

Received: 30 June 2026; **Revised:** 19 August 2026; **Accepted:** 02 September 2026; **Published:** 30 September 2026

Archs Sci. (2026) Volume 76(3), Pages 51-58, Paper ID 2026307.
<https://doi.org/10.68304/as/76307>

Model-Driven Enterprise Economic Audit and Management Framework Based on Genetic Algorithms and Deep Learning Under Edge–Cloud Collaboration

Mingyue Quan

Business School, Luoyang Normal University, Luoyang 471934, China; quanmingyue0927@163.com

Abstract Enterprise economic audit and management increasingly require adaptive decision-support systems capable of operating with heterogeneous data, nonlinear constraints, distributed computing resources, and incomplete supervision. This study proposes a model-driven enterprise economic management framework that combines genetic-algorithm-based optimization, deep-learning decision support, weakly supervised learning, model transformation, and edge–cloud collaboration. The framework separates business-level models from platform-specific implementation through model-driven engineering and Query/View/Transformation (QVT)-style transformations, while Open Neural Network Exchange (ONNX)-compatible deployment is used to support model portability across heterogeneous execution environments. Genetic algorithms are employed for constrained resource-allocation and scheduling problems, whereas deep-learning models provide data-driven prediction and representation capabilities. Weak supervision and active learning are incorporated to reduce dependence on fully labeled enterprise datasets. At the infrastructure level, software-defined networking and stochastic network calculus support adaptive routing, distributed task placement, and probabilistic delay analysis across edge and cloud resources. The case-study evaluation reports a 32.6% improvement in data-flow efficiency after genetic-algorithm optimization, resource utilization of 91.6% for the GA-based strategy, and active-learning classification accuracy exceeding 63% when only 1% of the data are labeled. These results are treated as case-specific simulation outcomes rather than universal performance guarantees. The proposed framework contributes an integrated architecture for transforming enterprise economic models into deployable intelligent services while maintaining model traceability, resource efficiency, and responsiveness under distributed operating conditions.

Index Terms enterprise economic management, economic audit, genetic algorithm, deep learning, weak supervision, model-driven engineering, edge–cloud collaboration, stochastic network calculus

1. Introduction

Enterprise economic management is becoming increasingly data-intensive as firms operate within volatile markets, digitized supply chains, distributed information systems, and rapidly changing regulatory and operational environments. Traditional management frameworks based primarily on static rules, centralized processing, and manually specified decision procedures are often insufficient when resource constraints, demand conditions, network states, and financial signals change continuously. Research on China's digital economy likewise indicates that data-driven economic activity is reshaping industrial organization, governance, and value creation, thereby increasing the importance of adaptive analytical systems [1].

Machine learning has consequently become an important component of economic prediction and enterprise decision support. A systematic review of deep learning in economics documents expanding applications in forecasting, classifica-

tion, risk analysis, and high-dimensional economic modeling [2]. More recent work on economic forecasting and small- and medium-sized enterprise growth similarly emphasizes the value of ensemble learning, deep learning, and interpretability methods in data-driven economic decision-making [3]. Hybrid optimization is particularly attractive because prediction and decision variables frequently interact. Bedoui et al., for example, combine deep learning and genetic algorithms in a portfolio-optimization setting, illustrating the practical value of coupling predictive models with evolutionary search [4].

Genetic algorithms (GAs) are suitable for enterprise resource allocation because they can search large, nonlinear, constrained solution spaces without requiring a differentiable objective function. The broader evolutionary-computation literature documents their applicability to complex continuous optimization involving many variables, constraints, objectives, and dynamic conditions [5]. Evidence from adjacent scheduling and network-design domains also demonstrates the

flexibility of metaheuristic approaches: genetic and simulated-annealing methods have been used for urban transit scheduling [6]; GA and NSGA-II have been applied to multi-objective transit-network design [7]; cuckoo search has been used for sustainable transit-network design [8]; grasshopper optimization has been hybridized for vehicle routing [9]; and heuristic approaches have been developed for resource-constrained home-healthcare scheduling [10]. These studies do not constitute direct evidence for enterprise auditing, but they demonstrate that evolutionary and metaheuristic optimization can remain effective when scheduling, routing, resource, and objective constraints interact.

Deep learning provides a complementary capability. Rather than directly optimizing a combinatorial allocation, deep models learn predictive representations from historical or streaming data. This is useful when enterprises must estimate demand, classify risks, detect anomalous behavior, or forecast economic outcomes from heterogeneous inputs. The usefulness of machine learning across structurally different domains is also illustrated by applications in biomechanical risk prediction [11] and broader reviews of machine learning in biomaterials, biomechanics, and biofabrication [12]. These examples are outside enterprise management, but they demonstrate the transferability of data-driven representation and prediction methods across complex systems.

A major practical limitation is the scarcity of reliable labels. Enterprise records may be incomplete, inconsistently categorized, commercially sensitive, or expensive to annotate. Weakly supervised learning addresses settings in which supervision is incomplete, inexact, or inaccurate [13]. Active learning further reduces labeling cost by querying the most informative instances. Figure 1 illustrates the role assigned to active selection, model training, and validation in the proposed framework. Cross-validation and held-out testing are used to estimate predictive error rather than relying on distributional significance tests that are unrelated to the predictive objective.

The enterprise system must also be deployable. Model-driven engineering (MDE) addresses the transformation from abstract domain models to platform-specific implementations and executable artifacts. QVT is a standardized family of model-transformation languages, and QVT Relations provides a declarative mechanism for specifying relationships between models [14]. Model-driven engineering more generally emphasizes the use of explicit models and automated transformations to manage software complexity [15]. These ideas are useful for enterprise economic systems because business rules, audit entities, resource constraints, and deployment targets can evolve independently.

Deployment occurs increasingly in distributed edge–cloud environments. Edge computing can reduce latency, bandwidth use, and centralized processing pressure, while cloud resources provide elastic computation and model-management capacity. Qiu et al. identify routing, task scheduling, data analytics, security, load balancing, and offloading as central issues for edge-enabled industrial systems [16]. Software-defined networking (SDN) adds programmable network control and global traffic-management capabilities [17]. Distributed learning and routing research in software-defined vehicular networks further demonstrates how edge intelligence can support low-latency decision-making [18].

The present study integrates these strands into a model-driven enterprise economic audit and management framework. The principal research questions are: (1) how can business models be transformed into deployable intelligent components while preserving model consistency; (2) how can genetic algorithms and deep learning be combined for resource allocation and predictive decision support; (3) how can weak supervision improve robustness when labeled enterprise data are scarce; and (4) how can edge–cloud scheduling and stochastic network calculus support distributed execution under latency and resource constraints?

II. Related Work

A. Optimization, Scheduling, and Resource Allocation

Scheduling problems provide useful methodological analogies for enterprise management because they involve constrained allocation of limited resources over time. Delay-aware geographic routing has been used to improve communication performance in urban vehicular networks [19], while multi-AGV routing research combines pre-planning, real-time adjustment, conflict avoidance, and time-window constraints [20]. Emergency and recovery scheduling also illustrates the need to optimize under severe constraints. Cui et al. formulate a two-stage rescue and reconstruction scheduling problem after tunnel collapse [21], whereas Liu et al. develop resilience-oriented restoration schedules for transportation networks [22]. In each case, the optimization problem combines limited resources, temporal constraints, operational priorities, and system-level performance.

Uncertainty creates an additional challenge. Green road–rail intermodal routing has been formulated using fuzzy programming to incorporate uncertain service and scheduling

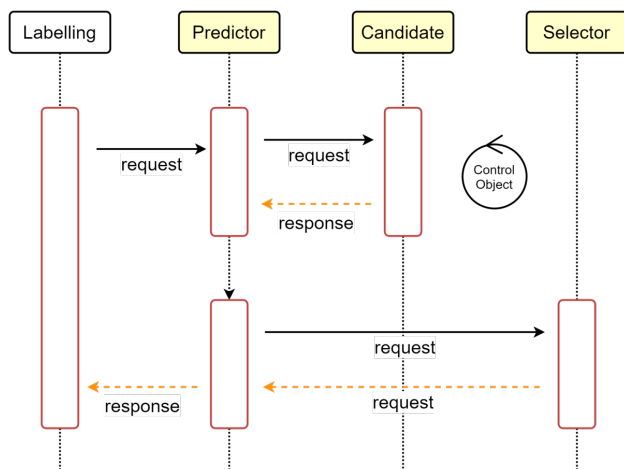


Figure 1: Active-Learning and Deep-Learning Workflow for Data Selection, Training, and Validation

conditions [23]. Such work is relevant to enterprise management because financial, logistical, and audit decisions frequently rely on uncertain demand, incomplete data, and fluctuating service capacity. The present framework therefore treats optimization as an adaptive process rather than a one-time deterministic calculation.

B. Deep Learning and Weak Supervision in Economic Decision Support

Deep learning has become increasingly important in economic applications because many enterprise relationships are non-linear and difficult to specify using fixed parametric forms. Zheng et al. review the strengths and limitations of deep learning in economics, including issues of interpretability, data requirements, and model selection [2]. Al-Karkhi and Rządowski extend this perspective to forecasting and SME growth prediction and emphasize the value of ensemble models and interpretability techniques [3].

The principal data challenge addressed in this study is limited supervision. Weakly supervised learning distinguishes incomplete, inexact, and inaccurate forms of supervision and provides a conceptual basis for learning when strong labels are unavailable [13]. Active learning can be layered on this setting by requesting labels selectively. The present framework therefore uses weak supervision not as a guarantee of high accuracy, but as a mechanism for maintaining usable prediction when complete enterprise labels cannot be obtained.

C. Model-Driven Engineering and Portable Deployment

Model transformation is central to the proposed architecture. QVT provides a standardized approach to transforming models defined at different abstraction levels [14]. In the present system, the platform-independent business model (PIM-BM) describes economic entities, relationships, audit rules, and decision constraints, while the platform-specific business-component model (PSM-BC) expresses deployable components, data interfaces, and execution requirements. Figure 2 summarizes this transformation logic. Figure 3 focuses on the mapping from business entities to platform-specific business objects. Entity types, attributes, keys, and relationships are transformed while preserving structural constraints.

Model portability is also important for trained neural networks. ONNX provides a widely used intermediate representation for exchanging trained models across frameworks and deployment toolchains; related work such as QONNX extends this representation for quantized neural networks [24]. In the present framework, ONNX-compatible export is used as a deployment abstraction rather than as part of the business-model transformation language.

D. Edge-Cloud Collaboration and Network Control

Distributed execution requires a network architecture capable of placing tasks close to data sources when low latency is important while still exploiting cloud resources for global optimization and intensive model training. Edge-computing

research in industrial environments emphasizes workload of flooding, routing, analytics, data sharing, and security as major architectural challenges [16]. SDN can improve network-wide traffic visibility and programmable control, and Guo and Yuan demonstrate the use of SDN and artificial intelligence for traffic optimization [17]. Distributed learning in software-defined Internet-of-Vehicles research similarly shows how edge intelligence can reduce routing-decision latency [18].

The network-performance layer in this study uses stochastic network calculus (SNC), not “Secure Network Communications.” SNC provides a mathematical framework for probabilistic traffic and service guarantees, including backlog and delay bounds [25]. This distinction is important because the mathematical treatment later in the paper concerns arrival processes, service processes, concatenation, and stochastic delay bounds.

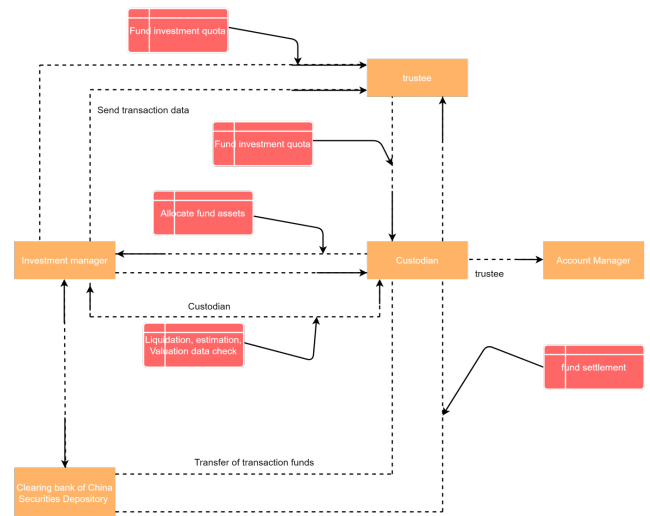


Figure 2: Model-Transformation Principle From a Platform-Independent Business Model to a Platform-Specific Business Component Model

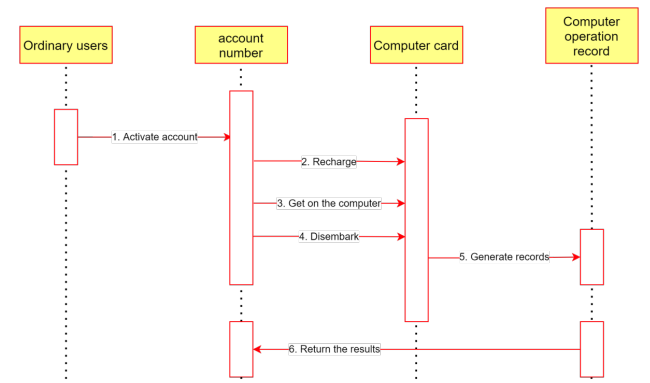


Figure 3: Transformation Relationship From Business Entities in PIM-BM to Business Objects in PSM-BC

III. Methods

A. Overall Framework

The proposed framework contains five interacting layers: (1) enterprise economic and audit models; (2) model transformation and synchronization; (3) predictive learning; (4) optimization and scheduling; and (5) edge-cloud execution. Business semantics are first represented at the PIM-BM level. QVT-style transformation rules convert these structures into PSM-BC components, after which executable services and model interfaces are generated. Predictive models are trained from enterprise data and exported to deployment-compatible formats. Genetic algorithms optimize resource allocation, scheduling, and model-placement decisions. The edge-cloud layer executes the resulting services subject to capacity, latency, and network constraints.

The architecture is designed to preserve traceability. A business entity or audit rule should remain identifiable after transformation into a platform-specific object or service. Similarly, a model version used in production should be linked to its training configuration and deployment target. This reduces the risk that enterprise policy changes, data-schema changes, or model updates become inconsistent across the system.

B. Model Transformation and Synchronization

A model transformation can be represented as a relation between a source-model domain and a target-model range. For a transformation relation R , each element should satisfy the declared source and target metamodel constraints. Nonredundancy requires that two relation elements with identical domain and range mappings represent the same transformation:

$$\forall e, f \in R, \quad \text{dom}(e) = \text{dom}(f) \wedge \text{ran}(e) = \text{ran}(f) \Rightarrow e = f.$$

This formulation is consistent with the declarative relation-oriented logic of QVT-style transformations [14].

Figure 2 illustrates the movement from a platform-independent model to a platform-related component model. The central methodological requirement is not simply syntactic translation; semantic consistency must be maintained between business concepts and executable components.

For relational persistence, entity transformation follows three cases. First, each entity type is converted into a relational schema in which entity attributes become relation attributes and entity identifiers become keys. Second, for a one-to-one relationship, the key and relationship attributes may be incorporated into either participating schema subject to integrity constraints. Third, for one-to-many and many-to-many relationships, foreign-key or separate relationship-schema representations are created as appropriate. Figure 3 presents this relationship graphically.

Model synchronization is version-aware. When a business model changes, transformation traces identify affected PSM-BC elements and executable components. Reverse or bidirectional consistency is desirable when platform-level implementation changes must be reconciled with higher-level models. This is one of the motivations for using relation-based transformation concepts rather than ad hoc code generation.

C. Deep-Learning and Weak-Supervision Layer

The learning layer supports forecasting, classification, anomaly identification, and decision support. In fully supervised settings, labeled historical observations can be used directly. When labels are incomplete, weak supervision and active learning are used to prioritize informative samples [13]. Figure 1 shows the interaction between active sample selection, training, and evaluation.

The framework separates predictive accuracy from optimization quality. A highly accurate predictor does not automatically generate an optimal resource-allocation decision, and an optimizer cannot compensate for systematically biased predictions. The predictive layer therefore exposes uncertainty and output scores to the optimization layer rather than collapsing all decisions into a single model.

The relevance of machine learning to economic forecasting is supported by the recent review literature [2], [3]. Hybridization with genetic algorithms is also established in financial optimization; Bedoui et al. combine deep learning and GA-based optimization within a portfolio model [4]. The current study generalizes the architectural idea to enterprise resource and audit decisions rather than reproducing that specific financial model.

D. Genetic-Algorithm Optimization

A chromosome represents a candidate allocation or scheduling plan. Genes can encode resource assignments, task order, model-placement decisions, or parameter choices. The initial population is evaluated using a fitness function that combines utilization, delay, cost, constraint violations, and application-specific priorities. Selection, crossover, and mutation generate new candidate populations until the termination criterion is met.

The choice of GA is supported by the broader evolutionary-computation literature [5]. Comparable metaheuristic optimization has been demonstrated in transport scheduling [6]–[8], vehicle routing [9], healthcare scheduling [10], and resilience planning [22]. These domains differ from enterprise economic management, but the common computational structure is constrained combinatorial or multi-objective optimization.

Distance metrics are used within clustering or similarity-based preprocessing. For two two-dimensional samples, Manhattan distance is

$$d_1 = |x_1 - x_2| + |y_1 - y_2|, \quad (1)$$

Euclidean distance is

$$d_2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \quad (2)$$

and Chebyshev distance is

$$d_\infty = \max(|x_1 - x_2|, |y_1 - y_2|). \quad (3)$$

The metric is selected according to the data geometry and the optimization task rather than assuming that Euclidean distance is universally preferable.

E. Edge–Cloud Task Scheduling

The MD Insider architecture is divided into model storage, model access, model definition, model transformation, and transformation-rule layers. Figure 4 shows these components and their relationships.

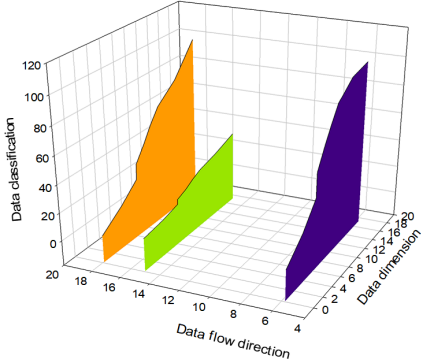


Figure 4: MD Insider Platform Architecture

The model-data storage layer maintains model versions and transformation artifacts. The access layer supports check-in, check-out, versioning, and retrieval. The definition layer stores the PIM-BM and PSM-BC metamodels. The transformation layer executes mappings among business models, platform-specific models, and code. The rule layer parses QVT-style relations and template-based generation rules.

Algorithm 1: Schematic Business-Flow Scheduling Procedure

Input: TT traffic set s , link set E , time slot set and path set

Output: number or rate of unscheduled time-triggered business flows

```

1: sum = 0
2: for  $\forall \lambda_k \in S$  do
3:   for  $\forall \varepsilon_{i,j} \in R_{\lambda_k}$  do
4:     Follow the constraint formula (41) to formula (43) to search the
       business flow  $\lambda_k$  transmission time on the link  $t$ 
5:   Mark the corresponding link time slot as occupied, record the time slot
       and link identifier, and add them to the scheduling record
6:   if  $t \leq \text{deadline} - \Delta t_{in} - \Delta t_x$ 
7:     continue
8:   else
9:     Restore the link-slot states modified during scheduling of this service
       flow to the unused state
10:  sum++
11: return sum
    
```

When the nearest edge node lacks sufficient capacity, tasks may be forwarded to another edge node or to the cloud. This creates a scheduling problem involving latency, bandwidth, resource utilization, and application priority. Research on delay-aware routing [19], multi-AGV scheduling [20], SDN optimization [17], and distributed edge learning [18] illustrates comparable problems in which local decisions interact with network-wide constraints.

Algorithm 1 summarizes the scheduling procedure retained in the proposed platform. Time-triggered business flows are assigned to feasible paths and link time slots; assignments that

violate deadlines are rolled back, and the unscheduled count is recorded.

F. Stochastic Network-Calculus Layer

SNC is used to characterize arrival and service processes and to obtain probabilistic backlog and delay bounds [25]. The analysis considers a through flow (TF) and interfering flows sharing service nodes. On each node, interference is aggregated; a residual service process is derived for the through flow; and per-node service processes are concatenated along the service chain. Figure 5 depicts the network scenario.

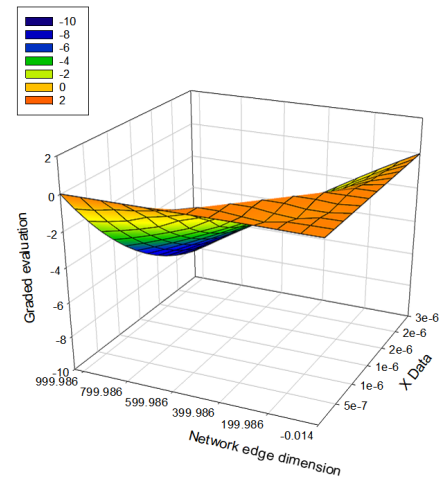


Figure 5: Mobile Edge–Cloud Network Structure

Figure 6 shows the analytical representation in which the z -flow is the through flow and the x - and y -flows represent interfering traffic. This abstraction permits delay analysis independently of a particular physical network implementation.

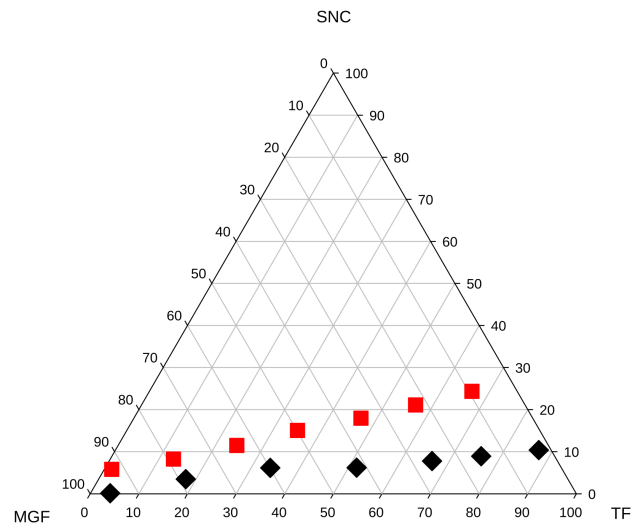


Figure 6: Stochastic Network-Calculus Model With Economic Data as the Through Flow

The use of stochastic service bounds is consistent with the framework’s broader objective of supporting uncertain enterprise environments. Related optimization studies in restoration planning [22] and green intermodal routing under uncertainty [23] similarly demonstrate that decision systems should represent uncertain or variable operating conditions explicitly.

IV. Case Study and Results

A. Integrated Enterprise Platform

The case study evaluates the proposed architecture as a model-driven rapid-development environment for enterprise economic management. The intended input is a platform-independent business model, and the outputs include a platform-specific component model, deployable code, and trained analytical services. The platform therefore couples MDE with data-driven models rather than treating software generation and machine learning as independent processes.

Figure 4 provides the overall platform structure, while Figures 2 and 3 show the transformation path at increasing levels of detail. Together, these figures demonstrate how business semantics move from an abstract model toward deployable components.

B. Genetic-Algorithm Data-Flow Optimization

The GA is used to optimize data-flow and resource-allocation decisions subject to system constraints. Figure 7 presents the reported data-flow behavior after GA optimization.

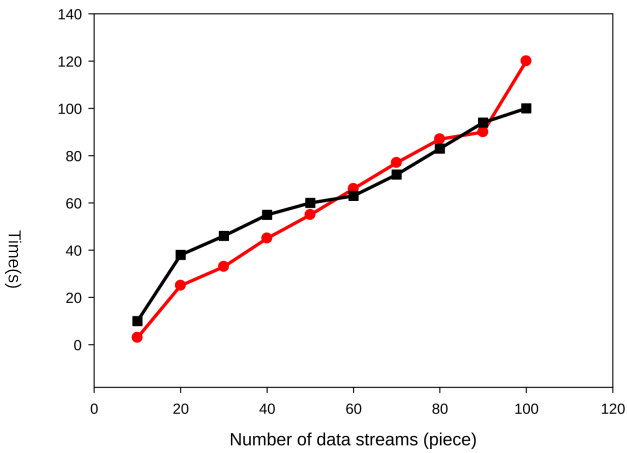


Figure 7: Data-Flow Quantity Under Genetic-Algorithm Optimization

The case results report a 32.6% improvement in data-flow efficiency relative to the baseline configuration. Because the supplied materials do not provide complete raw observations, confidence intervals, or repeated-run distributions, this value should be interpreted as a reported case-study improvement rather than a statistically generalized effect. The result is nevertheless consistent with the broader literature showing that evolutionary algorithms can improve complex allocation and scheduling solutions [5].

C. Weak-Supervision Robustness

The classification experiment evaluates prediction under limited labeled data. The integrated evaluation in Figure 8 indicates that active learning remains above 63% classification accuracy when only 1% of the data are labeled. This result supports the use of selective supervision in data-sparse enterprise contexts, although it does not establish superiority across all datasets or model families.

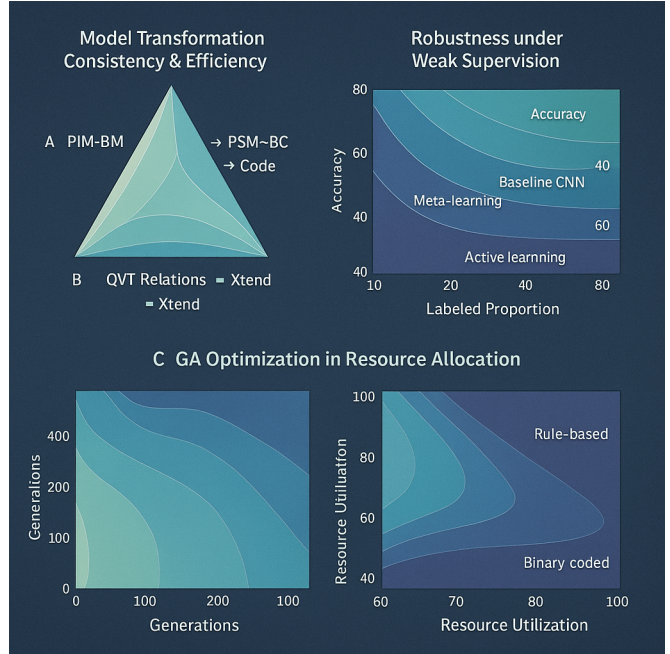


Figure 8: Integrated Evaluation of Model Transformation, Weak-Supervision Robustness, and Optimization Efficiency

Weak supervision is particularly relevant when audit labels require expert review. Rather than attempting to label all observations, the system can identify uncertain or informative records and request targeted annotation. This approach is consistent with the general weak-supervision taxonomy described by Zhou [13].

D. Resource Utilization

The final component of Figure 8 compares genetic-algorithm, linear-programming, and greedy strategies for resource utilization. The GA reaches a reported utilization rate of 91.6%, exceeding the alternatives in the supplied case. Again, the figure should be read as a scenario-specific comparison because detailed solver settings, repeated-run variance, and sensitivity analyses are not available.

The use of evolutionary optimization is nevertheless plausible in constraint-rich settings. Zhan et al. emphasize that contemporary evolutionary computation is particularly relevant for large-scale, dynamic, multi-objective, constrained, and expensive optimization problems [5]. The transportation and scheduling studies cited earlier provide additional evidence of the flexibility of metaheuristics in structurally related decision problems [6]–[10].

E. Integrated Evaluation

Figure 8 combines three reported performance dimensions: model-transformation time, classification robustness under reduced labeled data, and resource utilization under alternative optimization approaches.

The first panel compares QVT Relations and template-based transformation in local and cloud environments. Template-based transformation is reported as slightly faster, whereas QVT-based transformation is presented as more structurally consistent across platforms. This is compatible with the role of QVT as a declarative model-transformation approach [14], [15]. The second panel shows the weak-supervision comparison, and the third reports resource utilization.

These results suggest that the framework’s value lies in combining complementary mechanisms. Model-driven transformation supports traceability and deployment consistency; weak supervision addresses scarce labels; GA optimization addresses resource allocation; and edge–cloud scheduling supports distributed execution. No single component should be interpreted as sufficient on its own.

V. Discussion

A. Enterprise Relevance of the Hybrid Framework

The proposed architecture combines predictive, prescriptive, and engineering functions. Deep learning provides prediction, GA provides prescriptive optimization, MDE provides implementation traceability, and edge–cloud collaboration provides distributed execution. This division of responsibility is preferable to presenting a single “intelligent” model as responsible for all functions.

The economic relevance is supported by the growing use of deep learning in economic analysis [2], [3] and by hybrid predictive-optimization models in finance [4]. At the same time, digital-economy research emphasizes that data governance, privacy, data assets, and data-driven innovation are becoming central issues for enterprise and policy decision-making [1]. The present framework should therefore be understood as part of a wider shift toward data-intensive economic management.

B. Why the Uncited Optimization Literature Matters

Several references in the original bibliography were previously uncited. They are retained here only where their methodological relevance can be stated explicitly. Home-healthcare scheduling [10], vehicle routing [9], transit-network design [7], [8], emergency scheduling [21], AGV routing [20], and infrastructure restoration [22] are not enterprise-audit studies. Their relevance is computational: each addresses constrained allocation, route selection, resource scheduling, or resilience through heuristic or meta-heuristic methods. They therefore support the optimization rationale rather than claims about enterprise economics.

Similarly, SDN traffic optimization [17], software-defined vehicular routing [18], and delay-aware routing [19] are cited

in the network-control discussion because they demonstrate programmable and distributed routing principles. They are not presented as evidence of financial-management effectiveness.

The two biomechanics references [11], [12] are included only to illustrate that machine-learning representation and prediction methods can transfer across high-dimensional scientific domains. They are not used to justify enterprise-specific findings. This contextual placement prevents citation inflation and makes the evidential role of each source explicit.

C. Limitations

The study has several limitations. First, the reported performance values are based on simulation and case materials rather than a publicly available enterprise benchmark. The 32.6% data-flow improvement, 91.6% utilization rate, and weak-supervision accuracy should therefore be replicated using openly documented datasets, multiple random seeds, and sensitivity analyses.

Second, the GA configuration is not fully characterized in the source material. Future work should report population size, selection method, crossover and mutation rates, stopping criteria, objective weights, constraint penalties, and computational cost. Comparisons should include additional baselines, particularly exact optimization where tractable and modern evolutionary or swarm algorithms where appropriate.

Third, the deep-learning component requires a clearer description of architecture, training procedure, label quality, and evaluation protocol. Weak supervision can reduce labeling requirements but does not eliminate the risk of bias or label noise [13]. Enterprise use should therefore include human review and explainability mechanisms.

Fourth, ONNX-compatible deployment improves portability but does not automatically guarantee numerical equivalence, latency, or operator support across all runtimes. Deployment validation should compare outputs before and after transformation. QONNX research illustrates how representation standards may require extensions for specialized model formats such as low-precision networks [24].

Finally, SNC depends on assumptions about arrival and service processes. Delay bounds are only as meaningful as those assumptions. Empirical traffic characterization and validation are therefore necessary before the analytical guarantees are used in operational service-level agreements [25].

VI. Conclusion

This study proposes a model-driven enterprise economic audit and management framework integrating genetic algorithms, deep learning, weak supervision, model transformation, and edge–cloud collaboration. The architecture separates business semantics, predictive analytics, resource optimization, deployment transformation, and distributed execution while connecting them through explicit model and data interfaces.

The revised framework clarifies several important technical points. QVT-style transformation is used for model-driven engineering rather than as a learning algorithm; ONNX-compatible representation is used for neural-model porta-

bility; weak supervision and active learning address limited labels; SDN supports programmable network control; and stochastic network calculus provides probabilistic network-performance analysis. These components therefore have distinct roles within the system.

The case study reports three principal performance observations: a 32.6% improvement in data-flow efficiency after GA optimization, active-learning accuracy above 63% with only 1% labeled data, and GA-based resource utilization of 91.6%. These results support the feasibility of the integrated design but should be interpreted as case-specific findings pending reproducible benchmarking.

Future research should evaluate the framework using real enterprise audit and economic-management datasets, compare alternative optimization algorithms, document model and deployment parameters in full, and examine interpretability, privacy, and governance. Additional work should also investigate adaptive edge–cloud placement under changing workloads and integrate economic cost directly into the scheduling objective. With these extensions, model-driven engineering and intelligent optimization can provide a practical foundation for responsive, auditable, and scalable enterprise decision-support systems.

Funding

The author declares that no specific external funding information was provided for this study.

Conflict of Interest

The author declares no conflict of interest.

Data Availability

The experimental and simulation data supporting the findings of this study are available from the author upon request.

Ethics Statement

Not applicable.

References

- [1] Xu, Y., Tao, Y., Zhang, C., Xie, M., Li, W., & Tai, J. (2022). Review of digital economy research in China: A framework analysis based on bibliometrics. *Computational Intelligence and Neuroscience*, 2022, 2427034.
- [2] Zheng, Y., Xu, Z., & Xiao, A. (2023). Deep learning in economics: A systematic and critical review. *Artificial Intelligence Review*, 56(9), 9497–9539.
- [3] Al-Karkhi, M. I., & Rządowski, G. (2025). Innovative machine learning approaches for complexity in economic forecasting and SME growth: A comprehensive review. *Journal of Economy and Technology*, 3, 109–122.
- [4] Bedoui, R., Benkraiem, R., Guesmi, K., & Kedidi, I. (2023). Portfolio optimization through hybrid deep learning and genetic algorithms Vine Copula–GARCH–EVT–CVaR model. *Technological Forecasting and Social Change*, 197, 122887.
- [5] Zhan, Z.-H., Shi, L., Tan, K. C., & Zhang, J. (2022). A survey on evolutionary computation for complex continuous optimization. *Artificial Intelligence Review*, 55(1), 59–110.
- [6] Tang, J., Yang, Y., & Qi, Y. (2018). A hybrid algorithm for urban transit schedule optimization. *Physica A: Statistical Mechanics and Its Applications*, 512, 745–755.
- [7] Almasi, M. H., Oh, Y., Sadollah, A., Byon, Y. J., & Kang, S. (2021). Urban transit network optimization under variable demand with single and multi-objective approaches using metaheuristics: The case of Daejeon, Korea. *International Journal of Sustainable Transportation*, 15(5), 386–406.
- [8] Sadeghi, M., Aghayan, I., & Ghaznavi, M. (2021). A cuckoo search based approach to design sustainable transit network. *Transportation Letters*, 13(9), 635–648.
- [9] Soto-Mendoza, V., García-Calvillo, I., Ruiz-y-Ruiz, E., & Pérez-Terrazas, J. (2020). A hybrid grasshopper optimization algorithm applied to the open vehicle routing problem. *Algorithms*, 13(4), 96.
- [10] Fathollahi-Fard, A. M., Hajiaghayi-Keshetli, M., & Mirjalili, S. (2020). A set of efficient heuristics for a home healthcare problem. *Neural Computing and Applications*, 32(10), 6185–6205.
- [11] Han, R., Qi, F., Wang, H., & Yi, M. (2024). Innovative machine learning approach for analysing biomechanical factors in running-related injuries. *Molecular & Cellular Biomechanics*, 21(3), 530.
- [12] Wu, C., Xu, Y., Fang, J., & Li, Q. (2024). Machine learning in biomaterials, biomechanics/mechanobiology, and biofabrication: State of the art and perspective. *Archives of Computational Methods in Engineering*, 31(7), 3699–3765.
- [13] Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National Science Review*, 5(1), 44–53.
- [14] Kurtsev, I. (2008). State of the art of QVT: A model transformation language standard. In A. Schürr, M. Nagl, & A. Zündorf (Eds.), *Applications of Graph Transformations with Industrial Relevance*, Lecture Notes in Computer Science, Vol. 5088 (pp. 377–393). Springer.
- [15] Schmidt, D. C. (2006). Guest editor's introduction: Model-driven engineering. *Computer*, 39(2), 25–31.
- [16] Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., & Wu, D. O. (2020). Edge computing in Industrial Internet of Things: Architecture, advances and challenges. *IEEE Communications Surveys & Tutorials*, 22(4), 2462–2488.
- [17] Guo, A., & Yuan, C. (2021). Network intelligent control and traffic optimization based on SDN and artificial intelligence. *Electronics*, 10(6), 700.
- [18] Lin, K., Li, C., Li, Y., Savaglio, C., & Fortino, G. (2021). Distributed learning for vehicle routing decision in software defined Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(6), 3730–3741.
- [19] Chen, C., Liu, L., Qiu, T., Wu, D. O., & Ren, Z. (2019). Delay-aware grid-based geographic routing in urban VANETs: A backbone approach. *IEEE/ACM Transactions on Networking*, 27(6), 2324–2337.
- [20] Hu, Y., Dong, L., & Xu, L. (2020). Multi-AGV dispatching and routing problem based on a three-stage decomposition method. *Mathematical Biosciences and Engineering*, 17(5), 5150–5172.
- [21] Cui, H., Liu, L., Yang, Y., & Zhu, M. (2022). A two-stage scheduling model for the tunnel collapse under construction: Rescue and reconstruction. *Energies*, 15(3), 743.
- [22] Liu, K., Zhai, C., & Dong, Y. (2021). Optimal restoration schedules of transportation network considering resilience. *Structure and Infrastructure Engineering*, 17(8), 1141–1154.
- [23] Sun, Y., Yu, N., & Huang, B. (2022). Green road–rail intermodal routing problem with improved pickup and delivery services integrating truck departure time planning under uncertainty: An interactive fuzzy programming approach. *Complex & Intelligent Systems*, 8(2), 1459–1486.
- [24] Pappalardo, A., Umuroglu, Y., Blott, M., Mitrevski, J., Hawks, B., Tran, N., Loncar, V., Summers, S., Borrás, H., Muhizi, J., Trahms, M., Hsu, S.-C., Hauck, S., & Duarte, J. (2022). QONNX: Representing arbitrary-precision quantized neural networks. *arXiv preprint arXiv:2206.07527*.
- [25] Jiang, Y., & Liu, Y. (2008). *Stochastic Network Calculus*. Springer.

...