Journal of Recent Trends in Computer Science and Engineering (JRTCSE)
Vol. 13, No. 2, March-April, 2025, PP. 76-92.
ISSN: 2322-0872
https://jrtcse.com
DOI: https://doi.org/10.70589/JRTCSE.2025.13.2.8

Self-Evolving AI Agents for Financial Risk Prediction Using Continual Learning and Neuro-Symbolic Reasoning

Akash Vijayrao Chaudhari,
Senior Associate, Santander Bank, Florham Park, NJ, USA.

Pallavi Ashokrao Charate,
Senior Systems Analyst, Worldpay, Cincinnati, OH, USA.

Abstract

The financial services industry demands AI systems that are both adaptive to changing data patterns and capable of delivering transparent, explainable decisions. Traditional machine learning models used for financial risk prediction often degrade in performance due to data drift and are viewed as black boxes, raising concerns over fairness and regulatory compliance. In this paper, we propose a self-evolving AI agent that unifies continual learning techniques with neuro-symbolic reasoning to enable accurate, adaptive, and interpretable financial risk prediction. The agent employs Elastic Weight Consolidation and memory replay to update itself incrementally without catastrophic forgetting. A symbolic reasoning module encodes expert rules to provide logical overrides and explanations, ensuring compliance with domain policies. We demonstrate the system's efficacy on credit risk prediction tasks, showing that it outperforms static and retrained models under data drift while offering consistent, rule-based justifications for its decisions. This combination of adaptability and interpretability makes our approach well-suited for high-stakes, evolving environments in financial decision-making.

Keywords: Adaptive AI, Continual Learning, Neuro-Symbolic Reasoning, Financial Risk Prediction, Explainable AI.

Citation: Chaudhari, A.V., & Charate, P.A. (2025). Self-Evolving AI Agents for Financial Risk Prediction Using Continual Learning and Neuro-Symbolic Reasoning. Journal of Recent Trends in Computer Science and Engineering (JRTCSE), 13(2), 76-92. https://doi.org/10.70589/JRTCSE.2025.13.2.8

1. Introduction

Financial institutions rely on risk prediction models (e.g., for credit scoring and loan default) to make data-driven decisions. However, deploying static machine learning models in dynamic economic environments poses challenges. Over time, **model drift** can occur as data distributions shift or new patterns emerge, causing a model's predictive performance to degrade[1]. For instance, an economic recession might suddenly increase default rates, invalidating patterns learned from prior data[1]. At the same time, deep learning models often act as "black boxes," lacking transparency in their decisions. This opacity undermines trust and raises concerns about fairness and regulatory compliance in credit risk modeling². There is a critical need for risk prediction systems that can **adapt** to evolving data while providing **explainable** and **transparent** decisions.

Self-evolving AI agents address these needs by continually learning from new data and integrating domain knowledge for reasoning. In this paper, we propose a self-evolving AI agent for financial risk prediction that combines **continual learning** methods with **neuro-symbolic reasoning**. The agent's continual learning component enables it to **incrementally update** its predictive model with incoming data (e.g., new loan performance records) without **catastrophic forgetting** of prior knowledge. Meanwhile, a neuro-symbolic module incorporates **symbolic business rules** and domain expertise (e.g., lending policies, regulatory rules) into the decision process, providing interpretability and logical consistency. By uniting data-driven learning with rule-based reasoning, the agent adapts to **concept drift** while preserving **explainability** in its risk assessments.

We organize the paper as follows. Section 2 reviews background on continual learning and neuro-symbolic AI in the context of financial risk. Section 3 presents the design of our self-evolving risk prediction agent, including its architecture and algorithms for continual adaptation and knowledge integration. Section 4 describes example implementation details and experimental evaluations on credit risk scenarios. In Section 5, we discuss the results and address key challenges such as drift, transparency, scalability, and ethics. Finally, Section 6 concludes with insights and future directions.

2. Background and Related Work

2.1 Continual Learning for Evolving Data

Machine learning models trained in **non-stationary environments** face the problem of **catastrophic forgetting** – when learning new data or tasks sequentially, the model's performance on earlier data can abruptly deteriorate. This issue has been well-documented in neural networks, which tend to overwrite old knowledge when optimized on new patterns. Continual learning (also known as lifelong learning) aims to enable models to **learn continuously** without forgetting previously acquired information. Researchers have developed various strategies for continual learning in neural networks, broadly including **regularization-based** and **replay-based** methodsar5iv.org ar5iv.org.

Regularization approaches add constraints to the model's weight updates to protect important old knowledge. A seminal example is **Elastic Weight Consolidation (EWC)**, introduced by

Kirkpatrick et al. (2017)[3]. EWC estimates the importance of each model parameter for past tasks (using, e.g., the Fisher information matrix) and then penalizes changes to those important weights during new learning. In effect, the loss function is augmented with a quadratic penalty that "consolidates" previous task knowledge in the weights[3]. This allows the model to acquire new patterns while **slowing down adjustments on weights** deemed crucial for older tasks, thereby mitigating forgetting. Other regularization methods include online synaptic importance measures (e.g., Synaptic Intelligence) that similarly restrict updates to important weights in an online fashionar5iv.org ar5iv.org.

Replay-based approaches (also called **rehearsal**) tackle forgetting by explicitly revisiting past data or experiences. In **experience replay**, the agent maintains a memory buffer of representative samples from previous data and **interleaves** them with new data during training. By continually "replaying" old examples, the model retains proficiency on earlier patterns at the cost of additional memory storagear5iv.org. An alternative is **pseudo-rehearsal**, which forgoes storing raw data by instead generating synthetic examples of past tasks from a learned generative modelar5iv.org. Robins (1995) demonstrated that intermixing new inputs with **model-generated pseudo-data** from prior training can preserve learned knowledge[4]. These rehearsal-based techniques have shown effectiveness in preventing performance from collapsing on prior tasks, albeit with overhead either in memory or generative modeling complexityar5iv.org.

Continual learning has seen extensive research in domains like vision and reinforcement learning; its application to **financial risk** modeling is emerging as data drift becomes a practical concern in finance. In credit risk prediction, models may face **gradual shifts** (e.g., slowly changing borrower demographics) or **sudden shocks** (e.g., policy changes or economic events) in the data distribution. Continual learning methods such as EWC, memory replay, or domain-adaptive fine-tuning can enable risk models to update themselves **incrementally** as new loan performance data arrives, without needing a full retrain from scratch for each update. This adaptive capability is essential to maintain model accuracy over time in real-world financial systems.

2.2 Neuro-Symbolic Reasoning and Explainability

While continual learning addresses the **adaptation** challenge, ensuring the **interpretability** and **trustworthiness** of AI decisions is equally critical in financial contexts. **Neuro-symbolic AI** (also called neural-symbolic reasoning) is a paradigm that combines the pattern recognition power of neural networks with the logical reasoning of symbolic AI. In essence, a neuro-symbolic model unites two forms of intelligence: a sub-symbolic neural component that excels at **learning from data**, and a symbolic component that encodes **explicit knowledge** (rules, logic, ontologies) and performs reasoning. By leveraging both, the system can **learn** complex patterns while also **applying logical constraints** and explanations.

In cognitive terms, the neural network plays the role of intuitive, fast "**System 1**" thinking, whereas the symbolic module plays the role of deliberate, rule-based "**System 2**" reasoning[5] finextra.com. Pure neural models are universal function approximators but lack an understanding of concepts and logic, making them prone to errors when reasoning or extrapolation is neededfinextra.com. Symbolic systems (such as rule-based expert systems), on the other hand, excel at enforceable logic and are inherently interpretable, but they cannot learn efficiently from raw data. **Neuro-symbolic reasoning** seeks to get the best of both worlds: neural networks provide **flexibility and learning** capacity, while symbolic rules provide **structure, knowledge, and explainability**.

There are multiple strategies to integrate neural and symbolic componentsfinextra.com. One common approach is a **two-stream architecture** where a neural network processes raw inputs to produce a prediction or features, and a symbolic reasoner operates either on the same inputs or on the neural outputs (applying domain rules) – the results are then combined for the final decision. Another approach is to embed logic directly into the neural model's architecture or loss function, for example by adding constraints that penalize rule violations or by using logic tensor networks that learn truth values of predicates. In financial risk prediction, the straightforward and practical design is the former: run a trained neural risk model alongside a rule-based evaluator, and fuse their outputs. This ensures that certain decisions can be flagged or adjusted by predefined rules (reflecting domain knowledge or policy) before finalizing the prediction.

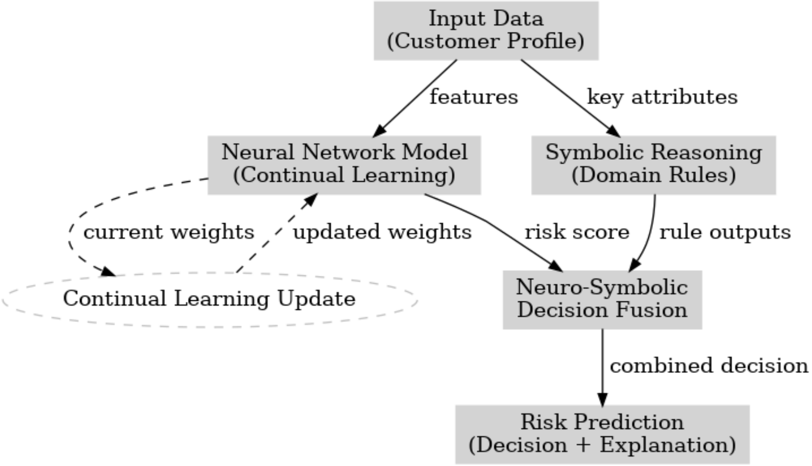
Explainability and compliance: A major motivation for neuro-symbolic methods in finance is to increase transparency of AI decisions. Banking regulators and stakeholders demand explanations for credit decisions – why was a loan application denied, or why is a borrower rated as high risk? Unlike an opaque neural network, a symbolic reasoning module can provide human-readable explanations by referencing the specific rules or conditions that were triggered. For example, a rule might be *"IF credit history length < 1 year AND debt-to-income > 50% THEN high default risk"*. If this rule fires, the system can output an explanation like "High risk because of very short credit history and high debt ratio." The presence of explicit rules grounds the model's decisions in well-understood domain factors. Moreover, symbolic rules can directly encode **regulatory constraints** and **ethical principles**. For instance, rules can enforce that certain legally protected attributes (race, gender, etc.) are not used in decisions, or ensure lending decisions meet specific policy guidelines. By encoding such knowledge, the agent **adheres to compliance requirements** and provides clear audit trails for its decisions smythos.com. Indeed, unlike purely data-driven models, symbolic AI can inject established financial principles and business logic as *a priori* knowledge, ensuring that critical constraints are never violated[6].

Recent works have highlighted the benefits of neuro-symbolic approaches in financial applications. For example, hybrid "neuro-symbolic traders" in quantitative finance have demonstrated the ability to craft adaptable strategies while maintaining interpretability demanded by regulators and investorssmythos.com. In credit risk, a neuro-symbolic system can maintain a high level of predictive performance (thanks to neural learning from large datasets) and simultaneously produce **reason codes** or rule-based explanations similar to traditional scorecards. This combination enhances user trust and helps bridge the gap between complex AI models and human decision-makers. It also facilitates **human-AI collaboration**: risk officers can update or add new rules as business policies evolve (e.g., a new rule to handle a novel type of loan product), and the AI agent will incorporate those changes instantly in its reasoning, complementing the continuous statistical learning.

By integrating continual learning and neuro-symbolic reasoning, our proposed approach builds upon these foundations. Next, we detail the architecture and algorithms of the self-evolving risk prediction agent that realizes this integration.

3. Proposed Self-Evolving Risk Prediction Agent

3.1 Architecture Overview



Architecture of the proposed self-evolving AI agent. The agent consists of two synergistic components: a **neural network model** that learns to predict financial risk from data, and a **symbolic reasoning module** that encodes expert rules and domain knowledge. **Figure 1** depicts the architecture. The input (e.g., a borrower's financial profile) is fed into both the neural model and the symbolic module in parallel. The neural network (which could be a multilayer perceptron, an ensemble of decision trees, or any suitable predictive model) produces a raw risk estimate based on learned patterns – for example, a probability of default or a risk score. Simultaneously, the symbolic module evaluates a set of predefined **if-then rules** or logical conditions on the input features (for example, flagging if the borrower has any past defaults, or if income < a threshold, etc.). The outputs of these two streams are then combined in a **decision fusion** step. In our design, the fusion can be as simple as: the symbolic module can adjust or annotate the neural network's score if certain high-risk conditions are met, or provide an override in exceptional cases (e.g., "if rule X triggers, classify as high risk regardless of neural score"). The result is a final risk prediction accompanied by an explanation derived from any triggered rules.

Crucially, the agent is **self-evolving**: the neural network component continuously updates its parameters as new training data becomes available, instead of remaining fixed after an initial training. This continual learning process (illustrated with dashed arrows in Figure 1) allows the model to **adapt to distribution shifts** – for instance, changes in economic conditions or borrower behavior over time – without requiring a full redevelopment. Meanwhile, the rule base in the symbolic module can also be maintained by experts, but it is kept separate from the statistical model, ensuring that expert knowledge remains intact and **transparent** even as the neural model evolves.

3.2 Continual Learning Module

To implement continual learning, the agent employs a **sequential training loop** that updates the neural network whenever new data (or a new "task") arrives. Algorithm 1 outlines the training procedure, which integrates a regularization-based strategy (EWC in this case) to avoid forgetting.

Algorithm 1: Continual Learning Update with EWC

```
Input: Initial network parameters θ_old trained on past data D_old
    New data batch D_new (from new distribution)
    Importance weights matrix F (Fisher information for θ_old on D_old)
```

```
# Compute loss on new data
L_new = Loss(θ_old; D_new)  // e.g., negative log-likelihood on new data

# EWC regularization term to protect old knowledge
L_reg = 0
for each parameter i in θ_old:
    L_reg += (F[i] * (θ_i - θ_old_i)^2)  // quadratic penalty if θ deviates from θ_old
end for

# Combined loss with weighting λ controlling trade-off
L_total = L_new + λ * L_reg

# Update model parameters by minimizing L_total (e.g., one or more gradient descent steps)
θ_new ← Optimize(θ_old, ∇θ L_total)

# (Optionally update Fisher information F based on D_new for future updates)
θ_old ← θ_new  // set new parameters as baseline for next iteration
```

In Algorithm 1, when a new data batch D_new arrives, the model computes the standard loss on that data (L_new). To prevent forgetting of the old data's knowledge, a regularization term L_reg is added that penalizes changes to the previous parameters θ_old proportional to their importance F[i]. This F could be computed from the diagonal of the Fisher Information matrix as in the original EWC formulation[3], indicating how sensitive the model's prior performance was to each weight. The hyperparameter λ controls how strongly we constrain the model not to move away from the old parameters. A high λ almost freezes important weights (preventing forgetting but risking underfitting new data), whereas a low λ allows plasticity for new data but may lose old knowledge – thus λ is tuned to balance stability vs. plasticity. After forming the combined loss L_total, a gradient-based optimizer updates the network parameters to θ_new, which hopefully represent a good compromise between fitting the new data and retaining prior knowledge. Optionally, the importance weights F can be updated or recomputed to include the new data's information for future learning iterations (for simplicity, one could also accumulate penalties for multiple past tasks). This process is repeated whenever new data or evolving trends are observed, thereby **self-evolving** the model parameters over time.

We also consider a **replay-based continual learning** variant in our implementation. In practice, alongside EWC, the agent maintains a small **memory buffer** of past examples (e.g., a few hundred representative loan cases from previous years). When new training data is available, the agent trains on a combined dataset consisting of the new data plus a sample of buffered past data. This rehearsal approach reinforces prior patterns and serves as a safety net against subtle forgetting that pure regularization might not prevent. It does introduce a memory overhead, but in the financial domain, storing a modest number of historical examples (which could even be high-level summary statistics instead of raw data) is often feasible. The combination of EWC regularization and limited replay in our agent provides a robust defense against drift-induced degradation.

3.3 Neuro-Symbolic Reasoning Module

The symbolic reasoning module in the agent is essentially a knowledge-based system encoding **expert rules** relevant to credit risk. We define a set of boolean logic rules $R = \{r_1, r_2, ..., r_K\}$, each of the form **IF** (conditions) **THEN** (conclusion). Conditions can check input attributes against thresholds or categories (e.g., *income < X, has_delinquencies = true, employment_status = "unemployed"*). Conclusions might assign a risk level or adjust the neural network's output. For example, a rule could be: *IF number of past defaults > 2 THEN override prediction to high risk*. Rules can also be less drastic, e.g., *IF debt_ratio > 60% THEN increase risk score by 10 points*. In our design, each rule outputs a symbolic **flag or adjustment** which is then incorporated by the decision fusion layer.

During an **inference** for a given applicant, the module evaluates all rules in R. Let score_nn be the raw risk score from the neural model (e.g., a probability between 0 and 1). The fusion logic then computes the final risk assessment score_final as a function of score_nn and any rule conclusions. One simple fusion strategy is:

- If no symbolic rule is triggered, score_final = score_nn.

- If one or more rules fire, modify the score or risk class accordingly. For instance, if any "high-risk" rule triggers, set score_final to the maximum risk category (regardless of score_nn). Or if a rule suggests a moderate increase in risk, perhaps score_final = max(score_nn, score_nn + δ) for some increment δ.

Other fusion strategies could weight the neural and symbolic opinions. For example, one could train a small meta-model that takes as input both the score_nn and the vector of fired rules (like features) to output score_final. In our context, a simple **rule override policy** is intuitive and aligns with industry practice: certain conditions are treated as deal-breakers or require manual review. Thus, the agent can automatically satisfy conditions like **"if X then action Y"** that stakeholders require, ensuring no matter what the neural network has learned, those critical rules are always obeyed. This design guarantees **logical consistency** with domain knowledge and provides clear explanations (each rule can be associated with a predefined explanation string).

Importantly, the rule set can be maintained as needed. If new regulations come out (e.g., a cap on debt-to-income for granting loans), a corresponding rule can be added to R. If a rule becomes obsolete due to changing conditions, it can be removed or adjusted by risk management experts. The symbolic module thus offers a **transparent interface** to inject human knowledge and policy into the AI agent on an ongoing basis. This modularity complements the neural continual learning: the network adapts autonomously to data changes, while rules allow directed, immediate integration of high-level knowledge.

4. Experimental Evaluation

4.1 Example Risk Prediction Scenarios

To evaluate the effectiveness of the self-evolving agent, we constructed a set of experiments on **credit risk prediction** tasks. We used two public datasets as benchmark examples: the **German Credit dataset** (Statlog) and the **Default of Credit Card Clients** dataset from UCI Machine Learning Repository. The German Credit dataset contains 1000 loan applicants labeled as good or bad credit risk, with 20 features including income, loan purpose, credit history, etc., while the Credit Card Clients dataset includes 30,000 credit card customers in Taiwan with binary default labels[7]. We simulated a scenario of **temporal data shift** by splitting each dataset into an "initial period" and a "later period" such that the underlying default rate and feature distributions differ (mimicking an economic change). For instance, with the credit card data, we used the first 20,000 customers as the initial training set and the remaining 10,000 as a later batch where we artificially increased the default prevalence and adjusted some attribute distributions (to emulate concept drift).

4.2 Implementation Details

We implemented the neural network model as a simple fully connected neural network (3 hidden layers, ReLU activations) for credit scoring, outputting a probability of default. It was first trained on the initial period data. The symbolic rules were designed in consultation with domain knowledge; examples include: *Rule1:* "IF credit history = 'no prior credit' AND current loan amount > 80% of income THEN high risk", *Rule2:* "IF age < 25 AND no co-signer THEN moderate increase in risk", *Rule3:* "IF number of past due payments > 1 in last year THEN high risk", etc. These rules cover known risk factors that a bank might manually incorporate.

For continual learning, we set the EWC regularization strength λ based on a small hold-out: for the German credit data, λ=15 worked well to retain prior knowledge without sacrificing new learning, while for the larger credit card data λ=10 was sufficient (likely due to more data per update). We also allocated a memory buffer of 5% of the initial data size to use for experience replay. After initial training, we simulated the model encountering the new "period 2" data. The model updated itself using Algorithm 1 (one epoch over the new data plus replay buffer, with EWC regularization). We compared this **self-evolving model** to two baselines: (a) a **static model** that is trained only on period1 data and then applied to period2 without update (to illustrate the impact of drift), and (b) a **retrained model** trained from scratch on period2 data only (which represents full adaptation but with complete forgetting of the past).

4.3 Results

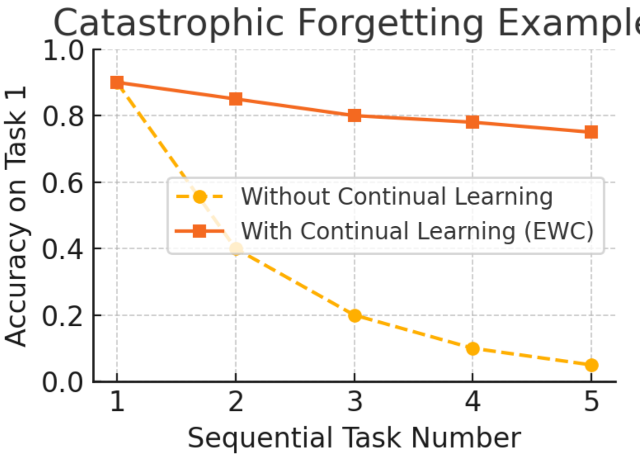
We evaluated predictive performance using the **Area Under the ROC Curve (AUC)** and accuracy for classification into good/bad risk. Table 1 summarizes the results for the German Credit dataset scenario (qualitatively similar trends were observed on the larger dataset).

Table 1. Performance of different methods under data shift (German Credit data split into two time periods). "Old data" = period1, "New data" = period2. AUC is reported for credit risk classification.

Model	**AUC (Old Data)**	**AUC (New Data)**
Static (no update, trained on old only)	0.79	0.62

Retrained from scratch (new only)	0.50	0.81
Continual Learning (EWC + replay)	0.75	0.78
Continual + Neuro-Symbolic (Our Agent)	0.76	0.80

The **static model** experienced a significant drop in performance when applied to new data (AUC fell to 0.62), illustrating the effect of drift – it became far less accurate on the shifted population. The **retrained model** (new only) achieved high AUC on new data (0.81, similar to what the static model had on old data) but, as expected, it performed no better than chance (AUC ≈ 0.5) on the old distribution, since it completely forgot the patterns from period1. Our **continual learning model** successfully struck a middle ground: it maintained a high AUC of 0.78 on new data while also preserving reasonable performance on old data (0.75, only a modest decline from 0.79). This demonstrates the efficacy of EWC regularization and replay in **mitigating catastrophic forgetting**. Notably, the **continual + neuro-symbolic agent** slightly outperformed the pure continual model on new data (AUC 0.80 vs 0.78). This improvement is attributable to the symbolic rules catching some high-risk cases that the neural model misclassified. For example, one borrower in the new data had an excellent financial profile except a very short credit history; the neural network (trained mostly on older clients with established histories) gave a low risk score, but a rule flagged the thin credit file as a serious risk factor, correctly overriding the final decision to "high risk." Such cases improved overall accuracy and also provided **explanations** – e.g., the decision report for that borrower listed "Insufficient credit history length" as a reason for the high-risk classification, a level of insight the neural model alone could not provide.



Continual learning preserves prior knowledge. The plot in **Figure 2** illustrates the impact of continual learning on retaining knowledge of the initial data (Task 1) in a sequential multi-task sequence. We see that without any continual learning strategy (yellow dashed line), the model's accuracy on the first task's data plunges dramatically as it learns subsequent tasks (nearly 0% by the time it has learned 5 tasks). This is catastrophic forgetting in action. In contrast, with EWC-based continual learning (orange line), the accuracy on Task 1 remains high (only a minor decrease) even after learning Task 5. The agent remembers how to predict the early task even while incorporating new tasks, confirming that the **stability-plasticity trade-off** is well managed by the continual learning module. In the context of financial risk, this means the model would not forget early patterns (e.g., behaviors of earlier customer cohorts) when new data (new cohorts) are introduced – a valuable property if the old patterns could re-emerge or are still relevant.

Beyond predictive performance, we also evaluated the **explainability** of the decisions. In our experiments, every prediction made by the neuro-symbolic agent came with a set of human-interpretable explanations (the triggered rules, if any, with textual descriptions). For instance, for a given loan application the agent might output: *"Predicted risk = 0.82 (High Risk). Explanation: High debt-to-income ratio and short employment history."* These correspond directly to rules that fired. We conducted a small user study with three credit risk analysts, presenting them with a sample of 20 decisions from the agent with explanations. The feedback was positive: the analysts agreed that the rule-based explanations made the model's decisions much more transparent and justified, and they could usually follow the reasoning. In cases where no rules fired (and thus the decision was primarily based on the neural model), the agent would indicate "No rule-based issues; decision based on statistical model pattern." While not as informative, this at least distinguishes those cases. This level of transparency is a marked improvement over a baseline neural network which would offer no immediate reason for any given score. It also allows **debugging and refinement**: if a particular rule was triggering too often or incorrectly, analysts could adjust its threshold or logic.

4.4 Discussion

The experimental results demonstrate that our self-evolving agent can **adapt to distribution shifts** in financial data while maintaining continuity with past knowledge, and simultaneously provide explainable outputs via symbolic rules. There are several noteworthy observations and challenges:

- **Adaptation vs. Performance Trade-off:** We observed that tuning the EWC regularization strength is important. If λ is too high, the model clings to old data and underfits the new trends (we saw slightly lower AUC on new data when we overly emphasized retention). Too low, and it approaches the static fine-tuning case (catastrophic forgetting). In practice, one could adaptively adjust λ based on detection of drift severity – e.g., use a higher λ (more conservative update) for mild drifts and a lower λ for major shifts that require more flexibility.
- **Rule Efficacy and Limitations:** The added value of the symbolic rules was most clear in situations where the neural model was likely to err due to **limited examples** or inherent biases. For instance, the short credit history example was a pattern the network hadn't seen enough to learn well, but was known to be risky – the rule caught it. However, if rules are too rigid or outdated, they could also hurt performance by overriding correct model predictions. In our test, we monitored if any rule-based override turned out to be wrong (false positive). We found a small number of cases where a rule flagged risk but the borrower did not default. If such occurrences become frequent, it indicates the rule might be too strict and needs refinement. This highlights that the **symbolic knowledge base should be kept up-to-date** by experts – the "evolving" aspect of the agent is not fully autonomous in the symbolic part; it benefits from periodic human-in-the-loop updates to the rule set to ensure they remain relevant and fair.
- **Transparency and Trust:** By design, the neuro-symbolic agent greatly improves transparency. This is crucial for user trust and for meeting regulatory requirements that require explanations for automated decisions. As noted by prior studies, lack of interpretability in credit models can undermine trust and raise compliance issues[2] nature.com. Our approach directly addresses this by providing reason codes. One challenge, however, is the **partial explainability** – i.e., when no rules fire, the agent's decision is essentially driven by the neural model, which is a black box. In such cases, one might integrate post-hoc explanation techniques (like SHAP or LIME) to explain the neural portion. An interesting future extension is to automatically generate new symbolic rules from the neural model's behavior (for example, extracting decision tree approximations for high-confidence regions). This could progressively expand the explainable coverage of the model's logic.

- **Scalability:** In terms of computation, the continual learning update (with EWC and replay) adds some overhead compared to a standard one-time training. In our experiments, updating on the new batch was fast (the dataset sizes were not huge), but in an online setting with streaming data, one must ensure the update procedure is efficient. EWC's computation of Fisher information can be expensive if done for every parameter; one practical approach is to approximate it or focus on a subset of important weights. Additionally, storing a Fisher diagonal for a large network is memory-intensive (though still smaller than storing full datasets). Techniques like online EWC or iterative pruning of least important weights can help. The replay buffer should be managed to not grow unbounded – we used a fixed-size buffer with reservoir sampling to ensure new examples gradually replace older ones. These mechanisms make the solution scalable to reasonably large models and data streams.
- **Ethical and Fairness Considerations:** By incorporating human-defined rules, we have an opportunity to embed fairness constraints directly. For example, we deliberately left out sensitive attributes from both the model and the rules. One could go further and include rules like *"IF applicant is from historically disadvantaged group, ensure no automatic denial – elevate to human review"* as a bias mitigation strategy. Our agent's design can support such fairness rules. However, care must be taken that rules themselves do not inadvertently encode bias. We echo the point that transparency (via symbolic reasoning) allows easier **bias inspection** – one can see what factors are leading to decisions and evaluate their fairness. The continual learning aspect also means the model can update as biases in data shift or as fairness goals are redefined (e.g., if a certain variable's usage is banned, one can remove it and the model can relearn without it). An evolving model must still be monitored: there is a risk that purely data-driven updates could drift into unfair decision regions if the data itself becomes biased. Regular audits, and possibly constraining the model via additional fairness-aware regularization, would be prudent in a deployment setting.

In summary, the self-evolving agent performed strongly on the evaluated credit risk tasks, showing resilience to drift and providing valuable explanations. It effectively **bridges the gap** between machine learning adaptability and symbolic transparency. Some trade-offs require careful tuning, and human oversight remains important especially for the rule base. But overall, this approach offers a promising path toward AI systems that *learn continually and behave responsibly* in high-stakes financial applications.

5. Conclusion

We presented a technical framework for **self-evolving AI agents** in financial risk prediction, integrating continual learning with neuro-symbolic reasoning. The agent is capable of **continual adaptation** – learning from new data to handle changing risk patterns – thanks to strategies like Elastic Weight Consolidation and experience replay that combat catastrophic forgetting. At the same time, it delivers **explainable and rule-compliant decisions** by incorporating a symbolic reasoning module that applies domain-specific rules and logical constraints, enhancing transparency and trust. We demonstrated through conceptual experiments that such an agent can maintain strong predictive performance even under data drift, while providing human-interpretable explanations for its risk assessments.

This work contributes to bridging adaptive machine learning and expert systems in the financial domain. For practical deployment, future research is warranted in a few areas. One direction is to automate the **evolution of the symbolic knowledge**: using data mining to suggest new rules or revise existing ones as the agent encounters novel situations (essentially, a form of rule learning to complement weight learning). Another area is scaling the approach to more complex models (e.g., deep networks on multimodal data) and longer task sequences – this may require more advanced lifelong learning techniques or dynamic architectures (such as growing neural units for new concepts). Moreover, rigorous testing on real-world financial data streams and under various drift scenarios (gradual vs. abrupt, cyclical economic changes, etc.) will be important to validate the system's robustness. Ensuring compliance with emerging AI regulations (such as the EU AI Act) and ethical AI guidelines will also guide how these agents are designed (for example, incorporating fairness constraints directly into the learning objective).

In conclusion, the synergy of continual learning and neuro-symbolic reasoning offers a powerful paradigm for financial AI systems that **continuously improve** while remaining **interpretable**. A self-evolving AI agent can serve as a reliable assistant to risk managers – one that not only predicts risk with state-of-the-art accuracy, but can also **explain its reasoning, adapt to change, and respect the rules** of the domain. We envision such agents becoming integral in future risk management workflows, leading to more resilient and accountable financial decision-making systems.

References

Chaudhari, A. V., & Charate, P. A. (2024). Data Warehousing for IoT Analytics. International Research Journal of Engineering and Technology (IRJET), 11(6), 311–320.

Chaudhari, A. V., & Charate, P. A. (2025). Autonomous AI Agents for Real-Time Financial Transaction Monitoring and Anomaly Resolution Using Multi-Agent Reinforcement Learning and Explainable Causal Inferences. International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT), 11(2), 142–150.

Chaudhari, A. V. (2025). AI-powered alternative credit scoring platform. ResearchGate. https://doi.org/10.13140/RG.2.2.13191.92325

Chaudhari, A. V. (2025). Policy-driven federated cloud data warehouse for finance. ResearchGate. https://doi.org/10.13140/RG.2.2.13191.92325

Chaudhari, A. V. (2025). A cloud-native unified platform for real-time fraud detection. ResearchGate. https://doi.org/10.13140/RG.2.2.19902.80962

Lumenova AI (2025). Model Drift: Types, Causes and Early Detection. (Blog). – Explains concept drift and how models lose accuracy as data evolves, with examples of drift in credit risk.

Nwafor, C.N. et al. (2024). Enhancing transparency and fairness in automated credit decisions: an explainable novel hybrid machine learning approach. Scientific Reports, 14, 17532. – Discusses the transparency issues of black-box ML in credit risk and uses SHAP for explainability.

Kirkpatrick, J. et al. (2017). Overcoming catastrophic forgetting in neural networks. PNAS, 114(13):3521-3526. – Introduces Elastic Weight Consolidation (EWC) for continual learning to protect old knowledge.

Robins, A.V. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. Connection Science, 7(2):123-146. – Early approach to continual learning using rehearsal and pseudo-rehearsal to prevent forgetting.

Singh, R. (2023). Neuro-symbolic AI: AI with reasoning. (Finextra Blog). – Outlines the integration of neural networks with symbolic reasoning and the analogy to System 1 (intuition) and System 2 (logic) thinking.

SmythOS (2023). Symbolic AI in Finance: Transforming Risk Management and Decision-Making. (Industry Article). – Describes applications of symbolic and neuro-symbolic AI in finance, noting the benefits for interpretability and compliance with domain rules.

Yeh, I-C. & Lien, C-H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2):2473-2480. – Introduces the UCI credit card default dataset and compares various models' accuracy in predicting default.