soloncouncil : Pages:1–19

Original Research

Decentralized Attribution Modeling: A Multi-Agent Approach to Enterprise-Scale Marketing Analytics

Paolo Reyes[1] and Jomar Villanueva[2]

[1]Luzon College of Technology, Department of Computer Science, Aurora Boulevard, Quezon City, Philippines.
[2]Mindanao Institute of Computing, Department of Computer Science, J.P. Laurel Avenue, Davao City, Philippines.

Abstract
Enterprise-scale marketing organizations increasingly rely on attribution models to allocate budgets across heterogeneous channels, yet conventional centralized approaches face structural challenges related to data silos, privacy regulations, and computational scalability. The consolidation of all interaction data into a single modeling environment is often incompatible with fragmented technology stacks, regional governance constraints, and independent experimentation agendas maintained by different business units. At the same time, organizations seek attribution estimates that are coherent at the global level, robust to heterogeneous data quality, and aligned with diverse local objectives such as revenue, profit, and long-term customer value. This paper investigates a decentralized attribution modeling framework based on a multi-agent perspective, in which autonomous agents control different subsets of data and decision variables while coordinating through constrained optimization protocols. The proposed view treats attribution as a cooperative game over a shared response surface, where each agent performs localized inference and contributes to a global allocation that satisfies conservation and consistency conditions. The paper develops a class of linear models that admit distributed estimation under communication and privacy constraints, characterizes equilibrium properties of agent interactions under convex objectives, and studies practical trade-offs between local flexibility and global coherence. A set of stylized enterprise scenarios illustrates how the framework can represent cross-region governance, channel ownership boundaries, and mixed measurement technologies. The analysis emphasizes modeling choices, algorithmic structure, and implementation considerations, with a focus on properties that can be evaluated under realistic enterprise constraints rather than on specific empirical benchmarks.

1. Introduction

Enterprise marketing environments are characterized by a large number of channels, complex customer journeys, and organizational boundaries that structure how data is collected and used [1]. Channels such as search, display, social, email, affiliate, and offline media operate with distinct optimization goals, measurement technologies, and contractual constraints. Attribution models attempt to assign credit for observed outcomes, such as conversions or revenue, to the sequence of touchpoints that precedes them. In many organizations, this task has historically been handled through centralized rule-based approaches or single-system statistical models that assume ready access to unified user-level data. As data flows become regulated, platforms adopt more restrictive data sharing practices, and internal teams maintain autonomy over their experimentation systems, the assumption of a fully centralized attribution pipeline becomes less realistic [2].

At the same time, the need for attribution remains. Budgeting processes require projections of marginal returns to incremental spend across channels and regions, while tactical bidding systems depend on attribution to translate high-level performance goals into local optimization signals. The tension between centralized and decentralized perspectives is therefore structural rather than incidental. Centralization simplifies modeling and yields a single global view, but it can be misaligned with data access patterns and organizational decision rights [3]. Decentralization respects autonomy and privacy but risks inconsistencies, double counting, or gaps if local models are not coordinated.

This paper considers attribution modeling from a multi-agent standpoint. Instead of assuming a single centralized learner, the enterprise is represented as a collection of agents, each responsible for a subset of marketing channels, markets, or technology stacks. Each agent observes local data, maintains local models, and participates in a coordination mechanism that aims to produce globally coherent attribution values. The modeling question becomes how to design objectives, constraints, and communication protocols such that the aggregate outcome of decentralized learning is interpretable as an attribution solution with desirable properties, for example conservation of total incremental value or approximate agreement with a counterfactual outcome definition.

Linear models play a central role in this analysis. They are expressive enough to capture many practical attribution schemes, from additive regression models over channel exposures to constrained budget allocation formulations that incorporate business rules. At the same time, linear structures yield optimization problems that can be decomposed along natural organizational boundaries and solved with distributed convex optimization methods [4]. The multi-agent view interacts with linear modeling in a specific way: agents own partitions of the design matrix or subsets of the parameter vector, and the coordination mechanism enforces consistency across these partitions.

The motivation for a decentralized approach is not only computational. Legal and commercial constraints can prohibit the movement of user-level data across regions or entities. Independent business units may wish to retain control over model features and objective choices [5]. Platforms may provide only aggregate or privacy-preserving signals that are difficult to reconcile in a central database. In such settings, attribution must be constructed from partial views, and multi-agent coordination becomes a natural abstraction.

The contributions of this work are conceptual, methodological, and algorithmic. Conceptually, it formulates attribution as a cooperative game between agents that control both data and decision variables [6]. Methodologically, it specifies a family of linear attribution models that can be decomposed and estimated under decentralization and describes conditions under which decentralized solutions coincide with a hypothetical centralized optimum. Algorithmically, it develops distributed optimization schemes that operate with limited communication, potentially noisy or privacy-perturbed messages, and asynchronous update patterns. Throughout the paper, the emphasis is on properties that can be evaluated in enterprise settings, such as stability of attribution under local model changes, sensitivity to heterogeneous noise levels, and transparency of the coordination mechanism.

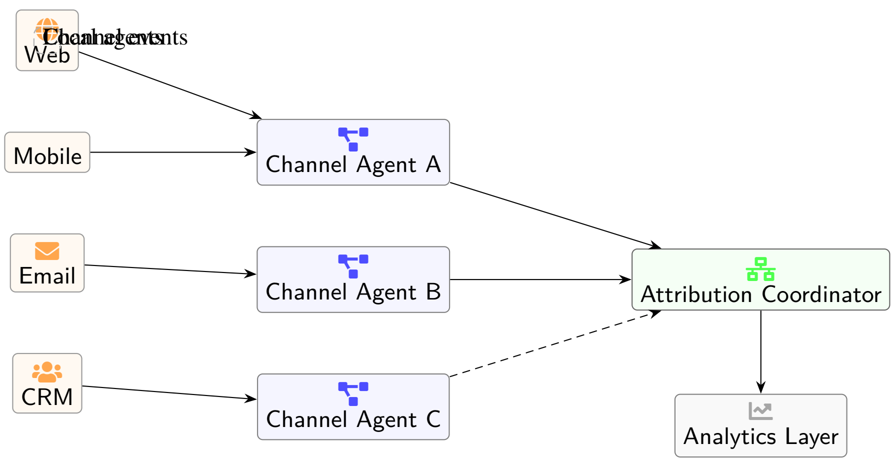

Figure 1: High-level multi-agent layout for decentralized attribution, where channel-specific agents summarize interaction streams and communicate compact signals to a central coordination layer that feeds enterprise analytics.

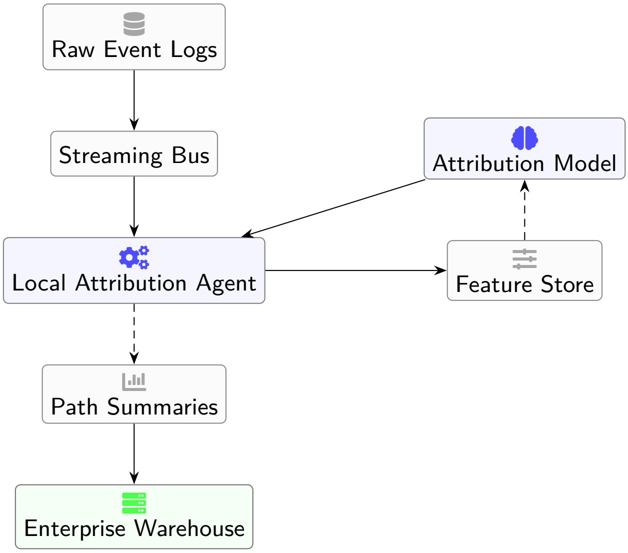

Figure 2: Event-level data flow from raw logs into a local attribution agent, which maintains a streaming feature store and model. Lightweight path summaries are persisted into the warehouse to enable downstream enterprise reporting without centralizing user-level traces.

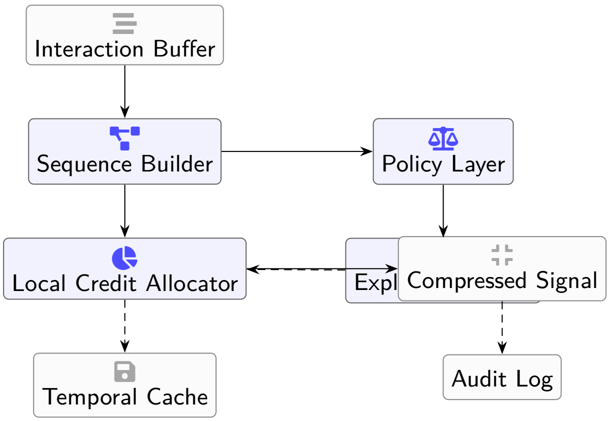

Figure 3: Internal decomposition of a local attribution agent into buffering, sequence construction, credit allocation, exploration, and compression components, enabling experimentation and logging while exposing only aggregated signals to external consumers.

2. Foundations of Attribution in Enterprise Marketing

Attribution modeling starts from a representation of customer interactions. Consider an enterprise that interacts with users through a set of channels indexed by a finite set [7] [8]. Each user generates a sequence of touchpoints over time, which can be represented as an ordered list of channel identifiers and possibly associated covariates such as timestamps, device types, and bids. A subset of these user journeys results in an outcome of interest, for example a purchase, subscription, or lead. The modeling task is to assign a share of the observed outcome value to the participating channels.

A canonical centralized formulation aggregates user journeys into a dataset where each observation corresponds to one user or one journey, with features that encode exposure to each channel and possibly temporal or contextual effects [9]. The outcome variable can be binary, representing conversion, or

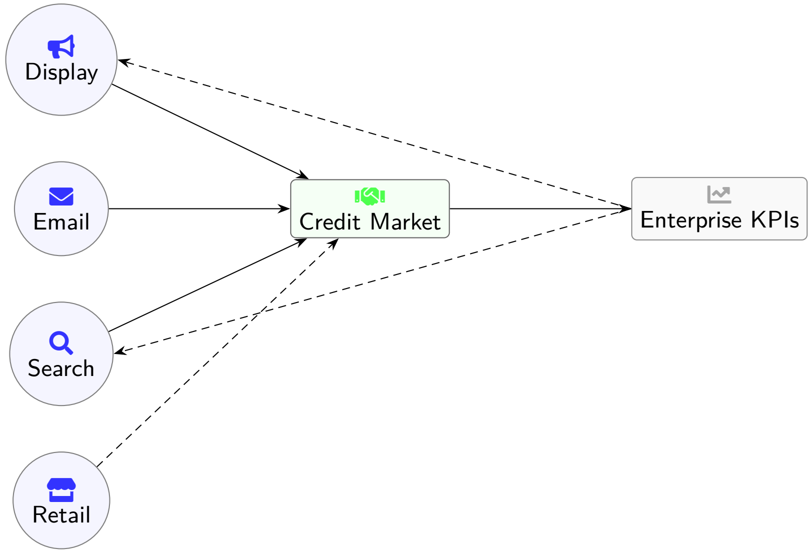

Figure 4: Multi-agent credit market where channel-specific agents submit local views of contribution and the market mechanism reconciles them into a global allocation consistent with enterprise-level outcome constraints. Selected feedback from realized KPIs is returned to agents to adapt bidding behavior over time.

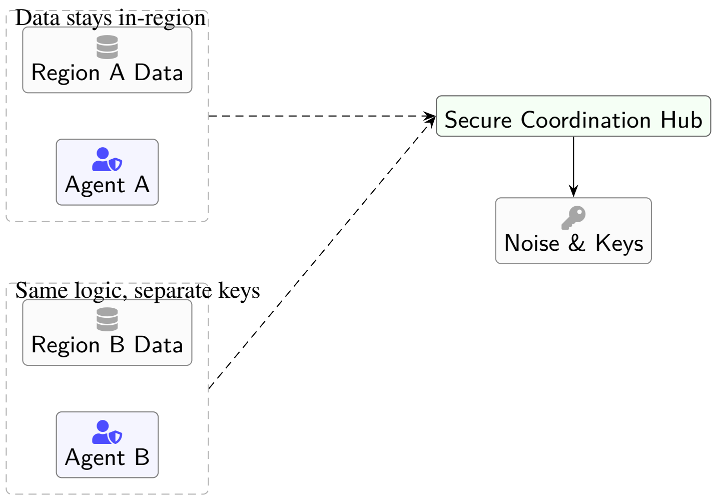

Figure 5: Separation of regional data zones and local agents from a secure coordination hub, ensuring that raw identifiers never cross boundaries while still enabling alignment of attribution signals through encrypted and noise-aware exchanges.

continuous, representing revenue or profit. A linear attribution model postulates that the expected outcome is a linear function of features that represent channel contributions, possibly after transformations that capture saturation or interaction effects. The parameters of this model can be interpreted as per-unit contributions, which are then aggregated to produce attribution values.

Formally, suppose journeys are indexed by an index that ranges over all observed journeys [10]. Each journey is associated with a feature vector in a real space of finite dimension and an outcome variable in the real numbers. A centralized linear model can be expressed as an equation of the form

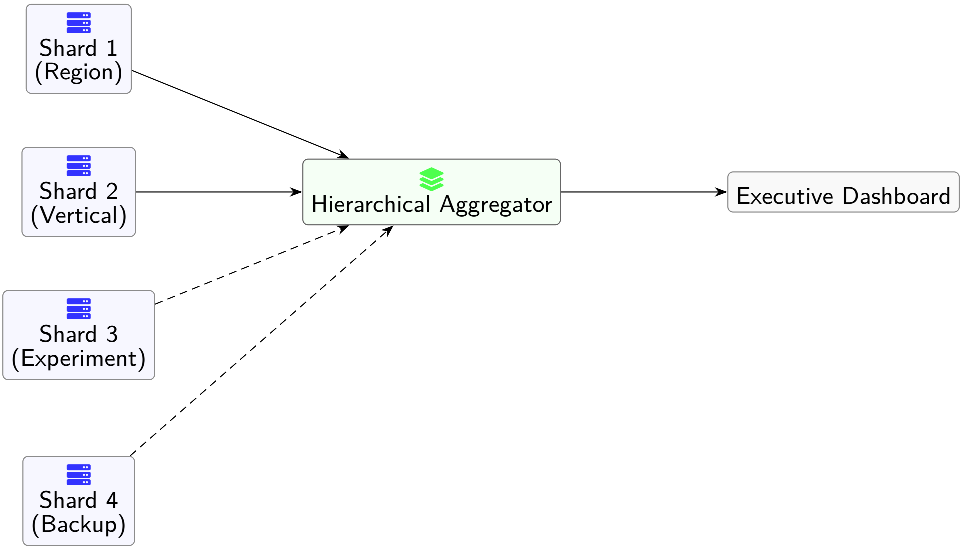

Figure 6: Hierarchical aggregation of attribution signals across multiple sharding dimensions (regional, business vertical, and experimental) into a unified layer, which supports stable enterprise-level dashboards while preserving the independence and scalability of underlying agents.

$$y_j = x_j^\top \beta \ \varepsilon_j 11$$

where the index enumerates journeys, the vector denotes features, the vector represents unknown parameters, and the error term captures unexplained variation. Features can encode channel exposures with various levels of granularity, such as counts, indicators for first or last touch, or time-discounted measures. Attribution for a given channel is typically computed as a function of the estimated parameters, sometimes after imposing constraints such as non-negativity or budget conservation [12].

In a counterfactual interpretation, attribution aims to measure the change in outcome that would occur if a given channel or set of channels were removed or suppressed, all else equal [13]. Under linearity and certain independence assumptions, the model parameters can be used to approximate such counterfactuals. For example, the incremental effect of channel removal for a given journey can be approximated by evaluating the linear model with altered features. Aggregating these incremental effects over journeys yields a global attribution measure for each channel.

Enterprise settings introduce several complications [14]. First, data may be distributed across systems, with some journeys observed only in certain regions or channels due to logging differences or privacy constraints. Second, outcome definitions may differ across business units; one team may focus on short-term revenue, another on long-term customer value, and yet another on intermediate engagement metrics. Third, channels may overlap or interact in ways that are difficult to capture in a purely additive model. There may also be delays between exposure and outcome, censored observations, and feedback loops where previous attribution decisions influence future data collection [15].

Despite these complexities, linear models remain a useful starting point because they offer a transparent mapping between features and attribution values. Linear structures define a design matrix whose rows correspond to journeys and whose columns represent features. The parameter vector can be interpreted as a set of coefficients that quantify the marginal effect of each feature on the outcome. Constraints can be imposed on these coefficients to reflect business rules, such as non-negative contributions or upper bounds based on domain knowledge. Regularization terms can be added to stabilize estimation in high-dimensional or noisy settings [16].

From a game-theoretic viewpoint, attribution can be viewed as the allocation of a total value associated with an outcome among contributing channels. Classical cooperative game concepts, such as marginal contribution-based allocations, provide one family of solutions. In an enterprise context, however, the payoff is not a scalar but a function of context, time, and cohort, and channels are not abstract players but elements of systems with different observability and control properties. This motivates a richer representation in which attribution arises from optimization problems defined over data and decision variables partitioned across organizational boundaries [17].

The problem formulation that will be used in subsequent sections proceeds as follows. The enterprise is partitioned into agents, each associated with a subset of channels or markets and a local dataset. Each agent has access to one or more blocks of the global design matrix and possibly to a local outcome definition. The goal is to estimate a global parameter vector that aggregates these local contributions while satisfying linear constraints that encode fairness, coherence, and conservation of value [18]. The attribution values assigned to channels are functions of this parameter vector and, in some cases, of local post-processing performed by agents.

Under this formulation, a centralized benchmark can be defined as the solution to a global optimization problem that treats all data and constraints jointly. The decentralized multi-agent approach will then be constructed as an approximation to this benchmark, implemented via distributed optimization and coordination protocols. The conditions under which the decentralized solution coincides with or approximates the centralized one will depend on the structure of the design matrix, the alignment of objectives, and the communication patterns among agents [19].

3. Decentralized Multi-Agent System Design

Table 1: Enterprise marketing datasets used for decentralized attribution experiments

Dataset	Impressions (M)	Conversions (K)	# Channels
Retail-Global	480.2	312.4	11
FinServ-EMEA	163.7	89.1	8
SaaS-Enterprise	92.5	41.8	7
Travel-MultiBrand	215.3	127.6	10
MobApp-Freemium	138.9	54.3	6

Table 2: Roles of collaborating agents in the multi-agent attribution system

Agent	Primary responsibility	Key input signals	Output artifacts
Collector	Event ingestion	Raw logs, pixels	Canonical user journeys
Channel-Expert	Channel-specific modeling	Channel features, bids	Per-channel credit curves
Path-Reasoner	Sequence pattern discovery	Ordered touchpoint paths	Path-level contribution scores
Budget-Planner	Spend and constraint encoding	Historical spend, targets	Budget-aware priors
Coordinator	Global belief aggregation	All local agent posteriors	Final attribution weights

To represent enterprise-scale attribution as a decentralized process, consider a set of agents indexed by an index that ranges over organizational units. Each agent may correspond to a regional marketing team, a channel owner, or a platform-specific measurement system. Agent data consists of one or more blocks of the global design matrix, representing journeys in which its channels participate, with features that might be partially overlapping across agents. Agents observe local outcomes, either directly measured or derived from shared outcomes via aggregation or attribution rules [20].

Table 3: Comparison of attribution approaches evaluated in the enterprise setting

Method	Decentralized execution	Online updating	Relative lift in ROI (%)
Last-touch heuristic	No	No	0.0
Markov chain model	No	Limited	4.7
Shapley value (central)	No	No	7.9
Proposed multi-agent	Yes	Yes	13.4

Table 4: Accuracy and reliability metrics: centralized baseline vs. decentralized attribution

Metric	Centralized baseline	Multi-agent decentralized	Relative change (%)
Conversion MAPE	9.8	6.1	−37.8
Channel ROI error (p.p.)	4.3	2.5	−41.9
Attribution latency (ms)	850	310	−63.5
System uptime (30 days, %)	98.2	99.6	1.4

Table 5: Inter-agent communication channels in the decentralized architecture

Channel type	Typical payload	Average frequency	Delivery reliability (%)
Event stream	Normalized user events	Per event	99.9
Belief broadcast	Channel credit distributions	Every 2 minutes	99.5
Gradient summaries	Model update statistics	Every 5 minutes	99.2
Control messages	Routing and throttling	On configuration change	99.8
Checkpoint sync	Model snapshots	Hourly	99.7

Table 6: Ablation study on coordination and learning in the multi-agent system

Configuration	Coordination protocol	Normalized MI (NMI)	Incremental revenue lift (%)
No coordination	None	0.41	4.3
Local-only aggregation	Ring gossip	0.56	7.1
Coordinator without priors	Central hub	0.63	9.2
Full system (no budgets)	Central hub + priors	0.68	11.0
Full system	Hub + priors + budgets	0.71	13.4

The multi-agent system is cooperative in the sense that all agents benefit from an accurate and stable attribution scheme, even though each agent may prioritize specific metrics or constraints. The global objective is defined in terms of a loss function that measures the fit of the attribution model to observed outcomes, regularization terms that enforce stability, and penalty terms that capture deviations from conservation or coherence constraints. Each agent controls a subset of the model parameters and possibly some auxiliary variables associated with its local implementation.

A basic representation of the joint model partitions the global parameter vector into blocks, one per agent [21]. Let the global parameter vector be written as a vertical concatenation of agent-specific parameter vectors. The centralized linear model equation for journey outcomes can then be expressed as

$$y_j = {}_{i=1}^{N} \, x_{i,j}^{\top} \beta_i \;\; \varepsilon_j$$ [22]

where the index enumerates agents, the vector denotes the subset of features for journey associated with agent, and the vector contains the parameters controlled by agent. Some journeys may not involve

Table 7: Scalability of decentralized attribution under increasing enterprise load

Scale regime	# Active accounts	Throughput (events/s)	Cost per 1M events (USD)
Pilot	25	8.1k	7.84
Regional	120	19.7k	6.51
Multi-region	420	55.3k	5.29
Global	1,050	96.8k	4.92
Peak season	1,650	131.4k	4.75

Table 8: Key learning and optimization hyperparameters of the agent policies

Component	Learning rate	Update interval	Reward signal
Channel-Expert agents	3×10^{-4}	5 minutes	Incremental conversions
Path-Reasoner agent	1×10^{-4}	10 minutes	Path-level log-lift
Budget-Planner agent	5×10^{-5}	30 minutes	Revenue minus spend
Coordinator agent	2×10^{-4}	5 minutes	Global ROI consistency

certain agents, in which case the corresponding feature block is zero. This representation allows agents to maintain responsibility for different parts of the design matrix.

Each agent defines a local loss over its data. For agent, with local design matrix and outcome vector, a typical quadratic loss takes the form [23]

$$L_i\beta_i = \frac{1}{2}\|X_i\beta_i - y_i\|_2^2 \;\; R_i\beta_i 24$$

where the function denotes a regularization term that can encode sparsity, smoothness, or prior beliefs about parameter magnitudes. This local loss is not yet sufficient to ensure global coherence, because each agent's view of the outcome may differ and shared constraints are not enforced.

To couple the agents, a global objective can be defined as the sum of local losses plus penalty terms for disagreements with global constraints. One approach introduces a global consensus parameter vector, which represents the attribution parameters that would be used in a hypothetical centralized model [25]. Each agent then maintains a local copy of this vector or of its block and incurs penalties for deviating from consensus. A simple quadratic penalty yields an objective of the form

$$J\{\beta_i\}, \beta = \overset{N}{\underset{i=1}{}} L_i\beta_i \;\; \overset{N}{\underset{i=1}{}} \frac{\rho_i}{2}\|\beta_i 26 - P_i\beta\|_2^2$$

where the vector denotes the consensus parameter vector, the matrix selects the relevant coordinates for agent, and the coefficients control the strength of coupling. Minimizing this objective over the local and consensus variables involves coordination because each agent's parameters appear in the consensus terms [27].

Alternatively, the problem can be formulated directly at the level of attribution values rather than model parameters. Each agent defines local attribution weights over its channels or activities, and a global constraint requires that the sum of agent attributions for any given outcome equals the observed value or a reference incremental value. Let denote the vector of attributions assigned by agent across outcomes. A conservation constraint can then be written as [28]

$$\overset{N}{\underset{i=1}{}} a_{i,j} = v_j$$

for each outcome index, where represents the target value to be allocated. Agents seek attributions that are compatible with their local data and objectives while satisfying these linear constraints across outcomes.

The multi-agent system can also be viewed as implementing a potential game. Define a potential function equal to the negative global loss, so that any unilateral change by an agent that improves its local objective corresponds to an increase in potential [29]. Under certain conditions on the structure of local losses and coupling terms, the system admits at least one Nash equilibrium that coincides with a local or global optimum of the potential. This perspective supports the use of best response dynamics and other distributed learning algorithms that converge to equilibria under appropriate assumptions.

Communication in this system is mediated by messages that summarize local information rather than by raw data exchanges. Agents may share sufficient statistics such as marginal feature-outcome correlations, gradient information evaluated at current parameter values, or compressed representations of local residuals. In privacy-sensitive settings, agents can add noise to these messages or share only aggregated quantities that do not reveal individual-level information [30]. The design of communication protocols must balance fidelity, privacy, and network constraints.

The decentralized system is subject to practical considerations such as asynchronous updates, partial participation, and failures. Some agents may update their models frequently, others infrequently, and some may temporarily disconnect. Algorithms must tolerate these variations without creating large inconsistencies in attribution [31]. The consensus representation is helpful here, as it allows global parameters to be maintained even when some agents are inactive, with their contributions interpolated or held constant.

This multi-agent design reframes attribution as a process executed by a network of learning systems rather than a static computation in a single central engine. The next sections translate this design into explicit linear models and optimization formulations, and analyze how properties of the multi-agent system affect the resulting attribution values.

4. Linear Attribution Models and Constraints

Linear models are attractive for decentralized attribution because they map naturally onto partitions of features and parameters and yield convex optimization problems [32]. Consider again the centralized linear model, expressed in matrix form as

$$y = X\beta \;\; \varepsilon$$

where the vector collects outcomes for journeys, the matrix is the design matrix, the vector is the parameter vector, and the vector is the error term. In a decentralized setting, the design matrix can be partitioned column-wise according to agents, so that [33]

$$X = X_1 \;\; X_2 \;\; \cdots \;\; X_N$$

and the parameter vector is partitioned accordingly as a vertical concatenation of the agent-specific vectors. The outcome vector remains shared, although individual agents may see only subsets depending on data access policies.

Attribution values are often computed as linear functions of model predictions [34]. For a given journey, the predicted outcome is given by the inner product between its feature vector and the parameter vector. The attribution for channel can be defined as the contribution of the corresponding features and parameters to this prediction. For example, if the feature encoding includes an indicator variable for exposure to channel, the attribution for that channel on a given journey could be taken as the product between the indicator and the associated coefficient. Aggregating these contributions across journeys yields channel-level attribution values [35].

To incorporate business rules, linear constraints can be imposed on the parameters or on attribution aggregates. A typical conservation requirement is that the sum of attributions across channels for each outcome equals the predicted or observed outcome value. In a linear model, this can be enforced by design if features partition outcomes, but in practice it is common to define attribution aggregates that must satisfy

$$_{k} A_{k,j} = v_j$$

for each outcome index, where denotes the attribution assigned to channel for outcome and denotes a target value such as observed revenue or estimated incremental value [36]. These aggregates can be expressed as linear functions of the parameters, leading to constraints of the form

$$C\beta = d$$

where the matrix encodes attribution aggregation rules and the vector collects target values.

Non-negativity constraints are another common requirement. Channels are typically not allowed to receive negative attribution, both for interpretability and for alignment with budgeting processes. This leads to constraints of the form [37]

$$\beta \geq 0$$

interpreted component-wise. In some cases, specific subsets of parameters may be constrained, while others remain unconstrained to allow for intercepts or adjustments.

When multiple outcome metrics are involved, multi-task linear models can be constructed by stacking parameter vectors or by considering a matrix of parameters with one column per outcome. Let denote a matrix whose columns correspond to different metrics [38]. The linear relationship can then be written as

$$Y = XB \;\; E$$

where the matrix collects outcomes across metrics, the matrix is the design matrix, the matrix contains parameters, and the matrix contains residuals. Attribution constraints can be extended to this setting by requiring conservation and non-negativity for each metric separately or for weighted combinations of metrics.

Regularization is often necessary in high-dimensional attribution models where features outnumber observations or where multicollinearity is present [39]. Linear and quadratic penalties can be used to stabilize estimation and to encode preferences such as sparsity or smoothness across related channels. A typical regularized objective takes the form

$$\min_{\beta} \; \frac{1}{2}\|X\beta - y\|_2^2 \;\; \lambda 40\|\beta\|_1 \;\; \frac{\mu}{2}\|\beta\|_2^2$$

subject to linear constraints such as conservation and non-negativity. Here, the parameters control the strength of sparsity and ridge penalties respectively. In a decentralized setting, these penalties can be applied locally per agent, allowing different regularization structures across organizational units [41].

From a multi-agent perspective, linear constraints also serve to couple agent decisions. Suppose each agent maintains local parameters and local attribution aggregates. Global coherence requires that these local aggregates satisfy shared constraints, such as matching total attribution to global outcomes or respecting cross-channel budget caps. These conditions can be expressed as a system of linear equations and inequalities that involve all agent parameters [42]. The global feasible set is then the intersection of local feasible sets with the shared constraint set, which is convex due to the linear structure.

An important concept in this setting is identifiability. Even in a centralized model, attribution parameters may not be uniquely determined by the data and constraints, particularly when features are highly correlated or when aggregation is coarse. Decentralization adds further ambiguities because agents may observe different subsets of data and because communication constraints limit the exchange of information [43]. Linear algebra provides tools to analyze identifiability, such as examining the rank of the design matrix and the null space of constraint matrices. In a multi-agent system, these properties can be studied both locally and globally.

In some cases, direct parameter estimation is not necessary; instead, attribution can be formulated as a linear programming problem that allocates value subject to constraints derived from observed data or model outputs. For example, suppose that for each journey we know the total incremental value and a set of feasible attribution vectors based on rules or prior models. A linear program can then select attribution values that minimize a cost function subject to conservation and fairness constraints [44]. Such formulations are amenable to decomposition across agents because the objective and constraints are linear and separable across subsets of variables, subject to coupling constraints that enforce global conditions.

The linear models and constraints described in this section provide the basis for the distributed optimization schemes developed next. The key properties are convexity, separability across agents, and the ability to encode business rules as linear equalities and inequalities. These properties support the design of multi-agent algorithms that converge to solutions interpretable as enterprise-wide attribution allocations [45].

5. Distributed Learning and Coordination Algorithms

Given a decentralized linear attribution formulation, the next step is to design algorithms that allow agents to learn parameters or attribution values while exchanging limited information. Distributed convex optimization provides a natural foundation. Many attribution objectives can be expressed as sums of convex functions plus linear constraints, which can be tackled with methods such as gradient-based consensus algorithms, dual decomposition, or variants of the alternating direction method of multipliers.

Consider a generic problem in which agents seek to minimize a global objective that is the sum of their local losses subject to a shared linear constraint on a consensus parameter vector [46]. A prototypical formulation is

$$\min_{\{\beta_i\},\beta} \; {}_{i=1}^{N} L_i \beta_i$$

subject to

$$\beta_i 47 = P_i \beta, \quad i = 1, \ldots, N$$

where the objective captures local losses and the constraints enforce consistency between local parameters and the global consensus. This structure fits the standard pattern for distributed optimization with consensus constraints [48].

One approach to solve this problem is an augmented Lagrangian method. Introduce dual variables associated with each constraint and form the augmented Lagrangian

$$\mathcal{L} = {}_{i=1}^{N} \left(L_i \beta_i \;\; \lambda_i^\top 49 \beta_i - P_i \beta \;\; \frac{\rho}{2} \|\beta_i - P_i \beta\|_2^2 50 \right)$$

where the vectors are dual variables and the scalar is a penalty parameter shared across agents for simplicity. An iterative algorithm alternates between minimizing with respect to primal variables and updating dual variables by gradient ascent. In a distributed setting, each agent updates its local parameters using only local data and the current consensus estimate, while the consensus variable aggregates information from agents.

The agent update step solves a convex problem of the form

$$\beta_i^{t1} = \arg\min_{\beta_i} \big(L_i\beta_i 51\ \ \lambda_i^{t\top}\beta_i\ \ \frac{\rho}{2}\|\beta_i - P_i\beta^t\|_2^2\big)$$

which can often be computed by closed-form updates or efficient numerical methods due to the structure of [52]. For quadratic losses with linear regularization, the update reduces to solving a linear system or performing a proximal gradient step. The consensus variable update aggregates the corrected local parameters, for example via a weighted average,

$$\beta^{t1} = \big(\ {}_{i=1}^{N} \rho P_i^\top P_i\big)^{-1}\ {}_{i=1}^{N} P_i^\top 53\rho\beta_i^{t1}\ \ \lambda_i^t$$

under the assumption that the matrix inside the inverse is positive definite. Dual variables are then updated as

$$\lambda_i^{t1} = \lambda_i^t\ \ \rho\beta_i^{t1} - P_i\beta^{t1}$$

These updates can be implemented with message passing: agents send their updated local parameters and dual variables to a coordinator or to neighboring agents, which compute the consensus update and return the new consensus vector.

In attribution applications, the choice of local loss function affects both interpretability and algorithmic efficiency [54]. When the outcome variables are continuous and the loss is quadratic, the updates take the form of linear algebra operations that can be optimized at scale. When outcomes are binary or counts and logistic or Poisson losses are used, local updates may require iterative methods, but the overall structure of the coordination scheme remains similar. The linear constraints that encode conservation and non-negativity can be handled via projection steps or by incorporating them into the augmented Lagrangian.

Another class of distributed algorithms relies on consensus averaging without explicit dual variables [55]. In these methods, each agent iteratively updates its parameters by combining local gradient steps with weighted averages of neighboring agent parameters. A simple iteration can be written as

$$\beta_i^{t1} = {}_j\, w_{ij}\beta_j^t - \eta_t \nabla L_i\beta_i^t$$

where the weights form a matrix that reflects the communication graph and satisfies standard conditions for consensus, and the step size controls the gradient update [56]. Under convexity and appropriate step size schedules, such algorithms converge to a common parameter vector that minimizes the sum of local losses. In practice, attribution systems may use such schemes over networks that reflect organizational reporting lines or data sharing agreements rather than physical connectivity.

Privacy considerations can be integrated into distributed learning via noise injection or secure aggregation. For example, agents may add calibrated noise to gradient or parameter messages to satisfy differential privacy constraints [57]. In a linear model, this translates into perturbations of sufficient statistics such as feature-outcome inner products. The impact of such perturbations on attribution estimates can be analyzed using sensitivity bounds and concentration inequalities. At a high level, noise introduces bias and variance trade-offs: stronger privacy guarantees require more noise, which may reduce attribution accuracy or increase the need for regularization.

Asynchronous operation is particularly relevant in enterprise environments where systems update at different cadences. In asynchronous variants of distributed algorithms, agents perform local updates based on potentially stale information about other agents' parameters or the consensus vector [58]. Convergence analysis typically requires assumptions about bounded staleness and bounded delays, as well as suitably small step sizes. In practice, asynchronous schemes are appealing because they accommodate diverse infrastructure constraints and reduce the need for global synchronization.

Distributed algorithms for attribution must also address robustness to local model misspecification. Different agents may adopt different feature sets, loss functions, or regularization schemes [59]. Coordinating such heterogeneous models requires mechanisms that reconcile differences in the implied attribution allocations. One approach is to define a global evaluation metric and adjust agent weights or constraints to align local models with this metric. Another is to treat local models as proposals and use a meta-optimization layer to reconcile their predictions into a coherent attribution. Linear structures facilitate such meta-optimization because the combination of linear predictions remains linear, and constraints can be expressed at the level of combined parameters or predictions [60].

The complexity of these algorithms can be analyzed in terms of problem dimension, number of agents, and communication patterns. For quadratic losses, each local update may require solving linear systems whose complexity is governed by the size and condition number of local design matrices. Communication complexity depends on the size of parameters or messages transmitted, which can be reduced via compression or low-rank representations. Trade-offs between computation and communication are central: more local computation can reduce the number of communication rounds required for convergence, while richer messages can accelerate convergence at the cost of higher communication overhead [61].

Overall, distributed learning and coordination algorithms provide the operational backbone for decentralized attribution models. They translate multi-agent design into concrete update rules that can be deployed in enterprise systems, subject to privacy, reliability, and scalability constraints. The next section discusses how these algorithms can be embedded into practical pipelines and evaluates their behavior in stylized enterprise scenarios.

6. Practical Deployment and Case Analysis

Implementing decentralized attribution in an enterprise involves both modeling and system-level decisions [62]. Data sources span ad servers, web and app analytics platforms, customer relationship management systems, and offline transaction databases. Each of these may be controlled by different teams or subject to distinct regulatory regimes. A multi-agent attribution system must align with existing data flows rather than replace them, which leads to architectures where agents are co-located with data sources and communicate summary statistics rather than raw records.

A typical deployment begins by defining the partition of the organization into agents [63]. For example, an enterprise might designate separate agents for major regions, product lines, or media types. Each agent integrates its local data into a feature representation that maps journeys into feature vectors, including exposure indicators, contextual variables, and derived metrics such as recency or frequency. Feature engineering remains local, allowing agents to adapt to data idiosyncrasies and business needs, while coordination focuses on shared parameters or attribution constraints.

The synchronization layer orchestrates distributed learning [64]. In systems where a central coordinator is acceptable, a service can collect gradient or parameter messages from agents, update consensus variables, and broadcast new values. In more constrained environments, peer-to-peer communication may be used, with agents forming a logical network over which consensus algorithms operate. The design must account for network reliability, latency, and security, particularly when agents reside in different regions or legal jurisdictions.

Privacy and compliance play a prominent role. Data minimization principles discourage the movement of detailed user-level logs across boundaries [65]. Instead, agents are encouraged to compute local sufficient statistics that support attribution modeling while aggregating or anonymizing data. For example, an agent can compute gram matrices and cross-product vectors over local features and outcomes and share only these aggregates. In a linear regression setting, such statistics can be sufficient for parameter estimation. When additional privacy guarantees are required, agents can perturb these statistics before sharing, at the cost of introducing noise into the estimation process [66].

Enterprise processes also impose constraints on update frequency. Budget decisions may be made monthly or quarterly, while bidding systems operate on hourly or daily horizons. Attribution updates need not match the highest-frequency decision layer as long as they maintain sufficient stability for operational use. Decentralized algorithms can be run on schedules that align with data availability and system capacity, for example with daily local updates and weekly global consensus refinements [67]. The asynchronous nature of multi-agent algorithms supports such staggered schedules.

Case analysis of decentralized attribution must consider both statistical and organizational dimensions. On the statistical side, one can compare decentralized estimates with those produced by hypothetical centralized models on overlapping subsets of data where centralization is feasible. Such comparisons reveal the extent to which linear models and distributed algorithms recover centralized solutions and how deviations arise due to data partitioning, communication noise, or heterogeneity in local modeling choices [68]. On the organizational side, one can examine how the introduction of decentralized attribution affects budgeting, channel optimization, and cross-team negotiations.

In a stylized scenario, consider an enterprise with agents corresponding to two major regions and one global brand team. Each region operates its own media mix, maintains local analytics, and reports revenue metrics that reflect regional currencies and tax structures. The brand team manages cross-region campaigns and cares about global reach and brand equity [69]. A decentralized attribution model can allow each region to maintain local outcome definitions and feature engineering while contributing to a global parameter vector that governs how shared channels are valued across regions. Linear constraints can enforce that shared channels receive consistent relative attribution across regions, subject to scaling factors that reflect market size or currency conversion.

Another scenario involves blended measurement technologies. Some channels provide user-level conversion logs, others provide only aggregate reports or modeled conversions due to privacy-preserving mechanisms [70]. Agents corresponding to channels with limited observability can encode their evidence via coarse-grained features or prior distributions on contribution parameters. Distributed algorithms can then incorporate these priors alongside more granular data from other agents, yielding attribution estimates that respect both direct observations and model-based inferences. Linear models facilitate this integration because prior information can often be represented as quadratic penalties or equality constraints on parameters.

Operational robustness requires monitoring and diagnostics [71]. Attribution systems must detect and handle anomalies such as data ingestion failures, sudden changes in channel behavior, or model drift. In a multi-agent system, anomalies may be localized to specific agents. Diagnostic procedures can include monitoring residual distributions, tracking parameter trajectories, and evaluating stability of attribution under perturbations of local data. Because linear models yield relatively interpretable parameters, stakeholders can inspect coefficients and aggregates to understand changes and to adjust constraints or regularization if necessary.

Decision integration is the final step [72]. Attribution outputs feed into processes such as budget allocation, bid strategy tuning, and creative optimization. In a decentralized environment, agents may use attribution both as a signal for their own local decisions and as a negotiation artifact when interacting with other teams. For example, channel owners may use attribution to argue for budget increases, while finance teams may use attribution to reconcile spending and outcome projections. A multi-agent attribution system must therefore provide interfaces that expose both local and global views, with clear documentation of how distributed algorithms and linear constraints shape the resulting allocations [73].

Taken together, these deployment considerations illustrate that decentralized attribution is not merely a theoretical construct but a practical response to organizational and technical constraints. The multi-agent linear modeling framework provides structure for designing and evaluating such systems, but successful implementation requires attention to data pipelines, privacy, governance, and integration with existing decision processes.

7. Conclusion

This paper has examined decentralized attribution modeling through the lens of multi-agent systems and linear optimization. Starting from a centralized representation of attribution as a linear mapping from features to outcomes, the analysis introduced a partition of data and parameters across agents that correspond to organizational units, channels, or measurement platforms [74]. Within this partitioned structure, attribution was formulated as a cooperative task in which agents manage local models and data while coordinating through consensus variables and shared linear constraints.

The linear modeling perspective enabled the definition of convex objectives and constraint sets that support distributed optimization. By expressing both model fitting and business rules as linear algebraic structures, the framework admits algorithms based on augmented Lagrangians, consensus gradients, and related methods. These algorithms operate on local sufficient statistics and exchange compressed messages rather than raw data, aligning with privacy and governance requirements common in enterprise environments [75]. The analysis highlighted conditions under which decentralized solutions coincide with those of a hypothetical centralized model and pointed out sources of discrepancy such as heterogeneous local objectives, measurement noise, and communication constraints.

Beyond the mathematical formulation, the discussion considered practical deployment aspects. Enterprises must align agent definitions with organizational boundaries, implement data pipelines that compute and share relevant statistics, and design synchronization schemes that reflect infrastructure and regulatory constraints. The multi-agent framework accommodates these requirements by allowing flexible partitions and communication patterns, while linear constraints help maintain global coherence in attribution allocations [76]. Stylized scenarios illustrated how regional differences, heterogeneous measurement technologies, and privacy-preserving mechanisms can be represented within the framework.

The study has focused on linear models because of their transparency, interpretability, and compatibility with distributed convex optimization. Nonlinear models, including those based on interactions or deep architectures, may capture more complex effects but pose additional challenges for decomposition and coordination. Extensions of the framework could explore how nonlinear components can be approximated or linearized for the purposes of decentralized attribution, or how hybrid systems can combine local nonlinear models with global linear coordination layers [77].

Future work can also investigate more detailed properties of multi-agent attribution systems, including dynamic adaptation to changing channel behavior, robustness to strategic responses by stakeholders, and integration with experimental designs such as randomized holdouts or geo-based tests. Another direction is to formalize the relationship between decentralized attribution and established value allocation principles from cooperative game theory under constraints on data access and communication.

In summary, the multi-agent linear framework described here offers a structured way to think about attribution in enterprises where centralization is constrained by organizational, technical, or regulatory factors. It provides tools for designing models and algorithms that respect decentralization while aiming for coherent and interpretable attribution outcomes. The emphasis on linear structures and distributed optimization supports both analytical understanding and practical implementation, leaving room for further exploration of richer models and coordination mechanisms in complex marketing environments [78].

References

[1] B. Liu, Y. Shi, H. Su, and X. Han, "Second-order controllability of multi-agent systems with multiple leaders," *Communications in Theoretical Physics*, vol. 65, pp. 585–592, 5 2016.

[2] H. Yang, Z. Zhang, and S. Zhang, "Consensus of second-order multi-agent systems with exogenous disturbances," *International Journal of Robust and Nonlinear Control*, vol. 21, pp. 945–956, 8 2010.

[3] S. Rudolph and M. Croitoru, *SGAI Conf. - Multi-Agent Knowledge Allocation*, pp. 165–178. Springer London, 10 2012.

[4] R. Brafman, G. Shani, and S. Zilberstein, “Qualitative planning under partial observability in multi-agent domains,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 27, pp. 130–137, 6 2013.

[5] W. L. Wang and Y. J. Wang, “Modeling of multi-agent system for drilling simulation,” *Advanced Materials Research*, vol. 383-390, pp. 1555–1561, 11 2011.

[6] T. Kristensen and M. Dyngeland, “Design and development of a multi-agent e-learning system,” *International Journal of Agent Technologies and Systems*, vol. 7, pp. 19–74, 4 2015.

[7] S. Landau and S. Picault, “Multi-agent-systems and applications - modeling adaptive multi-agent systems inspired by developmental biology,” *Lecture Notes in Computer Science*, vol. 2322, pp. 221–232, 4 2002.

[8] S. Kukkuhalli, “Implementing multi-touch attribution at scale for a global sales and marketing team in an enterprise environment,” *International Journal of Scientific Research in Engineering and Management., DOI*, vol. 10, 2024.

[9] J.-F. Martínez, A.-B. García, A.-M. Sanz, L. López, V. Hernández, and A. Dasilva, “Eatis - an approach for applying multi-agent technology into wireless sensor networks,” in *Proceedings of the 2007 Euro American conference on Telematics and information systems*, pp. 22–8, ACM, 5 2007.

[10] A. Kundu, S. K. Guha, A. R. Pal, T. Sarkar, S. Mandal, R. Dattagupta, and D. Mukhopadhyay, “Fuzzy based multi-agent system offering cost effective corporate envi- ronment,” *The Open Automation and Control Systems Journal*, vol. 1, pp. 65–81, 11 2008.

[11] W. Li, *A Multi-agent Economic Simulation System*, pp. 69–73. Germany: Springer Berlin Heidelberg, 11 2011.

[12] 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.

[13] R. Becker, A. Carlin, V. Lesser, and S. Zilberstein, “Analyzing myopic approaches for multi-agent communication,” *Computational Intelligence*, vol. 25, pp. 31–50, 2 2009.

[14] Z. Meng, Y. Cao, and W. Ren, “Stability and convergence analysis of multi-agent consensus with information reuse,” *International Journal of Control*, vol. 83, pp. 1081–1092, 4 2010.

[15] F. Bergenti and A. Poggi, *Developing Smart Emergency Applications with Multi-Agent Systems*, pp. 31–44. IGI Global, 4 2012.

[16] 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.

[17] F. A. Kolushev and A. A. Bogdanov, *Ershov Memorial Conference - Multi-agent Optimal Path Planning for Mobile Robots in Environment with Obstacles*, pp. 503–510. Germany: Springer Berlin Heidelberg, 1 2000.

[18] K. Lakkaraju and L. Gasser, “Aamas - a unified framework for multi-agent agreement,” in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pp. 21–3, ACM, 5 2007.

[19] H. R. Naji, “High performance physical environmental security using distributed cooperative sensor nodes,” *International Journal of Computer Applications*, vol. 12, pp. 34–38, 12 2010.

[20] S. S. Chouhan and R. Niyogi, “Mapja: Multi-agent planning with joint actions,” *Applied Intelligence*, vol. 47, pp. 1044–1058, 5 2017.

[21] Z. Li and D. Zhisheng, *Cooperative Control of Multi-Agent Systems*. CRC Press, 12 2017.

[22] N. O. Garanina, E. V. Bodin, and E. A. Sidorova, “An approach to model checking of multi-agent data analysis,” *Electronic Proceedings in Theoretical Computer Science*, vol. 168, pp. 32–44, 11 2014.

[23] R. W. Thomas and K. J. Hammond, “Iui - java settlers: a research environment for studying multi-agent negotiation,” in *Proceedings of the 7th international conference on Intelligent user interfaces*, pp. 240–240, ACM, 1 2002.

[24] S. Pujari and S. Mukhopadhyay, “Modeling of multi-agent oriented learning system for impaired students with jade,” *International Journal of Intelligent Systems and Applications*, vol. 4, pp. 48–56, 11 2012.

[25] M. Wang and K. Uchida, “Interval consensus problem of multi-agent systems in accordance with switching protocol,” *International Journal of Systems Science*, vol. 47, pp. 707–717, 3 2014.

[26] C. Cleophas, "Multi-agent modelling for revenue management," *Journal of Revenue and Pricing Management*, vol. 11, pp. 240–242, 2 2012.

[27] A. Z.Tsague, E. T. Fute, and L. P. Fotso, "Modeling of an environment for electronic archiving using a multi-agent approach," *International Journal of Applied Information Systems*, vol. 5, pp. 32–37, 5 2013.

[28] 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.

[29] M. Li and Q. Li, "Admissible consensus of multi-agent singular systems," *Asian Journal of Control*, vol. 16, pp. 1169–1178, 10 2013.

[30] Q. Hao, W. Shen, Z. Zhang, S.-W. Park, and J.-K. Lee, "A multi-agent framework for collaborative engineering design and optimization," in *Volume 4: 24th Computers and Information in Engineering Conference*, pp. 337–346, ASMEDC, 1 2004.

[31] L. Wang and Z. Liu, "Robust consensus of multi-agent systems with noise," *Science in China Series F: Information Sciences*, vol. 52, pp. 824–834, 3 2009.

[32] Y.-M. Jian and M. Chen, *Research on Mental Coefficient of a Multi-Agent Negotiation Model*, pp. 65–77. Springer Berlin Heidelberg, 7 2013.

[33] J. Mou, H. Ge, and J. Wang, "Weak average consensus of multi-agent system with partial flowing nodes:," *Transactions of the Institute of Measurement and Control*, vol. 39, pp. 0142331216634832–1224, 3 2016.

[34] H. Yu, G. Liu, and Y. Zhang, "Leader-following consensus of multi-agent in switching networks with time-delay," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, pp. 2976–2982, 6 2013.

[35] W. Pieprzyca, "Analysis of communication processes in the multi-agent systems," *Annales UMCS, Informatica*, vol. 9, pp. 111–122, 1 2009.

[36] K. Potiron, A. E. F. Seghrouchni, and P. Taillibert, *Multi-Agent System Properties*, pp. 5–10. Springer London, 3 2013.

[37] A. Al-Jumaily, *Multi-Agent System Concepts Theory and Application Phases*. Pro Literatur Verlag, Germany / ARS, Austria, 12 2006.

[38] D. P. Dash, "Improvement of co-ordination: re-scripting research for multi-agent environments," *Systems Research and Behavioral Science*, vol. 19, pp. 115–121, 3 2002.

[39] J. J. Pacheco-Reyes, O. Starostenko, V. Alarcon-Aquino, and J. Rodríguez-Asomoza, *Multi-agent Architecture for User Adaptive Information Retrieval Systems*, pp. 567–574. Germany: Springer International Publishing, 11 2014.

[40] 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.

[41] S. Zheng, Q. Zhang, R. Zheng, H. Biqin, S. Yilin, and X.-C. Chen, "Combining a multi-agent system and communication middleware for smart home control: A universal control platform architecture," *Sensors (Basel, Switzerland)*, vol. 17, pp. 2135–, 9 2017.

[42] G. K. T, K. Poornaselvan, and M. Sethumadhavan, "Fuzzy support vector machine-based multi-agent optimal path," *Defence Science Journal*, vol. 60, pp. 387–391, 5 2010.

[43] W. Ji, F. Wang, and P. Wu, *IFM - Synthesizing Coalitions for Multi-agent Games.*, pp. 243–259. Germany: Springer International Publishing, 8 2017.

[44] L. Meng, M. Lu, J. Lai, and X. Xu, "A multi-agent based approach to reliability prediction of trains control and monitoring software system," *International Journal of Digital Content Technology and its Applications*, vol. 6, pp. 149–157, 7 2012.

[45] J. Zhang, C. Xia, and W. Zhang, *ADMI - Accessory-Based Multi-agent Simulating Platform on the Web*, pp. 52–63. Germany: Springer International Publishing, 6 2015.

[46] D. A. Novikov, *Incentive Mechanisms for Multi-agent Organizational Systems*, pp. 35–57. United States: Springer International Publishing, 10 2015.

[47] V. Nissen and D. Saft, “Springsim - social emergence in organisational contexts: benefits from multi-agent simulations,” in *Proceedings of the 2010 Spring Simulation Multiconference*, pp. 10–8, Society for Computer Simulation International, 4 2010.

[48] M. d’Inverno, M. Luck, M. Fisher, and C. Preist, *Foundations and Applications of Multi-Agent Systems - Foundations and Applications of Multi-Agent Systems*. Germany: Springer Berlin Heidelberg, 7 2002.

[49] S. A. DeLoach, *Agent-Oriented Software Engineering - O-MaSE: An Extensible Methodology for Multi-agent Systems*. Springer Berlin Heidelberg, 3 2014.

[50] A. Torreño, E. Onaindia, and O. Sapena, “Fmap: Distributed cooperative multi-agent planning,” *Applied Intelligence*, vol. 41, pp. 606–626, 6 2014.

[51] 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.

[52] G. Aucher, “Generalizing agm to a multi-agent setting,” *Logic Journal of IGPL*, vol. 18, pp. 530–558, 1 2010.

[53] S. S. Chouhan and R. Niyogi, “Dimpp: a complete distributed algorithm for multi-agent path planning,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 29, pp. 1129–1148, 4 2017.

[54] L. Wang, Q. Li, L. Yishuai, and H. Chu, *ICSI (1) - Multi-agent Organization for Hiberarchy Dynamic Evolution*, pp. 570–577. Germany: Springer International Publishing, 6 2015.

[55] X. Dou, X. Quan, Z. Wu, M. Hu, K. Yang, J. Yuan, and M. Wang, “Hybrid multi-agent control in microgrids: Framework, models and implementations based on iec 61850,” *Energies*, vol. 8, pp. 31–58, 12 2014.

[56] M. Purvis, “Narrative structures for multi-agent interaction,” in *Proceedings. IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004. (IAT 2004).*, pp. 232–238, IEEE, 11 2004.

[57] T. C. Son, *LPNMR - Answer Set Programming and Its Applications in Planning and Multi-agent Systems*, pp. 23–35. Germany: Springer International Publishing, 6 2017.

[58] S. Mundle, N. Giri, A. Ray, and S. Bodhe, “Jade based multi agent system for mobile computing for cellular networks,” in *Proceedings of the International Conference on Advances in Computing, Communication and Control*, pp. 467–473, ACM, 1 2009.

[59] S. can Zhai, C. jun Hu, and W. ming Zhang, “Multi-agent distributed intrusion detection system model based on bp neural network,” *International Journal of Security and Its Applications*, vol. 8, pp. 183–192, 3 2014.

[60] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” 1 2017.

[61] D. Simões, N. Lau, and L. P. Reis, *EPIA - Multi-agent Double Deep Q-Networks*, pp. 123–134. Germany: Springer International Publishing, 8 2017.

[62] Y. Hong, X. Wang, and Z.-P. Jiang, “Distributed output regulation of leader–follower multi-agent systems,” *International Journal of Robust and Nonlinear Control*, vol. 23, pp. 48–66, 10 2011.

[63] V. Vijaykumar, R. Chandrasekar, and T. Srinivasan, “An obstacle avoidance strategy to ant colony optimization algorithm for classification in event logs,” in *2006 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1–6, 2006.

[64] H. Su, Y. Qiu, and L. Wang, “Semi-global output consensus of discrete-time multi-agent systems with input saturation and external disturbances.,” *ISA transactions*, vol. 67, pp. 131–139, 1 2017.

[65] T. Fu-xiao, G. Xinping, and L. Derong, “Consensus protocol for multi-agent continuous systems /,” *Chinese Physics B*, vol. 17, pp. 3531–3535, 10 2008.

[66] C. Zhang and V. Lesser, “Coordinated multi-agent reinforcement learning in networked distributed pomdps,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, pp. 764–770, 8 2011.

[67] P. M. Dolia, “Integrating ontologies into multi-agent systems engineering (mase) for university teaching environment,” *Journal of Emerging Technologies in Web Intelligence*, vol. 2, pp. 42–47, 2 2010.

[68] K. Whitebread, M. Hofmann, and L. A. Pridmore, “Multi-agent common operating environment (macoe),” 7 2002.

[69] D. Xu, R. A. Volz, T. R. Ioerger, and J. Yen, “Seke - modeling and verifying multi-agent behaviors using predicate/transition nets,” in *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, pp. 193–200, ACM, 7 2002.

[70] A. Claudi, P. Sernani, and A. F. Dragoni, “Towards multi-agent health information systems,” *International Journal of E-Health and Medical Communications*, vol. 6, pp. 20–38, 10 2015.

[71] E. A. Saci and Y. Cherruault, “The genicagent: a hybrid approach for multi-agent problem solving,” *Kybernetes*, vol. 30, pp. 26–34, 2 2001.

[72] L. Mo and T. Pan, “Mean-square consensus of heterogeneous multi-agent systemsunder markov switching topologies,” *SCIENTIA SINICA Informationis*, vol. 46, pp. 1621–1632, 11 2016.

[73] J. Zhou, C. Wang, and H. Qian, “Existence, properties and trajectory specification of generalised multi-agent flocking,” *International Journal of Control*, vol. 92, pp. 1434–1456, 11 2017.

[74] F. D. Priscoli, A. D. Giorgio, F. Lisi, S. Monaco, A. Pietrabissa, L. R. Celsi, and V. Suraci, “Multi-agent quality of experience control,” *International Journal of Control, Automation and Systems*, vol. 15, pp. 892–904, 1 2017.

[75] C. Ramachandran, S. Misra, and M. Obaidat, “On evaluating some agent-based intrusion detection schemes in mobile ad-hoc networks,” in *Proceedings of the SPECTS 2007*, (San Diego, CA), pp. 594–601, July 2007.

[76] X. Jiang, Z. Ji, T. Hou, and F. Sun, *Observability and Controllability Preservation for Multi-agent Systems with Time Delay and Time-Varying Topology*, pp. 313–325. Germany: Springer Singapore, 10 2017.

[77] N. Y. Mutovkina, V. N. Kuznetsov, and A. Klyushin, *The Formation of the Optimal Composition of Multi-Agent System*, pp. 293–302. Springer International Publishing, 8 2017.

[78] R. Gray, A. Franci, V. Srivastava, and N. E. Leonard, “Multi-agent decision-making dynamics inspired by honeybees,” 1 2017.