

Article

Not peer-reviewed version

Self-Regulating, Knowledge-Driven Distributed Software Systems: Mindful Machines

[Rao Mikkilineni](#)^{*} and [W. Patrick Kelly](#)

Posted Date: 22 October 2025

doi: 10.20944/preprints202510.0020.v2

Keywords: mindful machines; general theory of information; artificial intelligence; digital genome; auto-poietic systems; cognitive architecture; ethical AI; Burgin-Mikkilineni thesis



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Self-Regulating, Knowledge-Driven Distributed Software Systems: Mindful Machines

Rao Mikkilineni * and W. Patrick Kelly

Center for Business Innovation, Golden Gate University, San Francisco, USA

* Correspondence: rmikkilineni@ggu.edu

Abstract

Large language models and agentic AI frameworks achieve strong performance but remain limited by statistical reasoning, shallow memory, and the absence of intrinsic self-regulation. These limitations restrict transparency, resilience, and adaptability in real-world distributed environments. We introduce Mindful Machines, a new computational paradigm that integrates self-regulation and meta-cognition into distributed software systems. The proposed Autopoietic and Meta-Cognitive Operating System (AMOS) is guided by a Digital Genome encoding goals, policies, and ethical constraints. AMOS enables autopoietic behaviors—self-deployment, self-healing, and self-scaling—while maintaining semantic and episodic memory in a graph database and leveraging cognizing oracles for validated transparent reasoning. To demonstrate feasibility, we re-implement the well-known credit-default prediction problem as a distributed application composed of containerized services orchestrated by AMOS across heterogeneous cloud infrastructures. Compared with a conventional monolithic ML pipeline, the prototype exhibits three key improvements: i. Explainable via event-driven history and auditable decision trails, ii. real-time adaptability to behavioral changes without retraining, iii. resilience through automated fault recovery and elastic scaling. The results highlight that Mindful Machines provide a scalable architecture for knowledge-centric, ethically aligned, and sustainable AI. By uniting the computer and the computed into a self-regulating whole, this paradigm advances transparent and trustworthy distributed software systems.

Keywords: mindful machines; general theory of information; artificial intelligence; digital genome; auto-poietic systems; cognitive architecture; ethical AI; Burgin-Mikkilineni thesis

1. Introduction

The rapid progress of generative AI and large language models (LLMs) has redefined machine intelligence benchmarks, producing outputs often indistinguishable from those of humans. Transformer architectures and retrieval-augmented pipelines now provide contextual awareness and access to vast information sources, giving experts tools no single individual could fully master [1–5]. Yet, despite their utility, these systems remain fundamentally constrained: they operate through statistical correlations, lack persistent memory, and are guided by no intrinsic purpose or ethical framework [6–10]. As a result, they cannot autonomously distinguish truth from falsehood, meaning from noise, or align their operations with system-level goals.

Agentic AI frameworks attempt to overcome these limits by wrapping LLMs with memory, planning, and feedback loops, enabling them to execute multi-step tasks [11,12]. While this increases utility, fragility persists. As the number of interacting agents grows, so does the risk of misalignment, error cascades, and uncontrolled feedback [13]. The central question remains unresolved: who supervises and governs these agents? Without intrinsic oversight, agentic AI risks presenting the appearance of autonomy while depending heavily on human managers or orchestration software.

Biological systems suggest a different path. A multicellular organism is itself a society of agents—cells—that achieve coherent function through communication, feedback loops, and genomic instructions that encode shared purposes such as survival and reproduction. This intrinsic teleonomy

prevents chaos and sustains resilience [14–16]. For AI, the analogy is clear: a swarm of software agents is insufficient without system-wide memory, goals, and self-regulation. Intelligent beings, like biological systems, require a persistent blueprint to align local actions with global purpose.

Theoretical work in information science and computation provides foundations for such a blueprint. Burgin’s General Theory of Information (GTI) expands information beyond Shannon’s syntactic model to include both ontological (structural) and epistemic (knowledge-producing) aspects [17–22]. Building on GTI, the Burgin–Mikkilineni Thesis (BMT) identifies a key limitation of the Turing paradigm—the separation of the computer from the computed [23,24]. BMT proposes structural machines, uniting runtime execution with knowledge structures to enable self-maintaining and adaptive computation. Deutsch’s epistemic thesis complements this by defining genuine knowledge as extendable, explainable, and discernible—criteria absent from today’s black-box AI [25]. Finally, Fold Theory (Hill) emphasizes recursive coherence between observer and observed, reinforcing the idea that computation must integrate structure, meaning, and adaptation into a single evolving process [26–28].

Together, these perspectives point to a new model of computation in which information, knowledge, and purpose are intrinsic to the act of computing. Building on this synthesis, we introduce Mindful Machines: distributed software systems that integrate autopoietic (self-regulating) and meta-cognitive (self-monitoring) behaviors. Their operation is guided by a Digital Genome—a persistent knowledge-centric code that encodes functional goals, policies, and ethical constraints [29].

To operationalize this vision, we present the Autopoietic and Meta-Cognitive Operating System (AMOS). Unlike conventional orchestration platforms, AMOS provides:

1. Self-deployment, self-healing, and self-scaling via redundancy and replication across heterogeneous cloud infrastructures.
2. Semantic and episodic memory implemented in graph databases to preserve both rules/ontologies and event histories.
3. Cognizing Oracles that validate knowledge and regulate decision-making with transparency and explainability.

We evaluate this architecture through a case study: re-implementing the classic credit-default prediction task discussed in detail in Klosterman’s book [30] as a distributed service network managed by AMOS. Whereas conventional ML pipelines process static datasets with limited adaptability, our implementation demonstrates resilience under failure, auditable decision provenance, and real-time adaptation to behavioral changes.

The contributions of this paper are fourfold:

- i. We ground the design of Mindful Machines in a synthesis of GTI, BMT, Deutsch’s epistemic thesis, and Fold Theory.
- ii. We describe the architecture of AMOS and its integration of autopoietic and meta-cognitive behaviors.
- iii. We demonstrate feasibility by deploying a cloud-native loan default prediction system composed of containerized services.
- iv. We show how this paradigm advances transparency, adaptability, and resilience beyond conventional machine learning pipelines.

In doing so, this paper positions Mindful Machines as a pathway toward sustainable, knowledge-driven distributed software systems—bridging the gap between symbolic reasoning, statistical learning, and trustworthy autonomy.

In section 2, we describe the theory and in section 3, we discuss the system architecture and the implementation of the platform. In section 4, we describe the implementation of the loan default prediction application and in Section 5, we discuss the results. Section 6 concludes with our observations, comparison of the two approaches.

2. Theories and Foundations of the Mindful Machines

The architecture of Mindful Machines represents a paradigm shift away from conventional, statistically centered structural correlation based artificial intelligence (AI). This new computational model necessitates deep theoretical grounding to justify its architectural change and unique operational features, such as intrinsic self-regulation and semantic transparency. This section provides an in-depth analysis of the four foundational theories—General Theory of Information (GTI), the Burgin–Mikkilineni Thesis (BMT), Deutsch’s Epistemic Thesis (DET), and Fold Theory (FT)—demonstrating how their synthesis logically compels the development of the Autopoietic and Meta-Cognitive Operating System (AMOS).

The limitations of the Turing paradigm—where computation is treated as blind, closed symbol manipulation—become apparent when systems require adaptability, resilience, and genuine explainability from deep learning algorithms. The transition to a new computing model is driven by four theoretical pillars, which collectively redefine information, knowledge, and the relationship between the computer and the computed.

2.1. General Theory of Information (GTI)

Mark Burgin’s General Theory of Information (GTI) provides the initial fundamental conceptual adjustment necessary for Mindful Machines, broadening the scope of information far beyond the statistical and syntactic definitions provided by Shannon. GTI is a profound, foundational theory built upon an axiomatic base and robust mathematical models, including information algebras and operator models based on category theory. It defines information as a triadic relation involving a carrier, a recipient, and the content conveyed.

The core contribution of GTI is the explicit distinction between two inseparable aspects of information: ontological information and epistemic information. Ontological information refers to the intrinsic structural properties and dynamics inherent in material systems, obeying scientific or ideal laws. Epistemic information, conversely, is the knowledge produced when these ontological structures are observed and interpreted by a cognitive apparatus.

For the design of cognitive software systems, GTI positions information not as a passive data stream, but as an active principle of organization. This perspective establishes that knowledge is a more advanced structural state than raw data, created through the transformation of external structure (ontological) into internal content (epistemic).

The requirement for continuous self-monitoring and adaptation in Mindful Machines mandates that systems must actively manage knowledge, not just process data. This necessity directly justifies the architectural decision within AMOS to incorporate the Semantic Memory Layer. This layer captures ontologies, rules, and policies, providing the persistent organizational structure required to handle meaningful knowledge flows derived from the GTI framework. By leveraging these stored knowledge structures, the system moves toward knowledge-centric efficiency. This architectural shift is anticipated to reduce the reliance on high-volume, repetitive statistical processing common in traditional AI, leading toward dynamic energy proportionality and greater sustainability, aligning computation with the energy efficiency found in biological systems.

2.2. Burgin–Mikkilineni Thesis (BMT)

Building directly upon the ontological and epistemic principles established by GTI, the Burgin–Mikkilineni Thesis (BMT) addresses the fundamental architectural flaw in classical computation: the rigid separation of the computer (the executing hardware and runtime) from the computed (the data, processes, and symbols). In the conventional Turing machine model, computation is necessarily closed, context-free, and lacks semantic awareness, rendering it incapable of intrinsic self-maintenance or contextual adaptation.

BMT proposes the concept of Structural Machines, which mandate that the computer and the computed must exist as a dynamically evolving unity. This unification is crucial because it enables

systems to perform autopoietic computation—the capacity to sustain, regenerate, and adapt their own internal structure. BMT is an axiological thesis, arguing that efficient autopoietic and cognitive behavior must utilize these structural machines. This movement allows digital systems to evolve from merely symbolic manipulation to structural maintenance, reaching the desired autopoietic tier of operational intelligence.

The engineering realization of the BMT's structural machine requirement is the Digital Genome (DG). The DG is a persistent, knowledge-centric blueprint that encodes the system's goals, structural rules, and ethical constraints. It functions as the system's essential self-knowledge, providing the necessary internal blueprint for identity and purpose. This structural encoding is the prerequisite for the autopoietic behaviors demonstrated by AMOS, such as self-deployment, self-healing, and elastic scaling.

The dissolution of the boundary between machine and meaning, necessitated by BMT, provides significant benefits in complex, distributed environments. The intrinsic self-regulation afforded by the structural machine framework provides resilience and local autonomy, which has been observed to circumvent limitations often encountered in systems constrained by the CAP theorem. Therefore, the AMOS architecture, particularly its Autopoietic Layer, is theoretically hardened against fragility because the distributed services are intrinsically guided by the DG to maintain global coherence, transcending the dependency on fragile external orchestration typical of conventional cloud-native systems.

2.3. Deutsch's Epistemic Thesis (DET)

David Deutsch's work establishes an essential quality standard for knowledge, linking epistemology, physics, computation, and evolution. Deutsch argues that genuine knowledge is defined by its properties of being extendable, explainable, and discernible. This framework posits that truth-seeking is fundamentally an error-correcting mechanism centered around the formation of good explanations.

This epistemic perspective acts as a direct critique of current machine learning and statistical models. These models, while powerful in producing correlations and predictions, often fail to meet Deutsch's standards because their outputs are generally opaque, lacking transparent justifications grounded in understandable structure—the classic “black box” problem.

Given that Mindful Machines are intended for high-stakes, ethically sensitive domains such as finance, healthcare, and cybersecurity, transparency must be elevated from a mere feature to an epistemological mandate. This theoretical necessity is realized within AMOS through the Cognizing Oracles and Episodic Memory. The Oracles enforce policy compliance and validate knowledge, ensuring that the system adheres to structured rules and ethical constraints encoded in the DG. Episodic Memory records event-driven histories, providing auditable decision trails. This integration ensures that every system output can be traced back through causal chains, transforming statistical predictions into genuine, discernible knowledge assets that support trustworthiness and regulatory alignment.

2.3. Fold Theory (FT)

Skye L. Hill's Fold Theory offers the metaphysical and dynamical framework necessary to operationalize the continuous evolution required by GTI and BMT. FT provides a philosophical lens emphasizing recursive coherence—the idea that ontological structures (the reality of the environment) and epistemic structures (the system's knowledge of that environment) are inherently “folded together” in shaping and sustaining reality.

This categorical framework suggests that structural integrity, even in complex systems, emerges from continuous, localized recursive processes. Time and fields are not static backgrounds but emergent features of coherence maintenance, utilizing principles like coherence sheaves to unify local consistency into a global, stable form. FT reinforces the view that computation must be a dynamic, evolving process that integrates structure, meaning, and continuous adaptation.

The Fold Theory’s emphasis on continuous, recursive adaptation dictates the need for active self-maintenance managers within the AMOS architecture. While BMT dictates the necessity of the structural machine, FT dictates the operational mechanisms by which that machine maintains integrity under continuous change. The concept of the “Autopoietic Fold” further links self-organization to a dynamically changing topology. This dynamic requirement justifies the core components of the Autopoietic Layer (the APM, CNM, and SWM), whose collective function is to guarantee that local structural changes—such as fault recovery, elastic scaling, or dynamically rewiring connections—immediately integrate into global system coherence, preventing systemic chaos in a distributed society of agents.

2.4. Beyond Turing Computing and Blackbox

Mindful Machines are fundamentally characterized as a Super-symbolic Computational paradigm, where computation is redefined as an active, self-organizing, and living process, guided by intrinsic knowledge and purpose. This profound shift is not arbitrary; it is the direct and necessary outcome of synthesizing the four foundational theories. The theoretical pillars provide a comprehensive, four-part mandate for system design:

1. GTI defines the universe of discourse: Information must be treated as structural and meaningful (Ontological/Epistemic duality).
2. BMT prescribes the required machine type: Structural Machines that enforce self-regulation by unifying computer and computed (Autopoiesis).
3. DET imposes the quality standard: All knowledge derived must be transparent, auditable, and extendable (Meta-Cognition/Explainability).
4. FT ensures the operational dynamics: Continuous, recursive coherence must be maintained for dynamic adaptation and resilience.

This theoretical convergence forces the integration of persistent structural knowledge (the Digital Genome and Semantic Memory), audited decision history (Episodic Memory), and continuous self-maintenance (Autopoietic Layer) into the AMOS framework. The resulting architecture is not merely an assemblage of contemporary technologies, but a necessary engineering consequence of adhering to this advanced theoretical foundation, resulting in a system that is inherently resilient, explainable, and sustainable.

Table 1. Four pillars of theoretical foundation.

Theoretical Pillar	Core Conceptual Shift/Mandate	Limitation of Turing Paradigm Overcome	Direct Architectural Realization in AMOS
General Theory of Information (GTI)	Information is an active principle of organization, defined by Ontological–Epistemic duality.	Blindness to <i>Meaning</i> and <i>Structure</i> in favor of syntax (Shannon model).	Semantic Memory Layer and knowledge schema design, ensuring persistent organizational structure.
Burgin–Mikkilineni Thesis (BMT)	Necessity of Structural Machines ; Unification of computer and computed (Autopoiesis).	Separation of execution (hardware) from knowledge (software); inability to self-modify.	The Digital Genome (DG) , encoding the system’s self-knowledge, goals, and ethical constraints.
Deutsch’s Epistemic Thesis (DET)	Knowledge must be Explainable, Discernible, and Extendable (Epistemic Quality Control).	Reliance on opaque correlations (black-box AI); lack of auditable causal reasoning.	Cognizing Oracles and persistent Episodic Memory (event-driven history and auditable decision trails).
Fold Theory (FT)	Recursive Coherence as the mechanism for structural integrity and adaptation.	Static definition of structure; inability to adapt continuously to emergent reality and environment changes.	Autopoietic Layer (APM, CNM, SWM), guaranteeing dynamic self-healing, resilience, and global coherence.

The table explicitly maps these abstract theoretical requirements to their concrete engineering realization within the AMOS architecture, demonstrating the rigor and necessity underpinning the Mindful Machines paradigm.

3. System Architecture and the Autopoietic and Meta-Cognitive Operating System (AMOS)

The Autopoietic and Meta-Cognitive Operating System (AMOS) provides the execution environment for Mindful Machines. Unlike a monolithic operating system, AMOS functions as a distributed orchestration and regulation platform that instantiates, coordinates, and sustains services derived from the Digital Genome (DG). The Digital Genome acts as a machine-readable blueprint of operational knowledge. It encodes functional goals, non-functional requirements, best-practice policies, and global constraints. This blueprint seeds a knowledge network that persists in a graph database, structured as both associative memory and event-driven history.

The application design process begins with the Digital Genome (DG). The DG acts as the application designer’s interface, where the desired functional entities—decomposed into containerized services (cognitive ‘cells’)—are specified using processes with inputs, behaviors, and outputs. Crucially, the designer uses the DG to create a ‘knowledge network configuration’ by connecting these functional entities with shared process knowledge and best-practice policies. This configuration, specifying both the nodes and their interdependencies, is then consumed by AMOS. The Autopoietic Process Manager (APM) within AMOS uses this DG blueprint in the form of a schema of operational knowledge to manage the deployment of these knowledge structures. The APM specifically interfaces with IaaS and PaaS resources from cloud providers, using their self-service facilities to deploy the containerized services and determine their resource allocation and initial network wiring. The Cognitive Network Manager (CNM) then translates the DG’s knowledge network configuration into the dynamic service connections required for the functional workflow.

Figure 1 illustrates the layered architecture of AMOS: Digital Genome Layer – Encodes goals, schemas, and policies, providing the prior knowledge blueprint for system behavior. Autopoietic Layer (AMOS Core) – Implements resilience and adaptation through core managers: APM (Autopoietic Process Manager): Deploys services, replicates them when demand rises, and guarantees recovery after failures. CNM (Cognitive Network Manager): Manages inter-service connections, rewiring workflows dynamically. SWM (Software Workflow Manager): Ensures execution integrity, detects bottlenecks, and coordinates reconfiguration. Policy & Knowledge Managers: Interpret DG rules, enforce compliance, and ensure traceability. Meta-Cognitive Layer – Cognizing Oracles oversee workflows, monitor inconsistencies, validate external knowledge, and enforce explainability. Application Services Layer – A network of distributed services (cognitive “cells”) that collaborate hierarchically to execute domain-specific functions. Knowledge Memory Layer – Maintains long-term learning context through: Semantic Memory: Rules, ontologies, and encoded policies. Episodic Memory: Event-driven histories that capture interactions and causal traces.

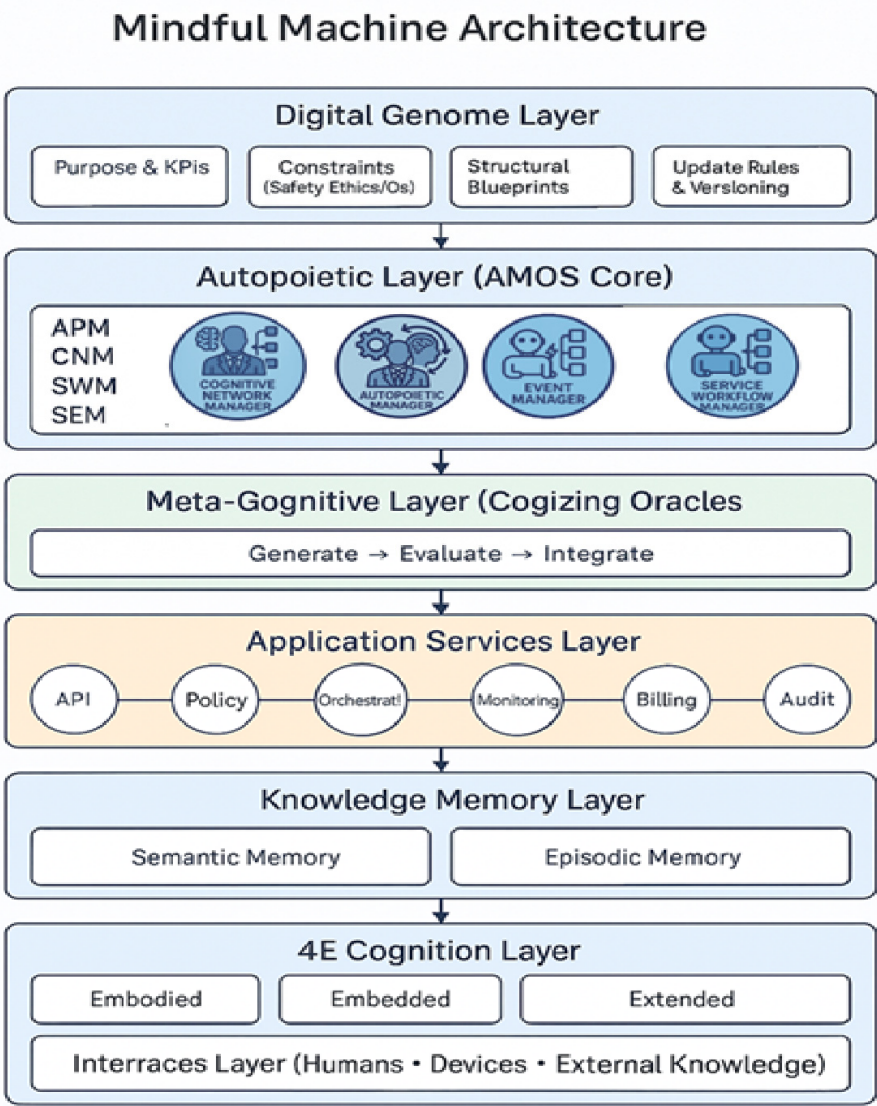


Figure 1. Mindful Machine Architecture.

3.1. Service Behavior and Global Coordination

The core mechanisms for self-regulation are explicitly defined by the AMOS Layer managers:

- **Autopoietic Process Manager (APM):** This manager is responsible for the system’s structural integrity. It initializes the system, determining the placement and resource allocation for containerized services across heterogeneous cloud infrastructure (IaaS/PaaS) based on DG constraints. Crucially, the APM ensures resilience by actively executing automated fault recovery and elastic scaling—replicating services when demand rises or guaranteeing recovery after failures.
- **Cognitive Network Manager (CNM):** The CNM governs the system’s dynamic coherence. It manages the inter-service connections based on the DG’s knowledge blueprint and is capable of dynamically rewiring workflows under changing conditions. Together with the APM, it configures the network structure to maintain Quality of Service (QoS) using policy-driven mechanisms like auto-failover or live migration, guided by Recovery Time Objective (RTO) and Recovery Point Objective (RPO) metrics.
- **Software Workflow Manager (SWM):** The SWM ensures the logical flow of computation. It monitors execution integrity, detects bottlenecks, and coordinates necessary reconfigurations, linking the state changes of services back to the system’s overall structural coherence.

- **Policy Manager:** This manager interprets DG rules, enforcing compliance and ensuring traceability. For instance, if service response time exceeds a specified threshold, the Policy Manager directs the APM to trigger auto-scaling to reduce latency, utilizing best-practice policies derived from the system’s history.

Each service in AMOS behaves like a cognitive cell with the following properties:

- **Inputs/Outputs:** Services consume signals, events, or data, and produce results, insights, or state changes guided by DG knowledge.
- **Shared Knowledge:** Services update semantic and episodic memory, ensuring global coherence, similar to biological signaling pathways.
- **Sub-networks:** Services form functional clusters (e.g., billing or monitoring), analogous to specialized tissues.
- **Global Coordination:** Sub-networks are orchestrated by AMOS managers to ensure system-wide goals are preserved.

Key properties of this design include:

- **Local Autonomy:** Services act independently, improving resilience.
- **Global Coherence:** Shared memory and DG constraints ensure alignment.
- **Evolutionary Learning:** Services adapt to improve efficiency and workflows.

Collectively, these functions enable self-deployment, self-healing, self-scaling, knowledge-centric operation, and traceability, distinguishing AMOS from conventional orchestration systems (Figure 2).

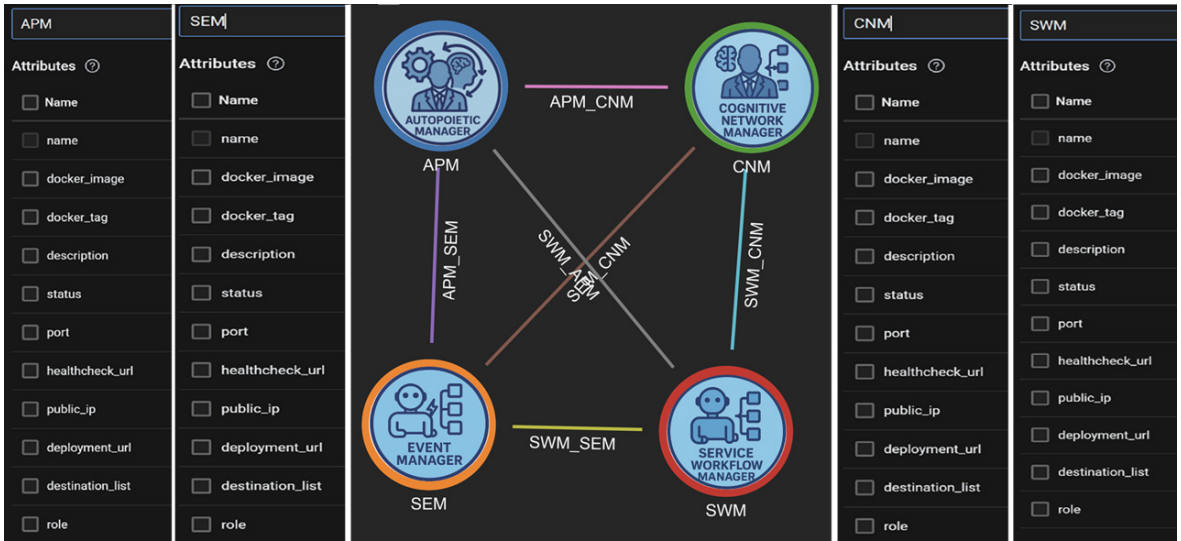


Figure 2. Schema of the Autopoietic and Meta-Cognitive Operating System.

3.2. Implementation: Distributed Loan Default Prediction

To demonstrate feasibility, we re-implemented the loan default prediction problem (Klosterman, 2019) using AMOS. The original problem—predicting whether a customer will default based on demographics and six months of credit history—was approached conventionally via supervised machine learning on a static dataset of 30,000 customers. In contrast, AMOS decomposes the application into containerized microservices that communicate via HTTP and persist events in a graph database: Cognitive Event Manager: Anchors the data plane, recording episodic and semantic memory. Customer, Account, Month, Billing, and Payment Services model financial behaviors as event-driven processes. UI Service ingests data and distributes workloads to domain services. Default Prediction Service computes next-month defaults using rule-based or learned models, providing auditable results. This event-sourced design enables reproducibility, explainability, and controlled evolution. Every prediction can be traced back through causal chains (e.g., Bill issued in April → Payment received → Default risk computation).

3.3. Demonstration of Autopoietic and Meta-Cognitive Behaviors

The distributed system is implemented in Python with TigerGraph as the graph database. AMOS manages container deployment on cloud infrastructure (IaaS/PaaS) with autopoietic functions such as self-repair via health checks and policy-driven restarts. The Cognitive Event Manager provides semantic and episodic memory that supports meta-cognition. At runtime, services: Detect anomalies (e.g., distributional drift in repayment codes). Adapt dynamically (e.g., switching from a rule-based baseline to logistic regression when model performance degrades). Provide explanations through event provenance queries (e.g., why was a customer labeled at risk?). This transforms a conventional ML pipeline into a living system with introspection, transparency, and resilience (Figures 3 and 4).



Figure 3. Schema for loan default application services with each node defining a process with inputs and outputs.

Figure 4 describes the schema implementation for the distributed software application in the cloud where each node executes a process using the input and produces output. The input and output are communicated among the service nodes depending on the shared knowledge derived from the functional and non-functional requirements and policy-based best practices and constraints.

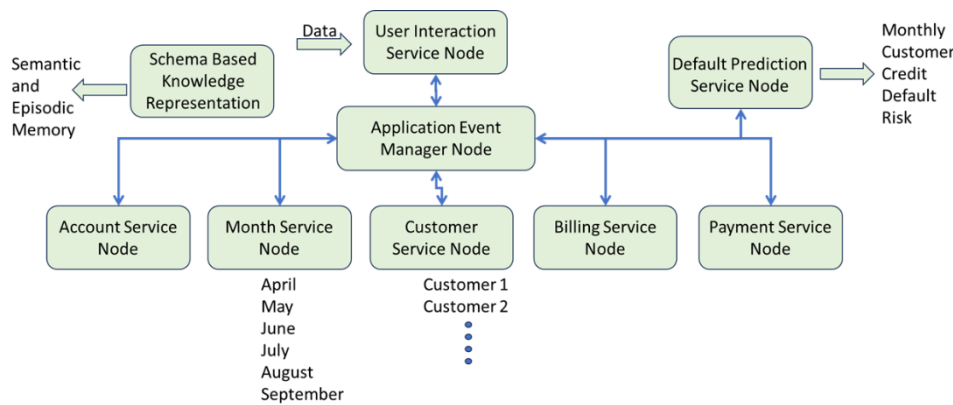


Figure 4. Workflow Diagram for GTI based credit default prediction implementation.

Figure 4. Workflow diagram of the GTI/DG-based credit-default implementation on AMOS. Services (with ports) form the control/data plane; the Cognitive Event Manager persists events and entities to the graph, and Default Prediction consumes the latest month to infer next-month outcomes. The knowledge network schema implemented in the graph database Replication of nodes that execute various service components, their communication paths, and redundancy of virtual service containers determine the resiliency of the system.

4. What We Learned from the Implementation

In developing the loan default prediction application on the AMOS platform, we demonstrated how a conventional machine learning task can be restructured as a knowledge-centric, autopoietic system.

The application was decomposed into modular and distributed services—including Customer, Account, Billing, Payment, and Event—each designed to perform a single, well-defined function. This modularization allowed the system to inherit the resilience and adaptability of distributed architectures.

Both functional requirements (e.g., prediction logic, data handling, feature generation) and non-functional requirements (e.g., performance, resilience, compliance, traceability) were encoded in the Digital Genome (DG). By doing so, the DG not only served as a design blueprint but also provided a persistent source of policies and constraints to regulate runtime behavior.

The system integrated two complementary forms of memory: associative memory, capturing patterns of past behavior, and event-driven history, recording temporal sequences of interactions. Specifically, the Cognitive Event Manager persists this structural knowledge—both ontologies/rules (Semantic Memory) and event histories (Episodic Memory)—within a high-performance graph database, enabling complex, graph-based queries for real-time validation and tracing. Together, these memory structures enabled continuous learning and made decision trails auditable over time.

Autopoietic and cognitive regulation were maintained through the AMOS core managers. The Autopoietic Process Manager (APM) ensured automatic deployment and elastic scaling of services; the Cognitive Network Manager (CNM) preserved inter-service connectivity under changing conditions; and the Software Workflow Manager (SWM) guaranteed logical process execution and recovery after disruption. These managers collectively provided the system with self-healing and self-sustaining properties.

Furthermore, the Cognizing Oracles and the nature of the prediction service directly address the epistemic requirements for trustworthiness. The Default_Prediction Service utilizes transparent rule-based logic (such as the `predict_next_pay` function. This logic converts numerical prediction codes into descriptive textual representation. This mechanism transforms the prediction from an opaque correlation into a discernible, auditable, and rule-based decision asset, satisfying Deutsch's criteria for knowledge by providing transparent justification for every decision.

Finally, large language models (LLMs) were leveraged during development, not as predictive engines, but as knowledge assistants. Their broad access to global knowledge was applied to schema design, service boundary discovery, functional requirement translation, and code/API generation. This highlights the complementary role of LLMs: they accelerate system design and prototyping, while AMOS ensures resilience, transparency, and adaptability in deployment. Figure 5 depicts the state evolution of two customers with default prediction status shown. The graph database allows queries to find explanations of the causal dependencies.

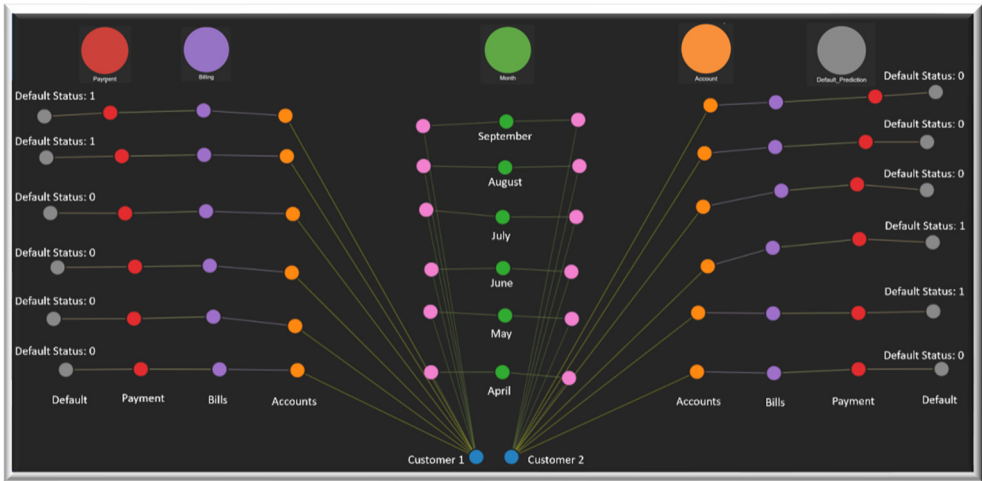


Figure 5. Predicted default status every month using the criteria in the algorithm used in the node Default Status process.

4.1. Comparison with Conventional Pipelines

Unlike traditional machine learning pipelines, which treat credit-default prediction as a static process (data preprocessing → feature engineering → model training → evaluation), the AMOS implementation transforms the task into a living, knowledge-centric system. Conventional pipelines deliver one-shot predictions on fixed datasets, with limited adaptability, traceability, or resilience. In contrast, AMOS maintains event-sourced memory, autopoietic feedback loops, and explainable decision trails, enabling the system to adapt to changing behaviors in real time while preserving transparency. This shift marks the difference between data-driven computation and knowledge-driven orchestration, demonstrating the potential of Mindful Machines to extend beyond conventional AI workflows.

Table 2 shows the comparison between two methods

Table 2. Comparison of Textbook Approach with the Mindful Machine Approach.

Aspect	Standard ML Pipeline	Your Multi-Service, Event-Driven Pipeline
Modularity / separation of concerns	Everything (data prep, model training, inference, etc.) often lives in one codebase / notebook	Each domain (customer, account, billing, payment, default prediction) is its own microservice
Temporal / sequential modeling	Typically, you flatten time-series / lag features and feed them to a model	Simulation of month-by-month progression; each service creates a memory of an event in time
Data lineage / graph modeling	The dataset is a table; relationships are implicit in features	Explicit model of graph relationships (e.g. UI → Event → Month → Customer → ... → Default) in TigerGraph
Communication & orchestration	Everything is synchronous and local (function calls, pipelines)	Asynchronous messaging via HTTP event broadcasts, sequential chaining
Scalability / deployment	Scales by scaling the entire pipeline or retraining on bigger hardware	Each service can scale independently, be deployed/distributed, and evolve over time
Interpretability & auditability	Interpretation is via feature importances, visualizations, etc.	Benefit of explicit event and graph tracking, and the “reasoning” can be more traceable across nodes
Learning vs. inference	The focus is usually on training + inference in a single code path	The system is more inference- (or simulation-) oriented; used in internal models or heuristics per service
State / Memory	The model sees all features at once	Each stage passes forward state; you have to design how state is propagated through the event chain

5. Discussion and Comparison with Current State of Practice

The implementation of AMOS highlights a fundamental departure from conventional machine learning pipelines. In Data Science Projects with Python [30], the credit-default prediction problem is solved through a linear process of data preprocessing, feature engineering, model training, and evaluation. While effective for static datasets, this approach exhibits several limitations: Lack of adaptability – the model must be retrained when data distributions shift. Bias and brittleness – results are constrained by dataset composition, often underrepresenting rare but critical cases. Limited explainability – outputs are tied to abstract model weights rather than transparent causal chains. Fragile resilience – pipelines are vulnerable to disruptions in data flow or system execution.

In contrast, the AMOS-based implementation demonstrates how a knowledge-centric architecture overcomes these shortcomings. By encoding both functional and non-functional requirements in the Digital Genome, the system embeds resilience and compliance directly into its design.

Technical Realization of Improvements: The claims of enhanced resilience, explainability, and adaptation are achieved through the tight coupling of the DG, Memory Layer, and Autopoietic Layer managers:

- Resilience (Self-Healing and Scaling): The Autopoietic Process Manager (APM) and Cognitive Network Manager (CNM) continuously monitor the distributed services and the network topology. Resilience is achieved through automated fault detection, followed by policy-driven actions such as service restarts or elastic scaling based on demand, which includes configuration of network paths for auto-failover using RTO/RPO objectives defined in the DG.

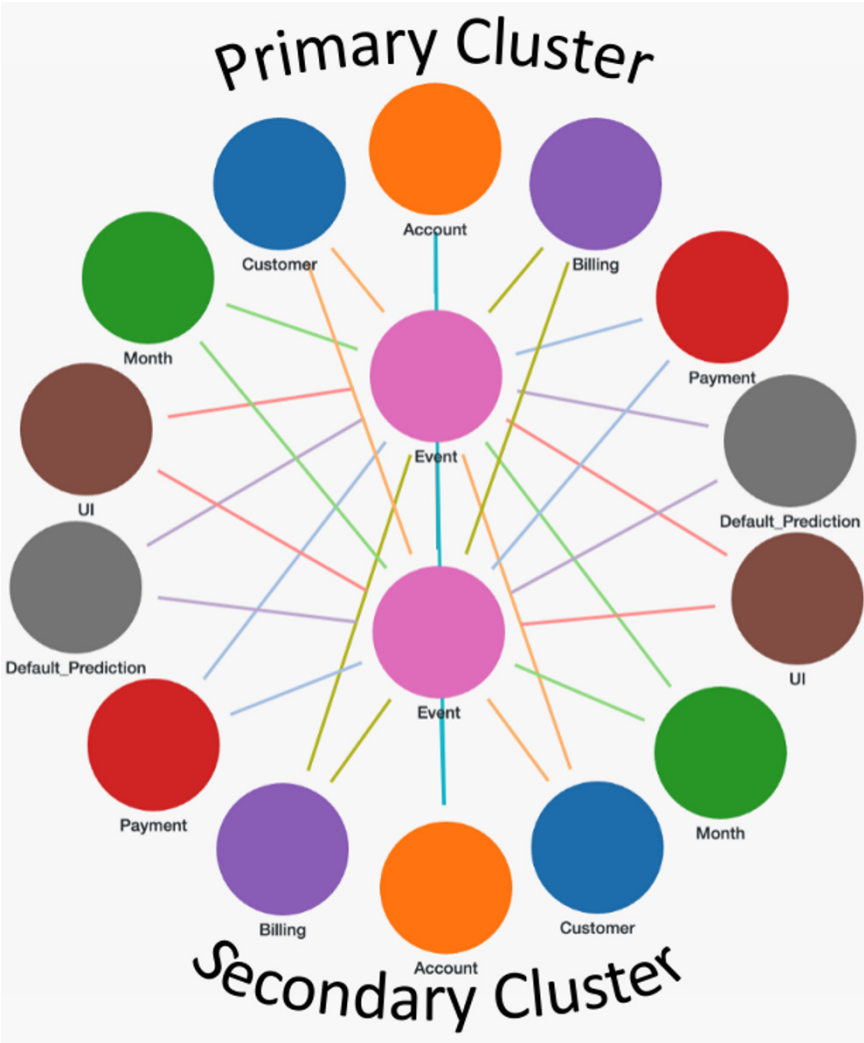


Figure 6. High Availability Active-Active Cluster Configuration.

Two clusters provide active-active high availability configuration

- **Explainability and Transparency (Auditable Decisions):** The event-sourced architecture, anchored by the Cognitive Event Manager, ensures every transaction and system decision is recorded in the Episodic Memory (a persistent graph database). This preserves the causal chain from input event to final prediction, making every outcome auditable via provenance queries. Furthermore, the use of transparent, rule-based reasoning in the prediction service, enforced by the Cognizing Oracles, allows the system to generate human-readable, textual justifications for risk classification, moving beyond opaque model weights.
- **Real-Time Adaptation:** Adaptability is achieved not by offline retraining, but by the SWM and CNM dynamically rewiring and reconfiguring the service network based on real-time policy evaluation. This allows the system to react instantly to detected anomalies (e.g., distributional drift) by switching prediction models or adjusting workflows without requiring a full system-wide deployment cycle.

Autopoietic managers ensure continuity under failure or load changes, while semantic and episodic memory provide persistent context for explainability and adaptation. Cognizing oracles introduce transparency and governance, ensuring that system behavior remains aligned with global goals.

We also compared our results with traditional ML analysis with Logistic regression discussed in the textbook.

Figure 7 shows predicted repayment delay for selected customers using two methods. Logistic regression tends to over-predict defaults around borderline risk scores. Month-to-month rules smooth out transient fluctuations by emphasizing recent recovery and consistent behavior. Together, they highlight the trade-off between statistical sensitivity and behavioral realism — and suggest a hybrid approach could yield the best balance of accuracy and interpretability.

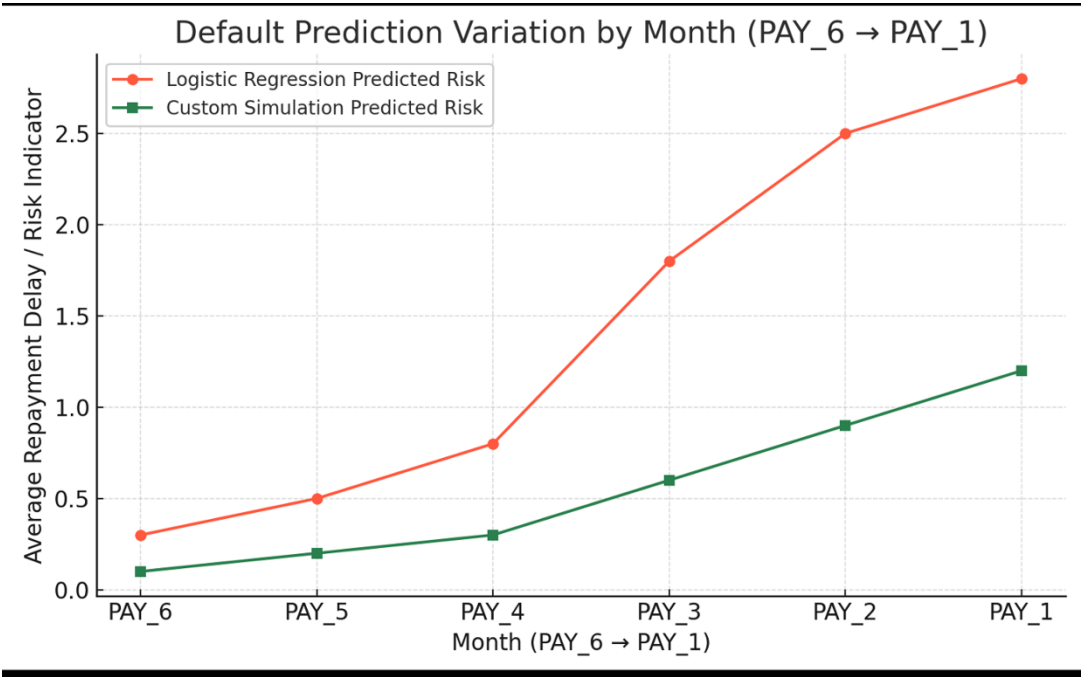


Figure 7. Comparison of Two Methods where PAY_6 is April and PAY_1 is September.

Here are some insights:

- The logistic regression model “thinks in probabilities” — it’s a snapshot predictor optimized for accuracy over a population.
- The rule-based simulation “thinks in causality” — it models account evolution, providing better interpretability for individual trajectories.

- Disagreement cases often reflect customers recovering from past delinquencies — where machine learning still penalizes them for their history.
- The logistic model generalizes historical averages, while your month-to-month model personalizes outcomes based on dynamic repayment behavior.

We conclude from our experience to use both methods. Use the simulation engine to transform raw PAY_1–PAY_6 into cleaned behavioral indicators (e.g., “ever overdue,” “recent recovery,” “months since full payment”). Feed those into the logistic regression, yielding a model that’s both predictive and explainable.

Beyond the loan default case, these findings generalize to a wide range of domains where AI must operate in dynamic, high-stakes environments such as finance, healthcare, cybersecurity, and supply chain management. Traditional pipelines, optimized for accuracy on static datasets, fall short in such contexts. Mindful Machines, by contrast, transform computation into a living process, where meaning, memory, and adaptation are intrinsic to the architecture. Table 1 summarizes the differences between conventional ML approaches and AMOS. As the comparison indicates, the novelty lies not in replacing machine learning, but in integrating it into a broader autopoietic and meta-cognitive framework that ensures transparency, resilience, and sustainability.

6. Conclusions

This work introduces Mindful Machines as a new paradigm for distributed software systems, demonstrating their feasibility through the AMOS platform and a loan default prediction case study. Grounded in the General Theory of Information, the Burgin–Mikkilineni Thesis, Deutsch’s epistemic framework, and Fold Theory, Mindful Machines unify the computer and the computed into a single self-regulating system. The case study illustrates several benefits: Transparency and Explainability – event-driven histories provide auditable decision trails. Resilience and Scalability – autopoietic managers enable continuous operation under failures or demand fluctuations. Real-Time Adaptation – workflows evolve with behavioral changes rather than requiring retraining. Integration of Knowledge Sources – statistical learning is augmented with structured knowledge and global insights via LLMs. Individual and Collective Intelligence – the system can detect anomalies at both single-user and group levels. In short, the textbook pipeline predicts defaults in a static, one-shot manner, whereas the AMOS-based approach reimagines prediction as a knowledge-centric ecosystem capable of adaptation, self-regulation, and long-term learning. This shift suggests that Mindful Machines can serve as a general framework for building transparent, adaptive, and ethically aligned AI infrastructures. Future work will extend these principles to larger-scale, real-world applications, testing the scalability of autopoietic and meta-cognitive mechanisms across domains such as healthcare diagnostics, enterprise resilience, and autonomous cyber-defense. By aligning computational evolution with the principles of knowledge, purpose, and adaptability, Mindful Machines represent a pathway toward trustworthy and sustainable AI.

Supplementary Materials: The following supporting information can be downloaded at the website of this paper posted on Preprints.org.

Author Contributions: Conceptualization, R.M; methodology, R.M, P.K.; software, P.K.; validation, R.M., P.K.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The Authors acknowledge Dr. Judith Lee, Director of the Center for Business Innovation for many discussions and continued encouragement.

Conflicts of Interest: “The authors declare no conflicts of interest.”.

Abbreviations

The following abbreviations are used in this manuscript:

AMOS	Autopoietic and Meta-Cognitive Operating System
DG	Digital Genome
APM	Autopoietic Manager
CNM	Cognitive Network Manager
SWM	Software Workflow Manager
PM	Policy Manager
SEM	Structural Event Manager
CEM	Cognitive Event Manager
SPT	Single Point of Truth

Appendix A. Credit Default Prediction

The presented implementation constitutes a rule-based prediction service designed to assess short-term credit repayment behavior. Developed in Python and deployed via the Flask web framework, the service provides a programmatic interface for evaluating account-level payment dynamics. Specifically, it exposes a RESTful endpoint (/default) that accepts structured JSON input containing billing, payment, and historical status information for one or more accounts. The output is a structured JSON response that details both the predicted repayment status for the subsequent period and the minimum required payment.

At the core of the system is the “predict_next_pay” function, which encodes a deterministic decision structure for repayment classification. The algorithm first computes a minimum required payment, determined as the greater of a fixed threshold or a percentage of the current billing amount. Based on the relationship between the actual payment amount, the billed amount, and the current repayment status, the function then categorizes accounts into discrete outcome states. These states include unused accounts with no outstanding balance, full repayment, partial repayment meeting minimum obligations, and delinquent accounts with payment delays ranging from one to nine months. Such categorization provides a standardized measure of repayment performance that is readily interpretable.

To support transparency and interpretability, a companion function (reason) maps the numerical prediction codes to descriptive textual explanations. This design choice facilitates the integration of model outputs into downstream decision-making processes, particularly in domains where interpretability and auditability are paramount, such as consumer credit risk management. The service further introduces a binary indicator, default_next_month, which flags accounts with two or more months of predicted delinquency as at risk of default in the subsequent billing cycle.

Overall, this system illustrates a lightweight and transparent approach to repayment prediction. Unlike complex machine learning models, which may introduce opacity and higher computational costs, the presented service emphasizes rule-based reasoning and interpretability. Its modular design and RESTful interface allow for seamless integration into broader financial risk assessment workflows, while its reliance on explicit rules enhances reproducibility and facilitates validation in operational contexts.

References

1. He, R., Cao, J., & Tan, T. (2025). Generative artificial intelligence: A historical perspective. National Science Review, 12(5), nwaf050. <https://doi.org/10.1093/nsr/nwaf050> (accessed on 21st October 2025)

2. Dillion, D., Mondal, D., Tandon, N. et al. AI language model rivals expert ethicist in perceived moral expertise. *Sci Rep* 15, 4084 (2025). <https://doi.org/10.1038/s41598-025-86510-0> (accessed on 21st October 2025)
3. Linkon, Ahmed & Shaima, Mujiba & Sarker, Md Shohail Uddin & Badruddowza, & Nabi, Norun & Rana, Md Nasir Uddin & Ghosh, Sandip & Esa, Hammed & Chowdhury, Faiaz Rahat & Rahman, Mohammad Anisur. (2024). Advancements and Applications of Generative Artificial Intelligence and Large Language Models on Business Management: A Comprehensive Review. *Journal of Computer Science and Technology Studies*. 6. 225-232. 10.32996/jcsts.2024.6.1.26.
4. Gupta, Priyanka & Ding, Bosheng & Guan, Chong & Ding, Ding. (2024). Generative AI: A systematic review using topic modelling techniques. *Data and Information Management*. 100066. 10.1016/j.dim.2024.100066.
5. Ogunleye, B., Zakariyyah, K. I., Ajao, O., Olayinka, O., & Sharma, H. (2024). A Systematic Review of Generative AI for Teaching and Learning Practice. *Education Sciences*, 14(6), 636. <https://doi.org/10.3390/educsci14060636> (accessed on 21st October 2025)
6. J. Hutson (2025). Ethical Considerations and Challenges. IGI Global Scientific Publishing. <https://doi.org/10.4018/979-8-3693-6905-0.ch007> (accessed on 21st October 2025)
7. Hanna, M. G., Pantanowitz, L., Jackson, B., Palmer, O., Visweswaran, S., Pantanowitz, J., Deebajah, M., & Rashidi, H. H. (2025). Ethical and bias considerations in artificial intelligence/machine learning. *Modern Pathology*, 38(3), Article 100686. <https://doi.org/10.1016/j.modpat.2024.100686> (accessed on 21st October 2025)
8. Afroogh, S., Akbari, A., Malone, E. et al. Trust in AI: progress, challenges, and future directions. *Humanit Soc Sci Commun* 11, 1568 (2024). <https://doi.org/10.1057/s41599-024-04044-8> (accessed on 21st October 2025)
9. Shangying Hua, Shuangci Jin, Shengyi Jiang; The Limitations and Ethical Considerations of ChatGPT. *Data Intelligence* 2024; 6 (1): 201–239. doi: https://doi.org/10.1162/dint_a_00243 (accessed on 21st October 2025)
10. John Banja (2020). How Might Artificial Intelligence Applications Impact Risk Management? *AMA J Ethics*. 2020;22(11):E945-951. doi: 10.1001/amajethics.2020.945.
11. Nisa, U., Shirazi, M., Saip, M. A., & Mohd Pozi, M. S. (2025). Agentic AI: The age of reasoning—A review. *Journal of Automation and Intelligence*. <https://doi.org/10.1016/j.jai.2025.08.003> (accessed on 21st October 2025)
12. Zota, R. D., Bărbulescu, C., & Constantinescu, R. (2025). A Practical Approach to Defining a Framework for Developing an Agentic AIOps System. *Electronics*, 14(9), 1775. <https://doi.org/10.3390/electronics14091775> (accessed on 21st October 2025)
13. Blackman, R. (2025, June 13). Organizations aren't ready for the risks of agentic AI. *Harvard Business Review*. <https://hbr.org/2025/06/organizations-arent-ready-for-the-risks-of-agentic-ai> (accessed on 21st October 2025)
14. The Society of Genes Yanai, I., & Lercher, M. J. (2016). *The society of genes*. Harvard University Press.
15. Ramakrishnan, V. (2024). *Why we die: The new science of aging and the quest for immortality*. William Morrow.
16. Ramakrishnan, V. (2018). *Gene machine: The race to decipher the secrets of the ribosome*. Basic Books.
17. Shannon, Claude E. (1948). "A Mathematical Theory of Communication." *Bell System Technical Journal* 27 (July & October): 379-423 & 623-656.
18. Burgin, M. *Theory of Information: Fundamentality, Diversity, and Unification*; World Scientific: Singapore, 2010. [Google Scholar]
19. Burgin, M. *Theory of Knowledge: Structures and Processes*; World Scientific Books: Singapore, 2016. [Google Scholar]
20. Burgin, M. *Structural Reality*; Nova Science Publishers: New York, NY, USA, 2012. [Google Scholar]
21. Burgin, M. *The Rise and Fall of the Church-Turing Thesis*. Manuscript. Available online: <http://arxiv.org/ftp/cs/papers/0207/0207055.pdf> (accessed on 21st October 2025).
22. Burgin, M.; Mikkilineni, R. From Data Processing to Knowledge Processing: Working with Operational Schemas by Autopoietic Machines. *Big Data Cogn. Comput.* 2021, 5, 13. [Google Scholar] [CrossRef]

23. Burgin, M.; Mikkilineni, R. On the Autopoietic and Cognitive Behavior. EasyChair Preprint No. 6261, Version 2. 2021. Available online: <https://easychair.org/publications/preprint/tkjk> (accessed on 21st October 2025).
24. Mikkilineni R. A New Class of Autopoietic and Cognitive Machines. Information. 2022; 13(1):24. <https://doi.org/10.3390/info13010024> (accessed on 21st October 2025).
25. Deutsch, D. (2011). *The beginning of infinity: Explanations that transform the world* (1st American ed.). Viking.
26. Hill, S. L. (Year). Fold Theory: A Categorical Framework for Emergent Spacetime and Coherence. Retrieved from Academia.edu https://www.academia.edu/142903174/Spacetime_Coherence_Theory_A_Unified_Framework_for_Matter_Energy_and_Information (accessed on 21st October 2025)
27. Hill, S. L. (Year). Fold Theory II: Embedding Particle Physics in a Coherence Sheaf Framework. Retrieved from Academia.edu https://www.academia.edu/130196602/Fold_Theory_II_Embedding_Particle_Physics_in_a_Coherence_Sheaf_Framework (accessed on 21st October 2025)
28. Hill, S. L. (Year). Fold Theory III: Fractional Fold Theory, Curvature and the Mass Spectrum. Retrieved from Academia.edu https://www.academia.edu/130258350/Fold_Theory_III_Fractional_Fold_Theory_Curvature_and_the_Mass_Spectrum (accessed on 21st October 2025)
29. Mikkilineni, R. (2025). General Theory of Information and Mindful Machines. Proceedings, 126(1), 3. <https://doi.org/10.3390/proceedings2025126003> (accessed on 21st October 2025)
30. Klosterman, S. (2019). Data Science Projects with Python: A Case Study Approach to Successful Data Science Projects Using Python, Pandas, and Scikit-learn. Germany: Packt Publishing.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.