a feature extraction process. For a given bucket t, agent k produces a feature vector $x_{k,t}$ in a space of dimension d_k. The union of all features across agents may be high dimensional and sparse, and many agents may not observe a given bucket due to filtering, sampling, or missing

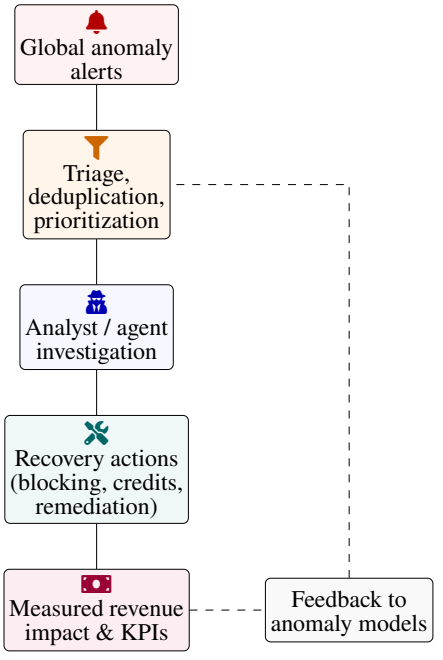

Figure 5: Revenue recovery workflow built on top of collaborative anomaly detection. Alerts are triaged and investigated, recovery actions are executed, and measured financial impact is fed back to tune thresholds and models.

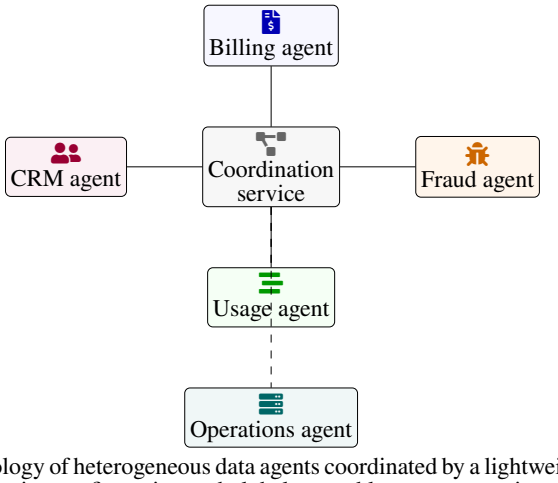

Figure 6: Collaboration topology of heterogeneous data agents coordinated by a lightweight service. Agents publish local anomaly signals and receive configuration and global ensemble parameters via sparse, low-latency channels.

data. Rather than centrally aggregating the raw $x_{k,t}$, each agent locally computes an anomaly score that summarizes its assessment of whether bucket t exhibits unusual behavior.

The anomaly score produced by agent k for bucket t is denoted $a_{k,t}$. This value can be interpreted in various ways depending on the underlying model: it may be a distance from a learned baseline, a negative log-likelihood, or a normalized deviation [15]. The present formulation treats it abstractly as a nonnegative scalar that increases with perceived abnormality. Agents may also provide auxiliary descriptors such as uncertainty estimates, data volume indicators, or labels for the type of anomaly they most suspect. These additional outputs are represented generically as vectors $z_{k,t}$ in moderate dimension.

Table 4: Ensemble architectures explored for collaborative detection.

Ensemble ID	Base Learners	Fusion Strategy	Collaboration Scope
E1	Per-agent isolation forests	Score averaging	Late fusion
E2	Heterogeneous GBMs + AEs	Stacking with meta-ML	Entity-level
E3	Agent-specific deep encoders	Attention-based fusion	Session-level
E4	Tree + sequence hybrids	Learned weighted voting	Customer-level

Table 5: Per-agent detection performance on labeled evaluation set.

Agent	Precision	Recall	F1-score
Usage Agent	0.81	0.63	0.71
Billing Agent	0.78	0.69	0.73
Customer Agent	0.72	0.58	0.64
Channel Agent	0.75	0.60	0.67
Access Agent	0.69	0.55	0.61
Global Oracle*	0.89	0.88	0.88

Table 6: Comparison of collaborative ensemble against baselines.

Method	AUC-PR	Recall@5% FP	Revenue Uplift (%)
Rule-based Controls	0.31	0.28	+0.0
Isolation Forest (global)	0.44	0.39	+4.7
Gradient Boosting (flat)	0.52	0.46	+7.9
Collaborative Ensemble (proposed)	0.68	0.61	+15.3
Ensemble w/o Cross-Agent Signals	0.57	0.49	+10.1

Table 7: Ablation study on collaboration mechanisms.

Variant	Cross-agent Messaging	Rel. F1 (%)	Rel. Revenue (%)
Full model	Enabled	100	100
No message passing	Disabled	92	86
Local-only calibration	Partial	95	90
Equal-weight score averaging	Enabled (static)	97	94
No uncertainty calibration	Enabled	93	88

The central challenge is that neither R_t^* nor E_t is known for the vast majority of buckets. Instead, a small subset of buckets is selected for manual or semi-automated investigation. For a subset T_{lab} of indices, investigation produces estimates of recovered revenue $\hat{E}_t$. These estimates are themselves noisy: recovery may be partial, some issues might be discovered long after the initial anomaly, and some anomalies may be confirmed as non-revenue affecting [16]. Nevertheless, the set $\{\hat{E}_t : t \in T_{\text{lab}}\}$ provides a training signal that links patterns in agent scores to financial outcomes.

From a detection perspective, the system must assign to each bucket t an ensemble anomaly score s_t intended to approximate the revenue relevance of that bucket. From a decision perspective, the system must select a subset T_{sel} of indices for investigation subject to resource constraints, such as a maximal number of tickets per day or a maximal volume of affected revenue segments per unit time. If C_t denotes

Table 8: Operational key performance indicators for deployment.

Metric	Definition	Value
Daily Volume	Scored entities per 24 hours	18M
Latency (p95)	End-to-end scoring latency	4.2s
Analyst Workload	Alerts per analyst per day	120
Precision@Top-100	Precision on top 100 alerts per day	0.91
Monthly Revenue Found	Confirmed recovered revenue per month	1.8M
False Positive Drop	Reduction vs. legacy rule-based system	37%

the cost of investigating bucket t and B is a budget, then the selection must satisfy a constraint of the form

$$\sum_{t \in T_{\text{sel}}} C_t \leq B.$$

The resulting selection determines which buckets enter an investigation pipeline that may involve engineers, analysts, and automated remediation tools.

An additional aspect of the problem is temporal evolution. Revenue systems are subject to changes in configuration, product launches, experiments, and external factors that affect user behavior [17]. Both normal patterns and anomaly patterns may shift over time. Data agents trained at one period may become miscalibrated, and ensemble weights derived from historical recovery outcomes may become less predictive. The system therefore needs mechanisms for incremental updating, retention of long-term structure, and control of drift. This temporal dimension interacts with the bounded investigation capacity, since delays in detection can reduce recoverable amounts [18].

In summary, the problem of collaborative anomaly detection for revenue recovery can be phrased as learning, from partially labeled historical data and streaming agent outputs, a mapping that ranks buckets by expected revenue leakage and selects a manageable subset for investigation at each decision point. The mapping must account for the heterogeneity of data agents, the sparsity and noise of labels, the operational constraints of investigation capacity, and the nonstationary nature of revenue systems.

3. Architecture of Heterogeneous Data Agents

A data agent in this context is a software component that maintains access to a particular subset of signals and is responsible for transforming them into anomaly scores and auxiliary descriptors. Agents may be colocated with logging infrastructure, billing services, ad serving systems, or downstream analytics stores [19]. The heterogeneity arises from differences in data schemas, time granularities, sampling strategies, and modeling capabilities across agents. The architecture aims to allow each agent to operate independently on its local data while still contributing to a shared detection process through standardized outputs.

Each agent k observes, for each relevant bucket t, a set of raw measurements. These may include counts, sums, ratios, and categorical breakdowns. The agent transforms these into a feature vector $x_{k,t}$ in a fixed-dimensional space. The transformation can incorporate domain knowledge encoded by the owning team, including normalization by traffic volume, seasonality adjustments, and filtering of low-support segments [20]. The design does not require global agreement on feature spaces, which is often difficult to achieve in large organizations. Instead, the only global contract is that agents expose a scalar anomaly score $a_{k,t}$ and optionally an auxiliary descriptor $z_{k,t}$ of modest dimension.

Internally, each agent maintains a detection model that maps from $x_{k,t}$ to a latent score before calibration. For example, an agent might fit a linear model for expected metric values and treat residuals as anomaly indicators. Another agent might maintain a probabilistic forecast and compute negative log-probabilities. Others might rely on density estimation or reconstruction errors [21]. The present

framework views all such variations through the unifying lens of an anomaly score $a_{k,t}$. Agents are free to retrain their internal models on local histories, adapt thresholds, and handle missing or delayed data according to their own constraints.

Despite local autonomy, some consistency across agents is useful. Raw anomaly scores may have different ranges and interpretations. One agent may output values that cluster near zero with occasional spikes, while another may produce scores in a narrow band between one and two [22]. In order to aggregate scores meaningfully, they must be calibrated to a common reference. A simple approach is for each agent to transform its internal score into a normalized statistic with approximately stable distribution under normal conditions. For example, an agent may map residuals to empirical quantiles so that under baseline behavior the distribution of $a_{k,t}$ is approximately uniform on an interval or standard normal. While perfect calibration is not required, such normalization encourages comparability and stabilizes training of ensemble models.

To express these ideas mathematically, consider agent k with internal feature vector $x_{k,t}$ and internal anomaly score $u_{k,t}$ produced by its chosen model. The agent applies a monotone calibration function c_k to produce the exported score

$$a_{k,t} = c_k(u_{k,t}).$$

The function c_k can be learned offline using local historical data or updated online using recent windows [23]. It may be nonparametric, for instance based on empirical cumulative distributions, or parametric with a small number of parameters. The key property is that higher values of $a_{k,t}$ correspond to stronger evidence of abnormality from the perspective of agent k.

Beyond scores, agents may publish reliability indicators. An agent operating on heavily sampled data may be less certain about its anomaly assessments for low-volume buckets. Another agent may experience intermittent data delays that affect timeliness [24]. Reliability can be exposed through a scalar weight $q_{k,t}$ in a bounded interval, interpreted as a soft confidence value. When agents emit both $a_{k,t}$ and $q_{k,t}$, the ensemble layer can account for variability in trust. For example, scores with low $q_{k,t}$ might be down-weighted or treated differently in learning. Reliability indicators also support monitoring of agents themselves by enabling meta-anomaly detection on their behavior.

Agents operate in environments where some buckets are not observable or are observed with delay. If agent k receives no data for bucket t, it can either abstain or emit a designated missing-score symbol. The ensemble must then incorporate variable dimensionality [25]. A practical strategy is to maintain, for each decision point, the set of active agents and treat missing scores as unobserved rather than imputed values. At training time, the model is exposed to the same patterns of missingness that occur at inference time, allowing it to implicitly learn which combinations of agents are most informative in which regions.

Communication between agents and the ensemble layer can be implemented through a message bus or data stream. Each agent writes, for each bucket t and decision period, a record containing identifiers, $a_{k,t}$, optional $z_{k,t}$, and optional $q_{k,t}$. The ensemble layer subscribes to these streams, joins messages across agents by bucket identifiers, and forms a multi-agent observation. This decoupled architecture avoids tight coupling between agent lifecycles and the central model [26]. Agents can be added, updated, or retired without structural changes to the ensemble, provided they adhere to the output interface. Versioning of agents and their calibration functions can be captured in metadata, enabling the ensemble to condition on agent versions to mitigate distribution shifts.

Conceptually, the architecture encourages viewing anomaly detection as a collaborative activity among specialized components rather than as a single monolithic detector. Each agent contributes a local perspective; the ensemble attempts to infer from their joint behavior which buckets warrant revenue-focused investigation [27]. This separation of concerns aligns with organizational structures in which different teams own different parts of the data and infrastructure but share responsibility for financial integrity. The mathematical formulation of the ensemble model, described in the next section, formalizes how agent outputs are combined and calibrated against historical recovery outcomes.

4. Collaborative Ensemble Anomaly Detection

The ensemble anomaly detection layer receives, for each bucket t, a set of agent scores and descriptors. Let the set of agents active for bucket t be denoted K_t [28]. For each $k \in K_t$, the ensemble observes a calibrated anomaly score $a_{k,t}$ and possibly auxiliary features $z_{k,t}$. The goal is to map these inputs to a scalar ensemble score s_t that reflects an estimate of revenue-related abnormality. A baseline approach would be to compute an unweighted average of $a_{k,t}$ across agents or a maximum. However, such simple rules ignore differences in agent reliability, differences in their relationships to revenue outcomes, and interactions among agent signals.

A more flexible approach is to embed the agent outputs into a vector representation and apply a trainable linear model that maps this vector to s_t. Consider a fixed set of K potential agents. For each bucket t, define a vector $v_t \in \mathbb{R}^K$ whose k-th component is a transformed version of the score from agent k. When an agent is inactive for a particular bucket, a neutral value such as zero can be used, with the understanding that the model learns how to interpret such cases [29]. Additional scalar features derived from auxiliary descriptors and bucket metadata can be concatenated to form a feature vector $f_t \in \mathbb{R}^d$. A simple linear ensemble model then takes the form

$$s_t = w^\top f_t + b,$$

where $w \in \mathbb{R}^d$ is a parameter vector and b is a scalar bias term. This formulation enables the inclusion of interactions, nonlinearity through basis expansions, and other transformations while retaining a linear parameterization.

Linking the ensemble score s_t to historical recovery outcomes involves a supervised learning problem. For buckets in the labeled set T_{lab}, observed recovered revenue $\hat{E}_t$ provides a target signal. Since $\hat{E}_t$ can take a wide range of values and may include zeros for non-revenue anomalies or false alarms, modeling it directly can be challenging. A common strategy is to model a transformed outcome such as an indicator of revenue-affecting anomaly or a monotone function of the recovered amount [30]. Let y_t denote such a transformed label for $t \in T_{\text{lab}}$. A regression model can then be trained to approximate y_t as a function of f_t.

A basic squared-loss formulation defines an empirical objective

$$J(w, b) = \sum_{t \in T_{\text{lab}}} \big(y_t - s_t\big)^2 + \lambda \|w\|_2^2, \text{[31]}$$

where λ is a regularization parameter controlling the magnitude of w. The regularization term mitigates overfitting to the limited set of labeled buckets and encourages smoother dependence on features. The minimizer of this convex function can be obtained in closed form or via standard optimization methods. This yields an ensemble model that linearly combines agent signals and descriptors in a manner tuned to historical labels.

In many settings, it is useful to account for heterogeneity across agents explicitly in the loss [32]. Some agents may cover high-value segments where historical recovery is more informative, while others operate on lower-value segments with noisy labels. Weighting errors differently for different buckets can reflect this heterogeneity. Let α_t be a nonnegative weight associated with bucket t, for instance proportional to its observed revenue R_t or estimated scale of potential leakage. The objective becomes [33]

$$J(w, b) = \sum_{t \in T_{\text{lab}}} \alpha_t \big(y_t - s_t\big)^2 + \lambda \|w\|_2^2.$$

This weighting emphasizes performance on buckets with higher revenue stake, aligning the ensemble more directly with the revenue recovery goal [34].

While linear models are relatively simple, they can be enhanced to incorporate structure in the agent graph. Suppose agents are nodes in a graph with edges representing similarity or shared ownership. It

may be desirable for the ensemble to treat similar agents in a consistent way. This can be expressed through a graph Laplacian regularizer. Let $L \in \mathbb{R}^{K \times K}$ be a positive semidefinite matrix encoding the agent graph and let $u \in \mathbb{R}^K$ be a subset of parameters corresponding to direct weights on agent scores. A smoothness penalty of the form [35]

$$\Omega(u) = u^\top L u$$

discourages large differences between weights of adjacent agents. Incorporating this term into the objective encourages collaborative behavior among related agents while still allowing specialized agents to deviate when justified by the data.

The ensemble can also be extended to multiple tasks corresponding to different slices of the business, such as markets, product lines, or time horizons. In a multi-task formulation, a separate parameter vector $w^{(m)}$ is learned for each task m, but these vectors share information through a low-rank structure. Let W be a matrix whose columns are the task-specific parameters [36]. A low-rank factorization $W = UV^\top$ with small latent dimension captures shared patterns. Linear models with such factorizations retain a linear structure in the features while introducing a structured parameter matrix. Optimization can proceed by alternating updates of U and V, or by directly penalizing the nuclear norm of W to encourage low rank.

In online or streaming settings, the ensemble parameters must be updated incrementally as new labels arrive. Suppose that at each update step a small batch of newly investigated buckets with labels $\{(f_t, y_t)\}$ becomes available. An incremental gradient step on the squared-loss objective leads to a parameter update of the form [37]

$$w_{n+1} = w_n - \eta \sum_t (s_t - y_t) f_t,$$

where η is a learning rate. Regularization can be integrated by shrinking w_n toward zero between updates. This simple linear update rule has low computational cost and allows the ensemble to adapt gradually to changes in agent behavior and business conditions. Careful selection of learning rates and regularization strength is required to balance stability with responsiveness [38].

The ensemble score s_t provides a ranking over buckets. In many applications, the absolute values of s_t are less important than their order, since investigation capacity is limited and only the top-ranked entries will be processed. Pairwise ranking losses or margin-based formulations can be adopted to better align the model with ranking performance metrics. However, the linear structure remains central [39]. For example, a pairwise hinge loss between buckets i and j with labels indicating $y_i > y_j$ depends on differences $w^\top (f_i - f_j)$, preserving linear dependence on the parameters. Such formulations can be optimized using stochastic gradient methods based on sampled pairs.

Overall, the collaborative ensemble anomaly detection model combines the expressive power of heterogeneous data agents with the tractability of linear modeling. It provides a mechanism for learning how to interpret combinations of agent scores in light of historical revenue recovery outcomes, while remaining suitable for high-volume, streaming environments due to its computational efficiency.

5. Revenue Recovery Modeling and Decision Policy

An ensemble anomaly score provides a ranking over buckets but does not, by itself, determine which anomalies should be investigated under resource constraints [40]. To connect detection to action, a revenue recovery model is introduced that maps ensemble scores and auxiliary features into estimates of expected recovered revenue and uses these estimates in an optimization problem. The resulting decision policy specifies, at each decision period, a set of buckets to route to investigation.

Let s_t denote the ensemble anomaly score for bucket t at a given decision time. Additional observable quantities include the observed revenue R_t, volume indicators, and agent-level summaries. The objective is to estimate the conditional expectation of recoverable revenue given these quantities [41]. Denote this

expectation by μ_t. A simple parametric model assumes that μ_t depends linearly on a feature vector g_t built from s_t and other descriptors. Specifically,

$$\mu_t = \beta^\top g_t,$$

where β is a parameter vector learned from historical data where estimates $\hat{E}_t$ are available. In contrast to the ensemble model, which may be trained on broader labels, this layer focuses on continuous estimates of recovery magnitude [42]. A squared-loss objective similar to that used for the ensemble can be employed, possibly with different regularization reflecting the scale and uncertainty of $\hat{E}_t$.

The investigation decision problem can be phrased as a constrained optimization. Consider a decision period in which a set $\mathcal{T}$ of buckets is eligible for investigation. For each $t \in \mathcal{T}$, the system has an estimated recoverable amount μ_t and a cost of investigation C_t. The cost may be measured in units of analyst time, engineering effort, or operational overhead. A total budget B captures how much cost can be incurred during the period. The decision variables are binary indicators x_t representing whether bucket t is selected for investigation [43]. A natural optimization model maximizes estimated recovered revenue under the budget constraint:

$$\max_{x} \sum_{t \in \mathcal{T}} \mu_t x_t$$

subject to

$$\sum_{t \in \mathcal{T}} C_t x_t \le B,$$

$$x_t \in \{0, 1\} \quad \text{for all } t.$$

This is a 0-1 knapsack problem [44]. For large numbers of buckets, exact solution may be computationally demanding, but a greedy heuristic that selects buckets sorted by the ratio μ_t / C_t yields a solution that is often adequate in practice. When all C_t are equal, the problem reduces to selecting the top K buckets by μ_t for a given K.

The basic optimization can be extended to incorporate additional operational constraints. For example, investigation teams may require diversity across markets, product lines, or anomaly types to avoid over-concentration on a single area [45]. This can be encoded through linear constraints that limit the number of selected buckets per category. Suppose categories are indexed by c and I_c denotes the set of buckets in category c. A constraint of the form

$$\sum_{t \in I_c} x_t \le B_c$$

limits selection from each category to at most B_c items [46]. Such constraints preserve linearity and can be integrated into the knapsack formulation, leading to a multi-dimensional knapsack problem. Approximate algorithms and relaxations can still be applied.

Another consideration is risk management. Estimates μ_t are uncertain; some may be overestimates while others may underestimate the true recoverable amount. A risk-averse policy might prefer a more conservative objective that discounts uncertain gains [47]. If σ_t represents an estimate of the uncertainty or variance associated with μ_t, a risk-adjusted value can be defined as

$$\tilde{\mu}_t = \mu_t - \gamma \sigma_t,$$

where γ is a nonnegative coefficient reflecting risk aversion. The optimization then maximizes the sum of $\tilde{\mu}_t x_t$. This penalizes buckets with high uncertainty, favoring those with more reliable estimates. Alternatively, robust optimization techniques can be used, modeling μ_t as belonging to an uncertainty set and optimizing the worst-case recovered revenue over this set [48].

On longer time horizons, feedback from investigations can be used to update both μ_t models and decision policies. Suppose that, over multiple periods, realized recoveries E_t are observed for selected buckets. Comparing these with prior estimates μ_t allows calibration of predictive models. One can define residuals [49]

$$\delta_t = E_t - \mu_t$$

and fit a correction model that adjusts future estimates. Linear models for δ_t as a function of features may capture systematic biases, such as consistent underestimation in certain markets. Incorporating such corrections incrementally tightens the alignment between estimated and realized recovery.

The decision policy itself can be adapted based on observed outcomes [50]. For instance, if investigation capacity is consistently under-utilized due to fewer anomalies with nontrivial recovery, the budget parameter B or selection thresholds can be adjusted. Conversely, if teams are overloaded, budget parameters can be reduced or more stringent selection criteria applied. In some settings, the policy may aim not only to maximize expected recovered revenue but also to explore regions of the space where the recovery model is highly uncertain, to improve future estimates. This can be formalized through exploration terms that allocate a fraction of capacity to buckets with high uncertainty σ_t, even if their current μ_t is moderate [51].

Overall, the revenue recovery modeling and decision policy layer translates anomaly scores into concrete, resource-constrained actions. By formulating the problem using linear models for expected recovery and linear constraints for operational budgets, the system maintains interpretability and tractability while enabling explicit reasoning about trade-offs between potential revenue gain and investigation cost.

6. Experimental Design and Analysis

To assess the behavior of collaborative anomaly detection with heterogeneous data agents, an experimental design must reflect both the structural complexity of revenue systems and the constraints of operational environments. While implementations and datasets vary across organizations, a generic evaluation approach can be described that highlights key aspects of performance [52]. The emphasis here is on the design of experiments rather than on specific numerical results.

A practical evaluation combines synthetic data, which allows controlled injections of anomalies, with production-inspired data that captures realistic variability. Synthetic experiments begin by simulating a baseline revenue process. Buckets indexed by t are assigned features drawn from distributions that mimic segment attributes such as geography, device type, or campaign identifiers. A baseline mapping from features to reference revenue R_t^* is specified using a linear or piecewise linear function with noise. Multiple telemetry streams are then derived from the same latent process, each with its own measurement noise, sampling, and potential biases [53]. For example, one stream might undercount certain events in specific segments, while another might exhibit delayed logging. These synthetic telemetry streams feed simulated data agents, which compute anomaly scores based on deviations from their learned baselines.

Anomalies representing revenue leakage are injected by perturbing the mapping from features to observed revenue R_t or by introducing discrepancies between telemetry streams. For instance, a systematic undercounting of impressions in a particular region may reduce R_t relative to R_t^* without affecting some telemetry streams. A misconfiguration in attribution logic may lead to missing conversions for subsets of campaigns [54]. These perturbations are configured to span a range of magnitudes and durations, from short-term spikes to longer-running drifts. The ground truth leakage $E_t = R_t^* - R_t$ is recorded for evaluation but not exposed to agents or ensembles.

Data agents in the synthetic setup are implemented using models consistent with their intended capabilities. One agent may track aggregate ratios and apply univariate control limits, producing anomaly scores when ratios deviate beyond limits. Another may maintain multivariate Gaussian models of metrics per segment [55]. A third may employ time-series forecasting for daily revenue and compute prediction

errors. Each agent only sees its own telemetry stream and does not have direct access to the latent reference revenue. Calibration functions c_k are fitted using pre-anomaly data windows, ensuring that agent scores under baseline behavior follow approximately stable distributions.

The collaborative ensemble model is trained using a subset of buckets labeled with estimated recovered revenue. In synthetic experiments, these labels can be derived directly from ground truth leakage, possibly with added noise to mimic imperfect investigations [56]. The labeled subset is sampled according to realistic investigation policies, for example by selecting high-scoring buckets under non-collaborative detectors or random sampling. This sampling process introduces selection bias, since labels are more likely to be available for extreme anomalies. Training procedures that ignore this bias may overestimate performance on such buckets. To study robustness, experiments can vary the labeling policy and examine how ensemble performance changes [57].

Performance metrics for collaborative anomaly detection in this context must capture both detection quality and revenue relevance. Classical metrics such as precision, recall, and area under the receiver operating characteristic curve can be computed with respect to thresholds on estimated leakage. However, because investigation capacity is limited, more relevant metrics focus on top segments. One such metric is precision at K, defined as the fraction of the top K selected buckets that correspond to true revenue-affecting anomalies [58]. Another is cumulative recovered revenue at K, computed as the sum of ground truth leakage over the top K selections. Comparing these metrics between collaborative ensembles, individual agents, and monolithic models trained on merged telemetry provides insight into the benefits and trade-offs of the collaborative architecture.

An important aspect of analysis concerns robustness to missing or degraded agents. In practice, some telemetry streams may fail or become unavailable due to system outages or schema changes. Experiments can simulate such scenarios by dropping one or more agents for segments of the data and observing the impact on ensemble performance [59]. The linear ensemble formulation naturally handles missing inputs by treating absent agent scores as zeros or by masking. Evaluation can quantify how gracefully performance degrades when key agents are removed and whether the ensemble redistributes weight to remaining agents during retraining. Sensitivity to individual agents can be analyzed by computing marginal contributions of each agent to ensemble scores across buckets.

Another focus of experimental analysis is calibration of ensemble scores with respect to revenue outcomes [60]. Well-calibrated scores should align with empirical probabilities or expected magnitudes of leakage. Calibration can be evaluated by grouping buckets into bins by predicted score and comparing average ground truth leakage within each bin. Deviations indicate overconfidence or underconfidence. Linear calibration layers or isotonic regression can be applied on top of ensemble scores to improve alignment [61]. In dynamic environments, calibration may drift over time, necessitating periodic adjustment. Experiments that simulate regime shifts, such as changes in traffic patterns or product configuration, can reveal how quickly the ensemble and calibration mechanisms adapt.

Production-inspired experiments rely on anonymized logs from operational systems, where true revenue leakage is only partially observed. In such datasets, labels from historical investigations provide a sparse and biased sample. To evaluate detection quality on unlabelled segments, proxy signals can be employed [62]. For example, anomalies confirmed by downstream reconciliation processes or adjustments in billing records can be used as partial ground truth. While these proxies do not capture all leakage, they offer additional validation channels. Analysis on these datasets emphasizes relative comparisons between collaborative ensembles and baselines rather than absolute measures of missed leakage.

Throughout experimental analysis, it is useful to investigate interpretability [63]. Linear models facilitate inspection of learned weights on agent scores and features. By examining the magnitude and sign of weights associated with each agent, analysts can infer which agents contribute most to detecting certain classes of anomalies. Combining this with case studies of specific incidents allows teams to reason about why particular buckets were prioritized for investigation. Interpretability can also highlight misalignments, such as an agent whose scores systematically downweight segments that historically yielded

meaningful recovery [64]. Identifying such patterns can inform adjustments to agent models or data collection processes.

In summary, the experimental design aims to characterise how ensembles of heterogeneous data agents behave in detecting revenue-related anomalies, with emphasis on revenue-oriented metrics, robustness to missing agents, calibration over time, and interpretability of learned weights and decision policies. While details of implementation and numerical results depend on the specific environment, the outlined framework provides a basis for systematic evaluation.

7. Conclusion

This paper has examined collaborative anomaly detection for revenue recovery in environments where multiple heterogeneous data agents monitor different aspects of revenue-generating processes. Starting from a formulation that links latent reference revenue, observed revenue, and leakage, the discussion introduced data agents as components that transform local telemetry into calibrated anomaly scores and auxiliary descriptors [65]. A linear ensemble model was proposed to combine these agent outputs into revenue-oriented anomaly scores, using partially labeled historical recovery outcomes as supervision. The ensemble design accounted for heterogeneity in agent reliability and potential structure in an agent graph through regularization.

To connect detection to operational action, a revenue recovery modeling layer was introduced. This layer mapped ensemble scores and additional features into estimates of expected recovered revenue and posed the investigation selection problem as a constrained optimization [66]. Linear models for expected recovery and linear capacity constraints allowed the selection problem to be expressed as a variant of the knapsack problem, with possible extensions for diversity and risk management. Incremental updates based on realized recoveries provided a mechanism for refining both predictive models and decision policies over time.

The architectural perspective emphasized decoupling of local detection, collaborative ensembling, and decision optimization. Agents retained autonomy over their internal models and feature engineering, while a common interface for anomaly scores and descriptors enabled shared learning about revenue impacts [67]. The ensemble, being linear in its parameters, supported scalable training and online adaptation as new labels became available. This structure also facilitated interpretability by allowing inspection of learned weights and their evolution.

Experimental considerations highlighted the importance of combining synthetic and production-inspired data to evaluate such systems. Synthetic experiments can systematically vary anomaly types, magnitudes, and agent failures to probe robustness, while production-like data captures real-world variability and partial labeling. Revenue-oriented metrics focusing on top-ranked anomalies, cumulative recovered amounts, and calibration provide a more direct assessment of utility for revenue protection than generic detection metrics alone [68]. Analysis of sensitivity to missing agents and the stability of calibration under regime shifts further informs deployment decisions.

Several limitations and potential extensions arise from this formulation. The linear models discussed provide a tractable starting point but may not fully capture complex interactions among agent signals and contextual features. More expressive models, possibly combined with constraints or regularization encouraging interpretability, could be explored [69]. Label sparsity and selection bias remain challenging, suggesting the value of methods that can leverage weak supervision, semi-supervised learning, or causal modeling to better relate observed anomalies to revenue outcomes. Finally, integrating such systems into organizational processes requires careful consideration of feedback loops, governance, and communication between technical and business stakeholders.

Despite these limitations, the framework presented offers a structured way to think about collaborative anomaly detection for revenue recovery. By treating anomaly detection as a coordinated activity among heterogeneous data agents and explicitly linking detection to revenue impact and operational constraints, it provides a basis for designing systems that align monitoring capabilities with financial objectives in complex data environments. and organizations [70].

References

[1] Q. Li, H. Chu, M. Zhang, M. Li, and L. Diao, "Collaboration strategy for software dynamic evolution of multi-agent system," *Journal of Central South University*, vol. 22, pp. 2629–2637, 7 2015.

[2] M. S. Güzel and H. Kayakökü, *A Collective Behaviour Framework for Multi-agent Systems*, pp. 61–71. United States: Springer International Publishing, 8 2016.

[3] J. Ferber, T. Stratulat, and J. Tranier, *Towards an Integral Approach of Organizations in Multi-Agent Systems*. IGI Global, 1 2011.

[4] R. Chandrasekar, V. Vijaykumar, and T. Srinivasan, "Probabilistic ant based clustering for distributed databases," in *2006 3rd International IEEE Conference Intelligent Systems*, pp. 538–545, IEEE, 2006.

[5] I. Mordatch and P. Abbeel, "Emergence of grounded compositional language in multi-agent populations," 1 2017.

[6] Y. Guan, Z. Ji, L. Zhang, and L. Wang, "Controllability of multi-agent systems under directed topology," *International Journal of Robust and Nonlinear Control*, vol. 27, pp. 4333–4347, 3 2017.

[7] T. Zhang, *Constitution of Virtual Service System Based on Multi-agent*, pp. 855–860. Springer Berlin Heidelberg, 4 2015.

[8] K. S. HEGDE, "Recovering lost revenue by augmenting internal customer data with external data for accurate invoicing for large b2b enterprises," *INTERNATIONAL JOURNAL*, vol. 13, no. 11, pp. 613–615, 2024.

[9] J. Jin, R. T. Maheswaran, R. Sanchez, and P. Szekely, "Iui - vizscript: visualizing complex interactions in multi-agent systems," in *Proceedings of the 12th international conference on Intelligent user interfaces*, pp. 369–372, ACM, 1 2007.

[10] A. Helleboogh, D. Weyns, and T. Holvoet, *On the Role of Software Architecture for Simulating Multi-Agent Systems*, pp. 167–214. CRC Press, 6 2009.

[11] Y. Wang, R. K. Dubey, N. Magnenat-Thalmann, and D. Thalmann, "An immersive multi-agent system for interactive applications," *The Visual Computer*, vol. 29, pp. 323–332, 6 2012.

[12] D. Martínez, E. Clotet, J. Moreno, M. Tresanchez, and J. Palacín, *PAAMS (Special Sessions) - A Proposal of a Multi-agent System Implementation for the Control of an Assistant Personal Robot*, pp. 171–179. Springer International Publishing, 6 2016.

[13] S. Vester, N. S. Boss, A. S. Jensen, and J. Villadsen, "Improving multi-agent systems using jason," *Annals of Mathematics and Artificial Intelligence*, vol. 61, pp. 297–307, 3 2011.

[14] O. Achbarou, M. A. E. Kiram, and S. Elbouanani, "Cloud security: A multi agent approach based intrusion detection system," *Indian Journal of Science and Technology*, vol. 10, pp. 1–6, 5 2017.

[15] R. Chandrasekar and S. Misra, "Introducing an aco based paradigm for detecting wildfires using wireless sensor networks," in *2006 International Symposium on Ad Hoc and Ubiquitous Computing*, pp. 112–117, IEEE, 2006.

[16] J. Valk, M. de Weerdt, and C. Witteveen, *Coordination in Multi-Agent Planning with an Application in Logistics*. IGI Global, 1 2011.

[17] J. Wang, Z. Jiang, Y. Deng, and C. Jiang, "Research on diffusion of warning based on multi-agent modeling," *Advanced Materials Research*, vol. 734-737, pp. 3259–3263, 8 2013.

[18] M. G. Buhnerkempe, M. G. Roberts, A. P. Dobson, H. Heesterbeek, P. J. Hudson, and J. O. Lloyd-Smith, "Eight challenges in modelling disease ecology in multi-host, multi-agent systems," *Epidemics*, vol. 10, pp. 26–30, 12 2014.

[19] A. Windham and S. Treado, "A review of multi-agent systems concepts and research related to building hvac control," *Science and Technology for the Built Environment*, vol. 22, pp. 50–66, 10 2015.

[20] B. Zhao, Y. Guan, and L. Wang, "Non-fragility of multi-agent controllability," *Science China Information Sciences*, vol. 61, pp. 052202–, 12 2017.

[21] D. Weyns, A. Helleboogh, T. Holvoet, and M. Schumacher, "The agent environment in multi-agent systems: A middleware perspective," *Multiagent and Grid Systems: An International Journal of Data Science and Artificial Intelligence*, vol. 5, pp. 93–108, 2 2009.

[22] C. Zhou, B. Sun, and Z. Liu, "Abstraction for model checking multi-agent systems," *Frontiers of Computer Science in China*, vol. 5, pp. 14–25, 12 2010.

[23] P.-M. Ricordel and Y. Demazeau, *CEEMAS - Volcano, a Vowels-Oriented Multi-agent Platform*. Germany: Springer Berlin Heidelberg, 3 2002.

[24] Q. Rui and S. Q. Qian, "Research on realization of over-current protection based on multi-agent strategy," *Applied Mechanics and Materials*, vol. 602-605, pp. 3009–3012, 8 2014.

[25] V. Vijaykumar, R. Chandrasekar, and T. Srinivasan, "An ant odor analysis approach to the ant colony optimization algorithm for data-aggregation in wireless sensor networks," in *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, IEEE, 2006.

[26] T. Chen, F. Song, and Z. Wu, "Global model checking on pushdown multi-agent systems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 3 2016.

[27] P. Blikstein, W. Rand, and U. Wilensky, "Aamas - participatory, embodied, multi-agent simulation," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 1457–1458, ACM, 5 2006.

[28] Y. Zhu, Y. Zheng, and L. Wang, "Quantised consensus of multi-agent systems with nonlinear dynamics," *International Journal of Systems Science*, vol. 46, pp. 2061–2071, 10 2013.

[29] Z. Ye, Y. Chen, and H. Zhang, "Distributed consensus of delayed multi-agent systems with nonlinear dynamics via intermittent control," *Asian Journal of Control*, vol. 18, pp. 964–975, 6 2015.

[30] A. Kilic and A. Arslan, "Advis - minimax fuzzy q-learning in cooperative multi-agent systems," *Lecture Notes in Computer Science*, pp. 264–272, 10 2002.

[31] P. Argoneto and P. Renna, *Multi-Agent Architecture*, pp. 31–47. Springer London, 6 2011.

[32] L.-S. Wang and Z.-H. Wu, "A novel model and behavior analysis for a swarm of multi-agent systems with finite velocity," *Chinese Physics B*, vol. 23, pp. 098901–, 9 2014.

[33] G. Mitra and S. Bandyopadhyay, *A Study on Some Aspects of Biologically Inspired Multi-agent Systems*, pp. 207–217. Springer Singapore, 12 2017.

[34] L. Zhang, Z. Qi, Q. Z. Wang, X. P. Wang, and X. Shen, "Building a multi-agent system for emergency logistics collaborative decision," *Applied Mechanics and Materials*, vol. 513-517, pp. 2041–2044, 2 2014.

[35] T. Salamon, *ISD - A Three-Layer Approach to Testing of Multi-agent Systems*. Springer US, 8 2009.

[36] R. Chandrasekar, R. Suresh, and S. Ponnambalam, "Evaluating an obstacle avoidance strategy to ant colony optimization algorithm for classification in event logs," in *2006 International Conference on Advanced Computing and Communications*, pp. 628–629, IEEE, 2006.

[37] H. J. Chang, "Integration of blackboard architecture into multi-agent architecture," *Journal of the Korea Academia-Industrial cooperation Society*, vol. 13, pp. 355–363, 1 2012.

[38] W. S. Duan, Y. Ma, L. P. Liu, and T. P. Dong, *Research on an Intelligent Distance Education System Based on Multi-agent*, pp. 579–586. Germany: Springer London, 2 2013.

[39] G. A. Boy, *Cognitive Function Analysis in the Design of Human and Machine Multi-Agent Systems*, pp. 189–206. CRC Press, 11 2017.

[40] Y. Jiang, J. Liu, and S. Wang, "A consensus-based multi-agent approach for estimation in robust fault detection.," *ISA transactions*, vol. 53, pp. 1562–1568, 6 2014.

[41] D. Li, J. Ma, H. Zhu, and M. Sun, "The consensus of multi-agent systems with uncertainties and randomly occurring nonlinearities via impulsive control," *International Journal of Control, Automation and Systems*, vol. 14, pp. 1005–1011, 6 2016.

[42] S. A. DeLoach and M. Kumar, *Multi-Agent Systems Engineering*. IGI Global, 1 2011.

[43] E. Lavendelis and J. Grundspenkis, "Requirements analysis of multi-agent based intelligent tutoring systems," *Scientific Journal of Riga Technical University. Computer Sciences*, vol. 38, pp. 37–47, 1 2009.

[44] E. Nawarecki, M. Kisiel-Dorohinicki, and G. Dobrowolski, *CEEMAS - Organisations in the Particular Class of Multi-agent Systems*. Germany: Springer Berlin Heidelberg, 3 2002.

[45] Y. Hu, P. Li, and J. Lam, "Brief paper - consensus of multi-agent systems: a simultaneous stabilisation approach," *IET Control Theory & Applications*, vol. 6, pp. 1758–1765, 7 2012.

[46] T. Srinivasan, V. Vijaykumar, and R. Chandrasekar, "An auction based task allocation scheme for power-aware intrusion detection in wireless ad-hoc networks," in *2006 IFIP International Conference on Wireless and Optical Communications Networks*, pp. 5–pp, IEEE, 2006.

[47] M. J. Lyell, A. Webb, and J. Nanda, "Human-autonomous system interaction framework to support astronaut- multi-agent system interactions," in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, 1 2009.

[48] S. Han, H. Y. Youn, and O. Song, "Efficient category-based service discovery on multi-agent platform," *Information Systems Frontiers*, vol. 14, pp. 601–616, 12 2010.

[49] X. Wu, Y. Tang, J. Cao, and W. Zhang, "Distributed consensus of stochastic delayed multi-agent systems under asynchronous switching," *IEEE transactions on cybernetics*, vol. 46, pp. 1817–1827, 8 2015.

[50] J. Cai, G. Wang, and H. Wu, "Research of negotiation in network trade system based on multi-agent," in *SPIE Proceedings*, vol. 7490, pp. 74901W–, SPIE, 7 2009.

[51] A. Bezek, M. Gams, and I. Bratko, "Aamas - multi-agent strategic modeling in a robotic soccer domain," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 457–464, ACM, 5 2006.

[52] D. Weyns, *Architectural Design of Multi-Agent Systems*, pp. 55–92. Springer Berlin Heidelberg, 2 2010.

[53] J. H. Chu and H. Y. Youn, "Hierarchical p2p networking and two-level compression scheme for multi-agent system supporting context-aware applications," *The KIPS Transactions:PartA*, vol. 16, pp. 1–8, 2 2009.

[54] Z. Qiang, S. Yuqiang, and C. Yang, "A clustering algorithm based on multi-agent meta-heuristic architecture," *International Journal of Hybrid Information Technology*, vol. 7, pp. 227–236, 3 2014.

[55] K. Sakurama and K. Nakano, "Necessary and sufficient condition for average consensus of networked multi-agent systems with heterogeneous time delays," *International Journal of Systems Science*, vol. 46, pp. 818–830, 5 2013.

[56] C. Ramachandran, R. Malik, X. Jin, J. Gao, K. Nahrstedt, and J. Han, "Videomule: a consensus learning approach to multi-label classification from noisy user-generated videos," in *Proceedings of the 17th ACM international conference on Multimedia*, pp. 721–724, 2009.

[57] M. I. Dekhtyar, A. Dikovsky, and M. K. Valiev, *JELIA - Complexity of Multi-agent Systems Behavior*, pp. 125–136. Germany: Springer Berlin Heidelberg, 9 2002.

[58] R. Dobbe, D. Fridovich-Keil, and C. Tomlin, "Fully decentralized policies for multi-agent systems: An information theoretic approach," 1 2017.

[59] A. Belousov, A. Goryachev, P. Skobelev, and M. Stepanov, "A multi-agent method for adaptive real-time train scheduling with conflict limitations," *International Journal of Design & Nature and Ecodynamics*, vol. 11, pp. 116–126, 4 2016.

[60] J. M. D. Miranda, J. Borges, D. Valério, and M. J. G. C. Mendes, "Multi-agent management system for electric vehicle charging," *International Transactions on Electrical Energy Systems*, vol. 25, pp. 770–788, 1 2014.

[61] G. Enee and C. Escazut, *IWLCS - A Minimal Model of Communication for a Multi-agent Classifier System*. Germany: Springer Berlin Heidelberg, 6 2002.

[62] H.-T. Zhang, F. Yu, and W. Li, "Step-coordination algorithm of traffic control based on multi-agent system," *International Journal of Automation and Computing*, vol. 6, pp. 308–313, 8 2009.

[63] S. Yang, J.-X. Xu, X. Li, and D. Shen, "Iterative learning control for multi-agent coordination with initial state error," 3 2017.

[64] D. Strnad and N. Guid, "A multi-agent system for university course timetabling," *Applied Artificial Intelligence*, vol. 21, pp. 137–153, 2 2007.

[65] T. Tatarenko, *Game Theory and Multi-Agent Optimization*, pp. 7–26. Springer International Publishing, 9 2017.

[66] Y. Lao and W. Leong, *PRICAI - A Multi-agent Based Approach to the Inventory Routing Problem*, pp. 345–354. Germany: Springer Berlin Heidelberg, 8 2002.

[67] R. Chandrasekar and S. Misra, "Using zonal agent distribution effectively for routing in mobile ad hoc networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 3, no. 2, pp. 82–89, 2008.

[68] S. Yang, X. Liao, Y. Liu, X. Chen, and D. Ge, "Consensus of delayed multi-agent systems via intermittent impulsive control," *Asian Journal of Control*, vol. 19, pp. 941–950, 12 2016.

[69] H. Salem, G. Attiya, and N. El-Fishawy, "A survey of multi-agent based intelligent decision support system for medical classification problems," *International Journal of Computer Applications*, vol. 123, pp. 20–25, 8 2015.

[70] Y.-C. Choi and H.-S. Ahn, "The bio-insect and artificial robots interaction based on multi-agent reinforcement learning," in *Volume 3: ASME/IEEE 2009 International Conference on Mechatronic and Embedded Systems and Applications; 20th Reliability, Stress Analysis, and Failure Prevention Conference*, pp. 9–15, ASMEDC, 1 2009.