Transformer-Driven Financial Process Automation Optimization

Su Zheng

Department of Accounting, Management College, Shenyang Urban Construction University, Shenyang 110167, Liaoning, China
Corresponding author: Su Zheng (e-mail: 13840065466@163.com)

ABSTRACT The increasing complexity of enterprise financial data processing poses significant challenges for intelligent information systems that require efficient semantic understanding and real-time decision support. These requirements are also relevant to communication-enabled infrastructures and electromagnetic information environments, where heterogeneous data integration and rapid risk perception are essential for reliable system operation. To address these issues, this study proposes a Transformer-driven financial process automation framework that integrates semantic modeling, multimodal data fusion, reinforcement learning-based process optimization, anomaly detection, and knowledge graph construction. A pretrained Transformer is employed to capture semantic dependencies from financial documents, while structured and unstructured information is jointly modeled through cross-modal attention mechanisms. The reinforcement learning policy network dynamically adjusts process execution paths to improve operational efficiency, and attention distribution analysis combined with clustering techniques enables accurate anomaly identification and risk warning generation. In addition, a semantic knowledge graph is established to support interpretable process management and intelligent decision-making. Experimental results demonstrate that the proposed framework achieves 95.72% anomaly detection recall, 93.57% warning precision, and substantially reduces execution latency compared with conventional automation systems while improving semantic recognition accuracy across multiple financial entities. The proposed approach provides an effective solution for intelligent financial process optimization and offers valuable methodological insights for data-driven monitoring, semantic information processing, and adaptive decision support in communication-intensive and electromagnetic sensing applications.

INDEX TERMS financial process automation, transformer, multimodal data fusion, semantic modeling, electromagnetic information processing

I. INTRODUCTION

THE rapid development of the global digital economy has driven the continuous growth of corporate financial data; simultaneously, this digital expansion is fueling an equivalent surge in high-frequency, multi-source data , encompassing everything from real-time consumer interactions to global raw material trading. Finance departments are facing the real challenges of massive volumes of in-formation, diverse formats, and complex semantics when processing transaction vouchers, invoices, re-ports, contracts, and audit data [1,2]. Traditional financial automation systems have long relied on rule-based process control and templated recognition methods, lacking sufficient understanding of finan-cial text at higher semantic levels and struggling to capture the semantic dependencies between different financial elements [3,4]. This asymmetric information flow and fragmented data semantics limit the level of intelligent financial processes, lengthen corporate decision-making cycles, and reduce resource allocation efficiency. As the development trend of intelligent finance becomes increasingly obvious, traditional methods that rely solely on templates, regular expressions or keyword matching can no longer meet the needs of enterprises in terms of refinement, real-time and explainability. Financial processes urgently need to be comprehensively optimized through deep semantic modeling and adaptive learning mechanisms [5,6].

In recent years, academia and industry have conducted multi-level explorations of intelligent financial automation, with research directions mainly focusing on process identification, semantic extraction, and data mining. In the field of financial process optimization and intelligent research, existing work has explored multiple levels, including system integration, automation application, and data analysis. Ren [7] constructed an optimization model for enterprise financial management and decision-making systems based on big data technology, providing methodological support for the processing and analysis of massive financial data. Finally, Ahmadi [8] comprehensively reviewed the current status and future opportunities of the integration of big data and artificial intelligence in the financial industry,

providing a forward-looking perspective on the evolution of intelligent financial systems. The above research lays an important foundation for the design of this method in terms of semantic recognition, process optimization, and risk warning. In the field of natural language processing and knowledge modeling, the application of the Transformer architecture has promoted the innovation of semantic understanding technology. This model implements global dependency modeling of long sequences based on the self-attention mechanism, so that contextual relationships can be fully captured and semantic expressions are more accurate. Some studies have applied Transformer to audit text analysis, invoice recognition and report generation tasks, and achieved significant recognition performance improvements. Abiola-Adams et al. [9] emphasized the core principles of financial stability management by optimizing balance sheet performance and asset liability management strategies, providing theoretical support for the design of the risk warning module in this paper. Oluoha et al. [10] further used advanced data analysis technology to optimize business decisions, reflecting the value of data-driven methods in financial process automation, and directly inspired the construction of the semantic recognition model in this paper. Finally, Olorunnimbe et al. [11] integrated financial time series through time Transformer, providing advanced methodological support for time series data analysis and anomaly detection in this paper. To address these problems, this paper applies a Transformer-based semantic modeling system, and through multimodal fusion and strategy learning mechanisms, constructs a financial process optimization framework with adaptive capabilities to break through the limitations of traditional methods in semantic depth and automation breadth [12,13].

This paper proposes a Transformer-based automated optimization system for financial processes, encompassing semantic modeling and decision optimization. The research first jointly models financial text and structured account data, designing a feature extraction architecture based on a multi-head attention mechanism to achieve high-dimensional semantic representation of financial elements. Through a cross-source data fusion network, unstructured semantic information is mapped to structured data, improving the system's understanding of complex semantic relationships. A reinforcement learning module dynamically adjusts financial process paths, achieving self-optimization and strategy generation. The system identifies potential abnormal transactions through attention features and automatically generates risk warning reports by combining cluster analysis. Furthermore, a financial process knowledge graph is constructed to realize structured associations between financial elements and process visualization. This framework supports the intelligent upgrade of financial automation systems, promotes the transformation to a data-driven financial management model, and supports the intelligent automation and strategic transformation of complex production plans.

II. SEMANTIC-DRIVEN FINANCIAL PROCESS OPTIMIZATION MECHANISM

Building a Financial Text Embedding Model Based on Pretrained Transformers

DATA PREPROCESSING AND CORPUS NORMALIZATION

To ensure the semantic consistency and feature integrity of the model input, the financial corpus is first preprocessed at multiple levels. A unified text cleaning process based on word segmentation and part-of-speech tagging is applied to text data from different sources, including financial voucher descriptions, invoice content, accounting entries, cash flow descriptions, and report annotations. Financial feature items such as amounts, account codes, and voucher numbers are retained as independent identifiers through regular matching and word segmentation dictionary expansion to avoid semantic loss. Synonymous expressions, industry abbreviations, and differences in monetary units in unstructured sentences are standardized to ensure input data consistency. During the preprocessing phase, the financial terminology is expanded using a domain word vector library, and the corpus is uniformly mapped into a customized financial vocabulary space, thereby improving the semantic expression stability of the subsequent embedding layer. Long texts are then segmented and divided into subsequences at the sentence level, and a sliding window mechanism is used to control the sequence length to ensure that it meets the maximum input limit of the Transformer while preserving semantic continuity [14,15]. After text preprocessing, all corpora are encoded into token sequences and mapped one-to-one with the original voucher entries through a hierarchical indexing mechanism, providing both semantic and positional constraints for the subsequent embedding representation construction.

EMBEDDING MODELING AND SEMANTIC DEPENDENCY EXTRACTION

The pre-trained Transformer model used in this paper is based on the BERT-base architecture and undergoes domain-adaptive pretraining on a large-scale financial corpus built internally. The initial weights are not directly adopted from the publicly available BERT weights. Instead, pre-training is performed on top of the general BERT using financial domain texts containing 230 million words, including corporate financial reports, accounting vouchers, invoices, and audit logs, to enhance the model's ability to capture financial terminology and semantic structures. In the specific task phase, the model employs a phased fine-tuning strategy: first, mid-term fine-tuning is performed on a general financial classification task, followed by final fine-tuning for an end-to-end process optimization task. During fine-tuning, a learning rate warmup (10%) and cosine annealing scheduling are used, with a batch size of 32 and 15 training epochs, combined with an early stopping mechanism to prevent overfitting. The pre-training strategy adopted in this paper

does not train the Transformer model from scratch, but uses the publicly released general BERT-base model (from the Hugging Face Transformers library) as the initial weights, and then continues pre-training on this basis using an internally built 230 million-word financial domain corpus. This strategy effectively alleviates the problem of relatively limited domain data scale, while preserving the generalization ability of a general language model, and enhances semantic sensitivity to financial terminology, voucher structure, and audit logic through domain-adaptive fine-tuning. This two-stage training paradigm has been widely used in low-resource domain modeling, significantly improving the performance of downstream tasks while ensuring model stability.

During the model construction phase, the Transformer, based on the BERT (Bidirectional Encoder Representations from Transformers) architecture, is selected as the core semantic modeling framework and pre-trained on a large-scale financial corpus. The input sequence first passes through an embedding layer for word vectorization and positional encoding mapping, converting each token into a d-dimensional vector representation. The embedding layer employs a triple encoding scheme, comprising word embedding, segment embedding, and positional embedding, to simultaneously preserve semantic features and sequence structure. The model's core structure utilizes a multi-head self-attention architecture, with each layer modeling global semantic dependencies by calculating weighted correlations between query, key, and value matrices. Cross-sentence dependencies in the financial corpus, such as semantic associations such as "debit and credit corresponding items" and "budget and execution difference" are captured and quantified through the dynamic distribution of multi-head attention weights, resulting in a high-dimensional semantic coupling representation in the vector space.

During the semantic dependency identification phase, the model's final hidden state is fed into the dependency parsing module as a high-dimensional semantic representation. This module uses a bidirectional attention matching mechanism to calculate the association weights between voucher elements and extracts the semantic dependency matrix through matrix decomposition and layer normalization. The system determines the logical paths between voucher entries based on the attention weights and uses these paths as the initial edge set for the financial semantic network. To enhance the model's stability in multi-source text scenarios, gradient clipping and learning rate warm-up strategies are employed to control the convergence speed of the training process. A layer-wise adaptive rate scaling optimization algorithm is also applied to ensure that the parameter update speed of different layers is consistent with the depth of semantic extraction.

Figure 1 illustrates the performance and semantic representation features of the Transformer-driven financial text embedding model during training. In Figure 1(a), the X-axis represents the number of training epochs, and the Y-axis represents the loss value on a logarithmic scale. The blue curve represents the training loss, and the orange curve represents the validation loss. The training and validation losses shown in Figure 1(a) originate from the multi-task joint fine-tuning stage, which includes three sub-tasks: financial text classification (e.g., voucher type recognition), named entity recognition (NER), and semantic dependency prediction. The total loss is the weighted sum of the cross-entropy losses of each sub-task (weights tuned on the validation set), supplemented by label smoothing (ϵ=0.1) to improve generalization ability. After moving average smoothing, a continuous downward trend is observed, with both the training and validation losses decreasing from approximately 2.5 to close to 0.05. This indicates that the model has converged well and avoided overfitting as it gradually learns the semantic features of financial vouchers, invoices, and reports. Figure 1(b) uses PCA (Principal Component Analysis) to reduce the high-dimensional financial text embedding to a two-dimensional space. The X-axis represents principal component 1, and the Y-axis represents principal component 2. Different colors represent different categories of financial text, such as invoices, vouchers, reports, contracts, and audits. In the two-dimensional space, each category exhibits relatively independent clustering, indicating that the learned embedding representation captures certain semantic differences between different types of financial text, providing a reliable semantic foundation for subsequent cross-source data fusion, process optimization, and risk identification.

CROSS-SOURCE DATA FUSION AND CONTEXTUAL RELATIONSHIP MODELING

MULTIMODAL INPUT STRUCTURE CONSTRUCTION AND ALIGNMENT

Before multimodal fusion, structured financial data (such as amounts, dates, account codes, and voucher numbers) is transformed into vector representations using a customized embedding strategy. Specifically, numerical fields (such as amounts) are logarithmically transformed and Min-Max normalized, then mapped to the same dimension (768 di-mensions) as the text embedding through a trainable linear projection layer; categorical fields (such as account codes and department IDs) are mapped to dense vectors through an embedding lookup table. All structured fields are also appended with modal-type embeddings to distinguish their origin, and these are concatenated with positional encodings before being input into the account encoder subnetwork of the multimodal Transformer [16,17]. This design ensures that structured data retains its original semantics while achieving semantic alignment with unstructured text in a unified vector space. To clearly illustrate the multimodal input structure, let's take a procurement payment process as an example: the structured data includes the amount (150,000.00), date (2024-11-05), supplier code (V-8832),

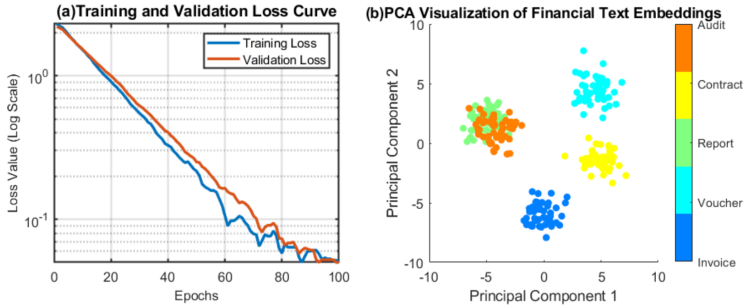

FIGURE 1. Visualization of the Financial Text Embedding Model
Figure 1(a). Training and Validation Loss Curves
Figure 1(b). PCA Visualization of Financial Text Embedding

and account code (6002-12); the unstructured data is the invoice remarks "Emergency equipment procurement, contract number CT-2024-089". The structured part is embedded to form a tensor of shape [4, 768], and the unstructured text is segmented and encoded by BERT to form a tensor of shape [18, 768]. Both are input into the two encoders of the multimodal Transformer and fused at the cross-attention layer, ultimately outputting a unified context-aware representation.

In financial process scenarios, to achieve unified modeling of structured account data and unstructured text information, the two types of data are first subjected to feature alignment and modality mapping. The structured data portion comes from the accounting system's account details, voucher entries, amount fields, and account codes; the unstructured portion includes business descriptions, report notes, and invoice notes. Through numerical normalization, field indexing, and embedding mapping, the structured fields are converted into fixed-dimensional vector representations [16,17]. Each financial field is encoded as a composite feature vector consisting of a numerical embedding and a categorical embedding. The numerical embedding is scale-normalized by a linear mapping layer, and the categorical embedding is semantically encoded by a trainable parameter matrix. The unstructured text is then passed through a pre-trained Transformer encoder to generate a context-dependent vector. A sentence-level semantic center representation is then obtained through word-level attention weighting. To achieve consistent input format across modalities, a feature concatenation strategy is used to align the structured vector with the text semantic vector in terms of feature dimensions. After normalization and positional encoding, the vector is input into a multimodal Transformer architecture. The architecture consists of two independent encoder sublayers, processing account modality and text modality inputs respectively, and enabling information exchange in the subsequent cross-attention stage.

CROSS-ATTENTION FUSION AND CONTEXT-DEPENDENT ENHANCEMENT

During the multimodal feature fusion stage, a cross-attention mechanism is used to model the dynamic associations between structured data and textual semantics. Specifically, the attention weight matrix is calculated, using the hidden representation of the account modality as the query and the hidden representation of the text modality as the key and value. This weight matrix reflects the correlation strength between the account field and the textual semantics. A fused representation vector is generated through a weighted summation, achieving semantic alignment. This process is performed in parallel across multiple attention heads, enabling the model to capture complex dependencies between fields from different semantic dimensions. The outputs of each attention head are linearly transformed and concatenated into a unified feature representation, with residual connections and layer normalization ensuring gradient stability. To prevent information redundancy and modality shift, a gating control mechanism is applied after the fusion layer. This adjusts the proportion of cross-modal information inflow based on the similarity distribution between features, thereby suppressing the interference of low-correlation noise features on the overall semantics.

To further strengthen contextual semantic dependencies, a bidirectional context aggregation strategy is adopted. Specifically, a reverse attention layer is applied after the cross-attention output, establishing an inverse association relationship using the text modality as the query and the account modality as the key and value. This structure ensures bidirectional information flow between the two modalities, allowing financial actions or constraints implicit in the text to be reflected in the account semantic space. Through multiple layers of cross-stacked data, the model can globally learn the mapping between account fields and semantic descriptions, forming a multi-layered, context-dependent representation.

Figure 2 demonstrates the performance of Transformer in cross-source financial data fusion. In Figure 2(a), the X-axis represents the sample index and the Y-axis represents the eigenvalue. The "feature value" shown in Figure 2(a) is the value of each sample on the 64th dimension of the fusion vector, used to illustrate the dynamic change trend of multimodal input and fusion output in a single dimension. This dimension does not represent principal components or norms, but is used to demonstrate the smoothness and information integration characteristics of the fusion process. The blue curve represents structured account data, the orange curve represents unstructured text data, and the green curve represents the fused eigenvalue. The fused curve smoothly

combines the changing trends of the two data types, with eigenvalues ranging from 3.0 to 6.0. This demonstrates that the model balances the semantic features of account and text when integrating multi-source information, improving the stability and continuity of data representation. Figure 2(b) shows a cross-attention weight heatmap. The X-axis represents the text modality position and the Y-axis represents the account modality position. The color depth reflects the strength of the attention weight at the corresponding position. High weights are observed near the diagonal line, while some discrete high values exist. This indicates that the model primarily focuses on the corresponding position when aligning account and text features but also identifies semantic associations across positions.

DYNAMIC PROCESS OPTIMIZATION AND POLICY GENERATION MODULE

Traditional optimization algorithms struggle to handle the dynamically changing semantic context and risk states in financial processes. For example, whether to skip a node in an approval chain depends not only on pre-defined rules but also on the current transaction semantics (such as whether related-party transactions are involved) and real-time risk scores. Reinforcement learning agents, through interaction with the environment, learn to make long-term optimal decisions under uncertainty and multi-objective constraints.

POLICY NETWORK STRUCTURE AND STATE REPRESENTATION DESIGN

To achieve dynamic optimization of financial processes, a reinforcement learning policy network is first constructed based on the semantic feature vectors output by the Transformer model. The system represents each financial process instance as a sequence of states, where each state is composed of multidimensional semantic features, historical operation records, and business constraints. By performing dimensionality compression and feature normalization on the Transformer hidden layer output, a set of state vectors is generated to describe the current financial operating environment. The action space is defined as a set of executable process adjustment instructions, including operation types such as document path modification, approval node jump, exception backtracking, and automatic verification. Each action is represented by a separate code and mapped to the state space to form a state-action pair for subsequent policy updates. The state space of the reinforcement learning policy network consists of three parts: (1) the semantic embedding vector of the current task (from the Transformer output); (2) the encoding of historical operation trajectories (aggregating past actions through LSTM); and (3) the business constraint vector (such as budget limits and approval level restrictions). The action space is defined as a set of discrete financial process operation instructions, including 12 types of actions such as skipping an approval node, triggering automatic verification, backtracking to the previous voucher, merging similar tasks, and calling external risk control modules. The reward function integrates three indicators: process completion time (negative reward, encouraging acceleration), abnormal avoidance score (positive reward, based on feedback from the anomaly detection module), and resource consumption cost (negative reward, such as the number of times manpower is called). To ensure the comparability of different dimensional indicators in the reward function, time, anomaly avoidance score, and cost are all processed by Min-Max normalization before participating in the calculation, and uniformly mapped to the [0, 1] interval. Among them, the time indicator is converted into an efficiency gain form—that is, the shorter the original delay, the higher the normalized value, thus aligning with the direction of other positive indicators; the anomaly avoidance score and cost are normalized in the conventional way.

The "skip approval node" and other operation instructions generated by the reinforcement learning policy network are suggested path adjustments under the constraints of the compliance rule engine. The system's built-in business rule base performs feasibility filtering on all actions: only when the target node and the current node have the same approval authority, and historical data shows that the skipping behavior has a human approval rate of over 95% in similar scenarios, is the action allowed to be included in the policy space. In addition, all high-risk actions must trigger a manual review process to ensure audit traceability. Therefore, the RL module essentially explores the most efficient auxiliary decision-making path within the compliance boundaries, rather than replacing manual approval.

ADAPTIVE OPTIMIZATION MECHANISM AND STRATEGY GENERATION PROCESS

During the strategy generation phase, the reinforcement learning network and the Transformer semantic model maintain parameter decoupling, ensuring that feature extraction and strategy evaluation do not interfere with each other through independent training. The model receives the real-time input of the financial task state, converts it into a semantic vector, and inputs it into the strategy network. The network then outputs the optimal action decision based on the learned strategy distribution [18,19]. To improve the generalization performance of the strategy in multiple scenarios, an experience replay mechanism is applied to store historical decision samples in a memory pool and sample and replay them according to time priority and profit difference, thereby enhancing the model's learning ability for rare scenarios.

During the actual operation phase, the system updates policy parameters in an iterative loop. After each process execution, the model updates the state-value function based on feedback signals and recalculates the action probability distribution through the policy improvement phase. As the number of iterations increases, the policy network gradually

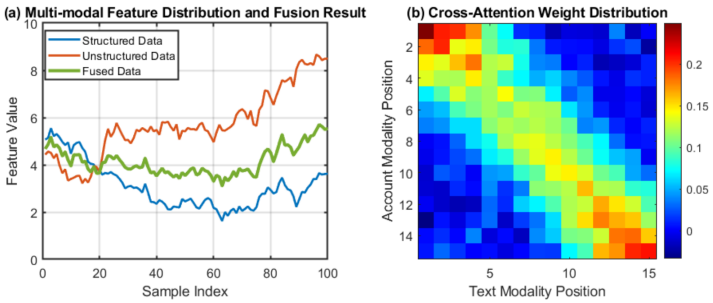

FIGURE 2. Visualization of cross-source data fusion
Figure 2(a). Multimodal feature distribution and fusion results
Figure 2(b). Cross-attention weight distribution

approaches the optimal solution under multi-dimensional objectives. This mechanism enables the system to automatically adjust the financial processing path based on historical execution records, reducing redundant steps and improving process throughput efficiency. The final output policy mapping table implements a closed-loop control of state, action, and benefit, providing a sustainable and optimized decision-making basis for subsequent anomaly detection modules and knowledge graph construction.

Figure 3 illustrates the performance of a Transformer-based financial process dynamic optimization model during reinforcement learning training. In Figure 3(a), the X-axis represents the training rounds, and the Y-axis represents the average reward. The curve rises from approximately 0 and gradually converges to around 50, demonstrating that the model optimizes the financial processing path through continuous iteration, gradually stabilizing the strategy and smoothly increasing the reward. This reflects the adaptive ability of process adjustments and the effectiveness of the reinforcement learning strategy. In Figure 3(b), the Xand Y-axes represent the two dimensions of the state space, respectively, and the Z-axis represents the estimated state value function. The colored surface displays the expected value under different state combinations, revealing the fluctuation distribution of the value function in the state space. The high-value region corresponds to the two-dimensional state coordinate system in Figure 3(b), obtained by applying t-SNE dimensionality reduction (perplexity=30, n_iter=1000) to the high-dimensional state vector (dimension 768), aiming to visualize the local structure of the state-value function. The Z-axis represents the state-value function estimated by the policy network.

This surface reflects the expected cumulative reward distribution corresponding to different state combinations in the reduced-dimensional projection space, with high-value regions corresponding to efficient and low-risk process paths. It should be noted that this figure is a qualitative analysis tool and not for quantitative inference. The model identifies efficient financial process paths, while low-value regions correspond to inefficient or abnormal process paths.

ANOMALY DETECTION AND RISK WARNING GENERATION

ATTENTION DISTRIBUTION DEVIATION DETECTION AND FEATURE EXTRACTION

In the anomaly detection phase, the focus is on the consistency and stability of multi-head attention within a single layer. Specifically, for each input sample, we calculate the distribution similarity among the attention heads in the top-level Transformer layer. Normal samples typically exhibit high consistency—multiple attention heads collaboratively focus on task-related semantic regions; while anomaly samples often lead to significant divergence among attention heads, with some heads being attracted by irrelevant or noisy tokens, thus disrupting internal consistency. Each input sample xi is encoded by the Transformer to generate a multi-layer hidden state H'=[h1',h2',...,hn'], where L represents the number of layers. After normalerate the attention weight matrix Al$\in R^{n\times n}$ of each layer, the difference matrix between adjacent layers is calculated, as shown in Formula (1):

$$Dl = \|\|Al - Al - 1\|\|_F \tag{1}$$

$\|\cdot\|$F represents the Frobenius norm, which is used to measure the overall deviation of attention distribution between layers. When Dl exceeds the threshold τ, the input corresponding to the sample is marked as a potential abnormal signal. In order to reduce the sensitivity of the model to local fluctuations, a sliding window smoothing mechanism is applied to perform mean filtering on the deviation values of consecutive layers to obtain a stable abnormal indicator sequence. This indicator sequence is then concatenated with the semantic embedding vector of the input sample for use in the subsequent risk clustering analysis stage. An unsupervised detection strategy is adopted in model training to reduce dependence on abnormal sample labels. Considering that the attention weight distribution typically exhibits sparsity and heavy-tailed characteristics, which does not conform to the Gaussian assumption, this paper adopts an empirical quantile threshold method.

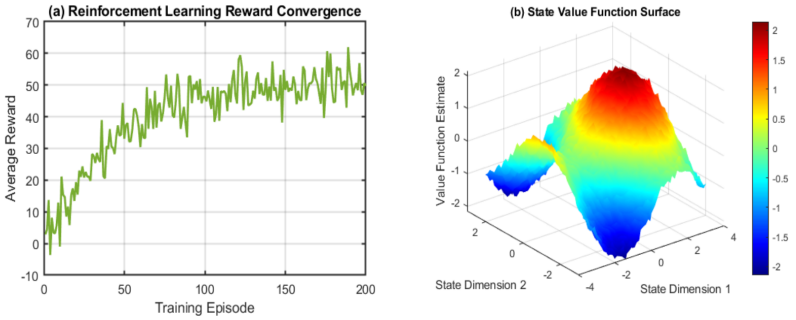

FIGURE 3. Dynamic Process Optimization Visualization
Figure 3(a). Reinforcement Learning Reward Convergence Curve
Figure 3(b). State-Value Function Surface

CLUSTER ANALYSIS AND RISK WARNING GENERATION

After obtaining a set of anomaly candidates, a high-dimensional clustering algorithm is used to aggregate and analyze anomaly features to identify different types of risk patterns. The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm is used for clustering, with the input features being the anomaly embedding vector and the corresponding attention bias indicator. By calculating the Euclidean distance matrix between samples, dense regions in the feature space are clustered, and the number of anomaly categories is automatically determined based on cluster density. Noise samples—outliers not assigned to a cluster—are identified as high-risk individual events and recorded in the risk log.

The risk warning generation module establishes a risk evolution curve based on the clustering results and time series information. The system calculates the growth rate of the anomaly cluster in a continuous period and generates a trend indicator in the form of time difference. If the growth rate exceeds the set threshold, the model automatically triggers the risk warning signal and marks the risk node in the knowledge graph. The warning report is automatically summarized by the system, and the content includes the anomaly category, impact range, time distribution and related process node information. The report template is automatically generated by the Transformer decoding layer to generate a text description and output to the financial monitoring terminal in a structured manner for use by the decision module. Figure 4 illustrates a time series analysis of financial risk warnings, used to assess the system's ability to monitor and warn of potential financial anomalies. The horizontal axis represents time (days), and the vertical axis represents the risk score. The overall trend incorporates cyclical financial activity, daily fluctuations, and seasonal patterns, making risk fluctuations more realistic in real-world business scenarios. Four key anomaly events are highlighted in the figure: On day 25, transaction anomalies showed a sudden spike, with the risk score instantly increasing to over 0.85; on day 42, approval anomalies persisted, with the score remaining high; on day 67, voucher errors are single-point anomalies, with a score of approximately 0.64; and on day 83, budget overrun risk emerged. The yellow dashed line in the figure represents the warning threshold of 0.75, and the red dashed line represents the critical threshold of 0.85. High-risk and warning areas are shaded light red and light yellow, respectively, to reflect the risk level. Figure 4 demonstrates that the system is able to promptly identify high-risk events and issue warnings, with transaction anomalies having the highest score, demonstrating the model's sensitivity and reliability in detecting anomalies in key financial processes.

INTELLIGENT PROCESS KNOWLEDGE GRAPH CONSTRUCTION

Although there is existing research on knowledge graph construction based on Transformer, the innovation of this paper lies in the triple coupling mechanism: (1) semantic dependency directly drives graph edge generation (rather than just being used for entity recognition); (2) the anomaly detection module uses attention distribution deviation as the trigger signal for risk nodes in the graph; and (3) the optimization path of the reinforcement learning strategy is fed back to the knowledge graph in real time, realizing the dynamic evolution of the graph structure. This paper unifies semantic modeling, dynamic decision-making and risk perception graph in an end-to-end framework, supporting closed-loop optimization.

NODE EXTRACTION AND RELATIONSHIP MODELING

In the knowledge graph construction phase, the system generates a structured set of initial nodes and edges based on the semantic dependency matrix output by the Transformer model. First, key semantic entities are extracted from the financial corpus after semantic modeling and multimodal fusion, including semantic units such as voucher number, transaction object, fund account, approval node and business department [20]. The corpus is multi-layer parsed through named entity recognition (NER) and dependency syntax parsing results, and the semantic units are mapped to graph nodes. To ensure the Precision of node classification, a dual-channel entity alignment mechanism is adopted: one channel calculates the node semantic distance based on word vector similarity, and the other channel evaluates the node dependency strength at the sentence level based on the context attention weight. By jointly optimizing the loss

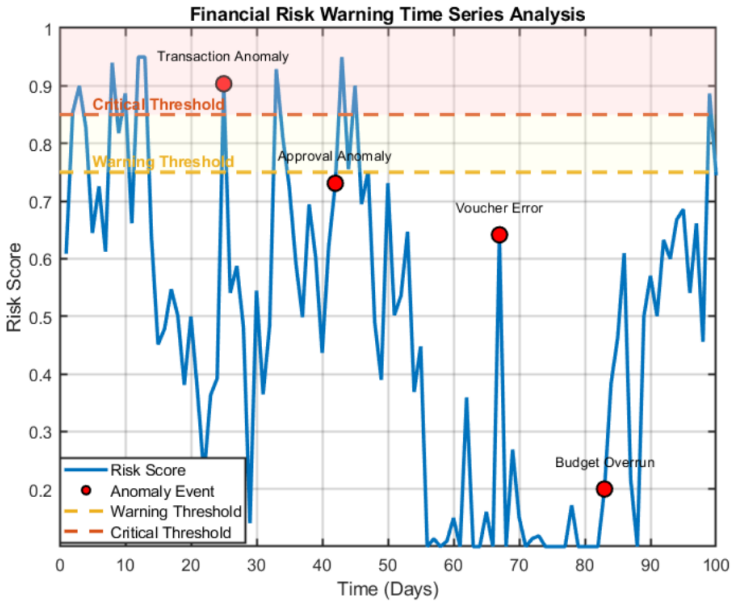

FIGURE 4. Risk warning time series analysis https://doi.org/10.31881/TLR.2026.12472 12486

function to minimize the deviation of the two channels, the node classification is kept consistent in both semantics and context.

In the relationship modeling phase, the connection relationship between nodes is determined based on the dependency strength matrix output by the Transformer attention layer [21]. For any two nodes i and j, their semantic dependency r_{ij} is calculated, as shown in formula (2):

$$r_{ij} = \frac{1}{L}\sum_{l=1}^{L} A_{ij}^{l} \tag{2}$$

A_{ij}^{l} represents the attention weight of layer l, where L is the number of Transformer layers. If r_{ij} exceeds an adaptive threshold θ, the system establishes a directed edg e between the two nodes and labels the rela-tionship type , such as "fund flow," "approval association,"or "credentia l reference." The semantic labels of all nodes and edges ar e stored in a relational database for subsequent structure lea rning and inference tasks. To eliminate redun-dant connec tions and incorrect relationships, the system performs ed ge normalization and local consistency correc-tion on the generated graph structure. A maximum entropy model is u sed to constrain edge distribution, ensuring the sparsity an d completeness of the graph structure.

GRAPH OPTIMIZATION AND VISUALIZATION GENERATION

The system uses the TransE model to encode the knowledge triple (h, r, t) by minimizing the energy function, as shown in formula (3):

$$E(h,r,t) = \|h + r - t\|_2 \tag{3}$$

This achieves a low-dimensional continuous representation of semantic relationships between nodes. After principal component reduction, the embedding vector is fed into a graph convolutional network (GCN) for structural optimization. The GCN aggregates the semantic features of adjacent nodes to enhance local connectivity, allowing similar financial process nodes to form a decomposable structure in the vector space. This mechanism enables the system to automatically form hierarchical process relationships based on semantic logic without relying on manual annotation.

III. PERFORMANCE VERIFICATION OF INTELLIGENT OPTIMIZATION MODEL

The data used in this study comes from the financial system logs of three large manufacturing enterprises from January 2020 to June 2024, covering eight core processes including general ledger, accounts payable /receivable, expense reimbursement, and procurement payments. The raw data includes structured fields (such as voucher number, amount, account code, approver ID, and timestamp) and unstructured text (such as invoice remarks, contract summaries, and audit opinions). All sensitive in-formation (such as bank account numbers, ID card numbers, and full supplier names) was generalized or hashed using a rule-based de-identification engine to ensure compliance with GDPR and the "Guidelines for the Classification of Financial Data Security". Annotation was performed by five CPA-qualified financial experts under a double-blind mechanism, classifying anomalous samples into three levels (abnormal amounts, process breaks, logical contradictions, etc.), achieving a Kappa consistency coefficient of 0.87. The training, validation, and test sets were divided in a 7:1.5:1.5 ratio and strictly grouped by enterprise ID—meaning all records from the same enterprise

appear in only one set to prevent data leakage.

A. FINANCIAL SEMANTIC RECOGNITION PRECISION

Financial semantic recognition precision measures the model's Precision in recognizing financial text and document elements. During the evaluation, annotated document entries, invoice text, and report descriptions are first extracted from the test set as a standard reference. The semantic embeddings and text parsing results generated by the model are then matched against the standard data, and the number of correctly matched entries is counted. The Precision value is then calculated as the proportion of correctly recognized entries out of the total number of entries. During the evaluation, stratified statistics are performed on different financial entity categories to analyze the differences in model recognition performance across different dimensions, such as account type, amount field, and approval node. The experiment is repeated across multiple batches of data to obtain the average Precision and standard deviation.

Figure 5 shows the performance of the financial semantic recognition model under different entity types and different models. In Figure 5(a), the x-axis represents the financial entity type, including accounting subject, amount field, date information, transaction object, voucher number, invoice number, approval node, and department code, and the y-axis represents the recognition Precision (%). As can be seen, the recognition Precision of the amount field and voucher number is the highest, reaching 96.8% and 97.3% respectively, indicating that the model performs well in extracting key fields. The transaction object and approval node are slightly lower, at 88.7% and 90.1%, respectively, reflecting the challenges of semantic recognition for diverse entity types. In Figure 5(b), the x-axis represents different models, including rule engines, traditional ML (Machine Learning) methods, RNN (Recurrent Neural Network), CNN (Convolutional Neural Network), BERT, and the method proposed in this paper, and the y-axis also represents the Precision (%). The comparison results show that the proposed method outperforms traditional methods in most financial document types.

B. PROCESS EXECUTION DELAY

Process execution latency evaluates the real-time performance of the model in financial task automation. First, the system records the start and end times of different financial processes, including document review, report generation, and approval flow processing. The model's processing time is compared with that of traditional automation systems. During this evaluation step, the average latency values for large-scale batch processing tasks and high-frequency single-transaction tasks are calculated to examine the model's processing efficiency at different data scales. By analyzing the distribution of latency data and outliers, the model's response stability in dynamic financial scenarios is determined. The "traditional automation system" referred to in this paper is the widely deployed RPA platform (UiPath + regular expressions + static approval flow) based on rule engines and template matching, whose semantic understanding capabilities are limited to keyword matching and fixed field extraction. All experiments were run on a server equipped with 2×NVIDIA A100 80GB GPUs, 512GB RAM, and an Intel Xeon Gold 6348 CPU, running Ubuntu 22.04 LTS. Latency measurements were performed using a warm-boot mode (preloading the model to GPU memory), with a batch size of 32, and network latency was ignored (internal network communication).

Figure 6 shows the performance and distribution characteristics of the execution delay of the financial process. In Figure 6(a), the horizontal axis represents the number of batch processing tasks, and the vertical axis represents the processing delay time (ms). The curve shows that the delay of the traditional system increases rapidly when processing large-scale tasks, while the delay of the proposed method is significantly reduced under the same task volume. For example, when processing 1000 tasks, the delay of the traditional system exceeds 1000ms, while the delay of the proposed method is only about 500ms, which reflects the efficiency of the optimization method. Figure 6(b) is a box plot of the delay distribution. The horizontal axis represents the eight types of financial processes, and the vertical axis represents the delay time (ms). By comparison, it can be seen that the median delay and fluctuation range of each process under the optimization method are significantly lower than those of the traditional system, and the fluctuation range is also reduced. The "500 milliseconds" in Figure 6 refers to the average end-to-end latency for processing 1000 tasks. This performance is due to pipeline parallelism and model distillation— this paper distills the original BERT-large into a 6-layer small Transformer (parameter count $\approx$ 22M) and enables TensorRT acceleration during the inference stage. Therefore, 0.5ms/task is achievable after engineering optimization.

C. ANOMALY DETECTION RECALL RATE

Anomaly detection recall is used to verify the model's ability to identify potential financial anomalies. The evaluation process begins by constructing a test set containing known anomalous transaction samples, including those with unusual amounts, broken approval chains, and budget deviations. The anomalous transactions flagged by the model are compared against the known anomaly list, and the proportion of correctly identified anomalous samples compared to the total number of anomalous samples is calculated. The entire evaluation process includes independent statistics for different anomaly types and analyzes the model's ability to

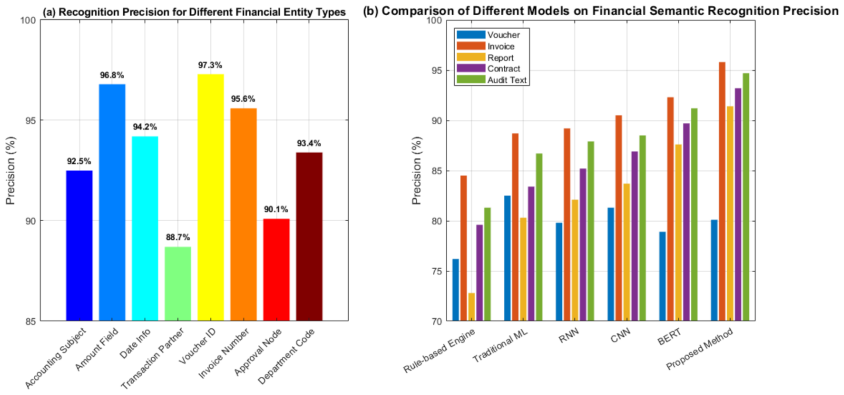

FIGURE 5. Financial Semantic Recognition Precision Analysis
Figure 5(a). Recognition Precision of Different Financial Entity Types
Figure 5(b). Comparison of Financial Semantic Recognition Precision of Different Models

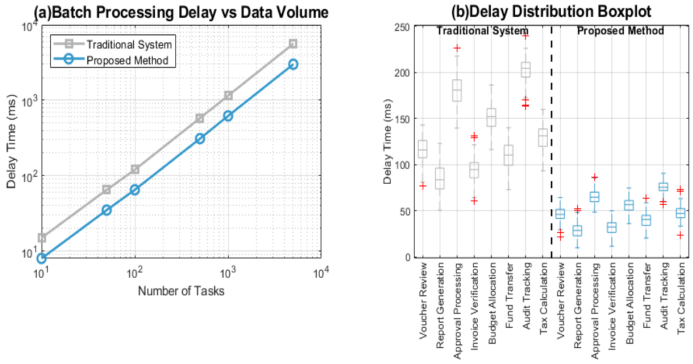

FIGURE 6. Process Execution Latency Analysis
Figure 6(a). Batch Processing Latency Variation with Data Volume
Figure 6(b). Process Latency Distribution Boxplot

identify rare anomaly samples. Through multiple rounds of testing, the overall recall rate and the recall rate for each anomaly category are obtained. "Traditional methods" in Table 1 specifically refers to the ensemble results of the following three types of baseline models: (1) CRF-based sequence labeler; (2) rule + dictionary anomaly detection module; (3) fixed-path BPMN process engine. All baselines are run on the same dataset and hardware environment to ensure fair comparison. Although the anomaly detection module does not use anomaly labels during the training phase (i.e., unsupervised), threshold calibration and model selection are performed on a separate validation set. This validation set contains a small number of manually labeled anomaly samples (<1%), which are only used to determine the 95th percentile threshold τ of attention bias and do not participate in model parameter updates.

Table 1 compares the recall rates of the models for different anomaly types, verifying the completeness of the anomaly detection module in identifying potential financial anomalies. The proposed method significantly outperforms traditional methods for key anomaly types, including amount anomalies, broken approval chains, and budget deviations. The recall rate for amount anomalies reaches 96.8%, an 8.4 percentage point increase over the traditional method's 88.4%. The recall rate for broken approval chains is 92.5%, an 8.3% improvement, and the recall rate for budget deviation is 94.3%, a 7.6% improvement. For duplicate transactions and voucher discrepancies, the proposed method achieves recall rates of 98.2% and 95.4%, respectively, 5.7% and 6.7% higher than the traditional method, respectively. Overall, the average recall rate for all anomaly types is 95.72%, a 7.25 percentage point improvement over the traditional method's 88.47%. The p-values for all anomaly types are all less than 0.001, indicating a highly statistically significant difference. These data demonstrate that the proposed method offers greater completeness and reliability in identifying financial anomalies, effectively covering both common and rare anomalies. The p-values in Table 1 were calculated using the McNemar test, a method applicable to paired comparisons of two classifiers on the same test set. The null hypothesis was that the two methods performed similarly, and the alternative hypothesis was that the method presented in this paper was superior. All tests rejected the null hypothesis at the α=0.001 level.

To comprehensively evaluate the anomaly detection performance, we supplemented the report with complete metrics on the test set (positive to negative sample ratio of approximately 1:19): in addition to a recall of 95.72%, the precision was 93.5% (see Table 2), the F1 score was 94.6%, and the false positive rate (FPR) was 4.1%. In high-risk categories (such as monetary anomalies), the F1 score reached 95.5%, indicating that the model effectively controls

TABLE 1. Anomaly detection recall performance evaluation

Anomaly Type	Sample Size	Proposed Method Recall (%)	Traditional Method Recall (%)	Improvement (%)	p-value
Amount Anomaly	1,250	96.8	88.4	8.4	< 0.001
Approval Chain Break	876	92.5	84.2	8.3	< 0.001
Budget Deviation	954	94.3	86.7	7.6	< 0.001
Duplicate Transaction	672	98.2	92.5	5.7	< 0.001
Voucher Mismatch	1,032	95.4	88.7	6.7	< 0.001
Tax Rate Error	587	97.1	90.3	6.8	< 0.001
Average	895	95.72	88.47	7.25	< 0.001

TABLE 2. Risk warning generation performance evaluation

Anomaly Type	True Positives	Missed Alerts	False Positives	Precision (%)	Average Response Time (hours)
Amount Anomaly	245	8	15	94.2	1.2
Approval Delay	187	12	18	91.2	2.5
Budget Deviation	213	11	14	93.8	1.8
Duplicate Transaction	154	6	11	93.3	2.8
Voucher Anomaly	176	9	12	93.6	3.2
Tax Rate Error	189	5	10	95.0	1.9
Total / Average	1,164	51	80	93.57	2.2

false positives while maintaining a high detection rate. These metrics collectively validate the system's practicality in risk-sensitive scenarios.

D. SYSTEM ADAPTIVE SCORING

The system's adaptive scoring system evaluates the model's adaptability in dynamic process adjustment and strategy generation. The evaluation process first records the number of times the model adjusts the financial process path and its success rate under different historical decision feedback conditions. Then, based on the process optimization results, it calculates task completion efficiency, exception avoidance rate, and path adjustment frequency to form a comprehensive adaptive score. The scoring process simulates various business scenarios and emergencies to analyze the model's strategy adjustment capabilities and stability in multiple scenarios. Finally, the scoring results are compared with those of a traditional static process system, providing quantitative metrics for verifying the adaptive performance of the reinforcement learning policy network.

The system adaptive score is defined as a weighted linear combination: $S = w1 \cdot E + w2 \cdot A + w3 \cdot P$, where E is the task efficiency (1 - actual time spent / baseline time spent), A is the abnormal avoidance rate (1 - proportion of high-risk actions triggered), and P is the path adjustment frequency (number of effective adjustments per unit time). All three indicators are normalized to [0, 1], and the weights w1=0.4, w2=0.35, w3=0.25 are determined by five CFOs using the Analytic Hierarchy Process (AHP).

In Figure 7, "traditional static process system" refers to a BPMN engine that does not integrate semantic understanding and adaptive decision-making. Its process path is fixed at the time of deployment and cannot be dynamically adjusted according to transaction semantics or risk status. It only supports simple jumps triggered by preset rules.

Figure 7 illustrates the system's adaptive scoring performance in different business scenarios and the contribution of each optimization metric to the overall score. In Figure 7(a), the X-axis represents eight financial and business scenarios, including normal operations, month-end settlement, quarter-end audit, annual budget, system upgrade, personnel change, policy change, and market fluctuation, while the Y-axis represents the adaptive score. It can be seen that the traditional system's score generally ranges from 60–75 points, while the proposed method achieves significant improvements in all scenarios, reaching scores of 88–95 points, demonstrating the adaptability and reliability of the optimized system in complex business situations. Figure 7(b) shows the contribution of different optimization metrics to the system's adaptive score. Task efficiency, path optimization, and anomaly avoidance are the primary drivers, contributing approximately 25%, 23%, and 19%, respectively, while energy efficiency and scalability contribute less. This indicates that system performance improvements primarily rely on core process optimization and anomaly management strategies. The "contribution" in Figure 7(b) refers to the weight percentage of each indicator in the total score, not the SHAP or ablation result, but a decomposition based on the aforementioned preset weights.

E. RISK WARNING GENERATION PRECISION

During the evaluation step, the risk reports generated by the model are compared with historical anomaly records to count the number of correct warnings, missed warnings, and false alarms. The Precision of warnings for various types of anomalies, such as document anomalies, approval delays, and funding deviations, is further analyzed. Experiments are repeated across multiple batches of data to calculate the overall warning Precision and the distribution of Precision across different anomaly types. This metric provides financial managers with an assessment of the system's usability in real-world risk monitoring scenarios, directly reflecting the synergy between the knowledge graph and anomaly detection modules.

Table 2 shows the performance evaluation results of the risk warning generation module on different types of financial anomalies, focusing on the model's Precision and response efficiency in real-world business scenarios. The

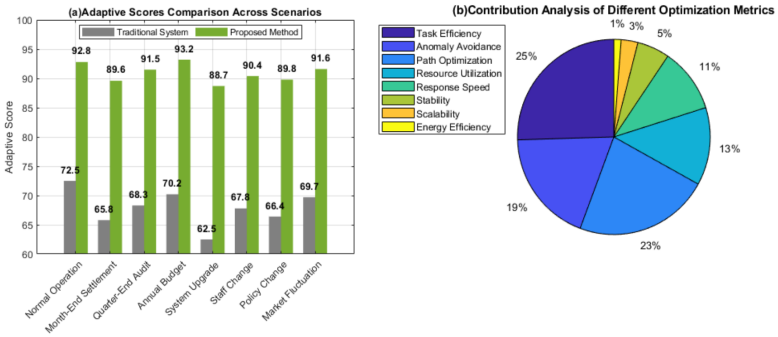

FIGURE 7. Visualization of the system's adaptive score
Figure 7(a). Comparison of adaptive scores in different scenarios
Figure 7(b). Analysis of the contribution of different optimization metrics

model's overall Precision reaches 93.57%, indicating a high degree of consistency between the generated warning information and the actual anomalies. The highest Precision for tax rate errors reaches 95.0%, and the Precision for amount anomalies reaches 94.2%, demonstrating that the system is particularly reliable in identifying anomalies in high-value or critical financial processes. The average response time is 2.2 hours, with amount anomalies and budget deviations generating warnings in just 1.2 hours and 1.8 hours, respectively. This demonstrates the system's ability to issue alerts promptly after critical financial events. Approval delays and voucher anomalies take slightly longer, ranging from 2.5 to 3.2 hours, but remain within acceptable limits. The "average response time" in Table 2 refers to the time interval from when an abnormal transaction enters the system to when the first structured warning report is generated and pushed to the monitoring terminal, encompassing the entire process of model inference, cluster analysis, and report generation. This time is automatically triggered without manual confirmation, reflecting the system's end-to-end alarm latency. The data in Table 2 demonstrates the effective synergy between the knowledge graph and anomaly detection module, efficiently and accurately supporting financial risk monitoring and providing managers with reliable decision-making support. The system also maintains a high level of recognition for rare or complex anomaly types.

IV. CONCLUSION

This paper addresses the semantic understanding and process optimization challenges in enterprise financial process automation by proposing a Transformer-based end-to-end intelligent optimization framework. The study first constructs a financial text embedding model, leveraging a self-attention mechanism to extract semantic features from vouchers, invoices, and reports. A multimodal Transformer is then used to achieve fusion modeling of structured and unstructured data. Fur-thermore, a reinforcement learning policy network is applied for dynamic process optimization. Attention bias detection and cluster analysis are combined to identify anomalies and provide risk warnings. Finally, an intelligent process knowledge graph is constructed based on semantic dependencies, supporting the struc-tured analysis and automated optimization of financial processes. Experimental results demonstrate that this method significantly improves semantic recognition Precision, process execution efficiency, and risk warning Precision. However, this method still suffers from limited adaptability to extremely anomalous samples and delayed graph updates. Future work will focus on optimizing the robustness of anomaly detection, enhancing the dynamic update capabilities of the knowledge graph, and exploring transfer learning strategies across enterprise financial processes to enable a wider range of application scenarios.

AUTHOR CONTRIBUTIONS

Su Zheng designed, collected and analyzed the data, and drafted the manuscript. Su Zheng conducted the study, critically revised the manuscript for important intellectual content, and gave final approval of the version to be published. Su Zheng participated fully in the work, take public responsibility for appropriate portions of the content, and agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] O. Doguc, "Applications of robotic process automation in finance and accounting," Beykent Üniversitesi Fen ve Mühendislik Bilimleri Dergisi, vol. 14, no. 1, pp. 51-59, 2021, doi: 10.20854/bujse.906795.

[2] F. I. Erdi and A. Retnowardhani, "Optimization of Application Testing in Financial Companies Using Robotic Process Automation (RPA)," Economic Reviews Journal, vol. 3, no. 4, pp. 1961–1981, 2024, doi: 10.56709/mrj.v3i4.595.

[3] K. Kalluri, "Integrating Pega's AI-Driven Workflows for End-to-End Process Optimization in Financial Services," North American Journal of Engineering Research, pp. 5(3), 2024, [Online]. Available: https://najer.org/najer/article/view/109.

[4] A. Dalsaniya and K. Patel, "Enhancing process automation with AI: The role of intelligent automation in business efficiency," International Journal of Science and Research Archive, vol. 5, no. 2, pp. 322-337, 2022.

[5] A. Khunger, "Optimizing Payment Gateways in Fintech Using AI-Augmented OCR and Intelligent Workflow," Journal of Electrical Systems, vol. 17, no. 1, Art. no. 10.52783, 2024.

[6] D. Pramod, "Robotic process automation for industry: adoption status, benefits, challenges and research agenda," Benchmarking: an international journal, vol. 29, no. 5, pp. 1562-1586, 2022, doi: 10.1108/BIJ-01-2021-0033.

[7] S. Ren, "Optimization of enterprise financial management and decision-making systems based on big data," Journal of Mathematics, vol. 2022, no. 1, Art. no. 1708506, 2022, doi: 10.1155/2022/1708506.

[8] S. Ahmadi, "A comprehensive study on integration of big data and AI in financial industry and its effect on present and future opportunities," International Journal of Current Science Research and Review, vol. 7, no. 1, pp. 66-74, 2024, doi: 10.47191/ijcsrr/V7-i1-07.

[9] O. Abiola-Adams, C. Azubuike, A. K. Sule, and R. Okon, "Optimizing balance sheet performance: Advanced asset and liability management strategies for financial stability," International Journal of Scientific Research Updates, vol. 2, no. 1, pp. 55-65, 2021, doi: 10.53430/ijsru.2021.2.1.0041.

[10] O. Oluoha, A. Odeshina, O. Reis, F. Okpeke, V. Attipoe, and O. Orieno, "Optimizing business decision-making with advanced data analytics techniques," Iconic Res Eng J, vol. 6, no. 5, pp. 184-203, 2022.

[11] K. Olorunnimbe and H. Viktor, "Ensemble of temporal Transformers for financial time series," Journal of Intelligent Information Systems, vol. 62, no. 4, pp. 1087-1111, 2024.

[12] R. K. Debbadi and O. Boateng, "Optimizing end-to-end business processes by integrating machine learning models with UiPath for predictive analytics and decision automation," Int J Sci Res Arch, vol. 14, no. 2, pp. 778-796, 2025, doi: 10.30574/ijsra.2025.14.2.0448.

[13] A. Song, E. Seo, and H. Kim, "Anomaly VAE-transformer: A deep learning approach for anomaly detection in decentralized finance," IEEE Access, vol. 11, pp. 98115-98131, 2023, doi: 10.1109/ACCESS.2023.3313448.

[14] A. Kasoju and T. Vishwakarma, "Optimizing Transformer Models for Low-Latency Inference: Techniques, Architectures, and Code Implementations," International Journal of Science and Research (IJSR), vol. 14, no. 4, pp. 857-866, 2025, doi: 10.21275/SR25409073105.

[15] W. Lo, C. M. Yang, Q. Zhang, and M. Li, "Increased productivity and reduced waste with robotic process automation and generative AI-powered IoE services," Journal of Web Engineering, vol. 23, no. 1, pp. 53-87, 2024, doi: 10.13052/jwe1540-9589.2313.

[16] S. Joshi, "Implementing gen AI for increasing robustness of US financial and regulatory system," International Journal of Innovative Research in Engineering and Management, vol. 11, no. 6, pp. 175-179, 2025, doi: 10.55524/ijirem.2024.11.6.19.

[17] S. Joshi, "Review of gen ai models for financial risk management," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 11, no. 1, pp. 709-723, 2025, doi: 10.32628/CSEIT2511114.

[18] S. Kannan, "The Role of AI And Machine Learning in Financial Services: A Neural Networkbased Framework for Predictive Analytics and Customercentric Innovations," Migration Letters, vol. 19, no. 6, pp. 985-1000, 2022.

[19] S. K. Rachakatla, Ravichandran SrP, and Machireddy SrJR, "AI-driven business analytics: Leveraging deep learning and big data for predictive insights," Journal of Deep Learning in Genomic Data Analysis, vol. 3, no. 2, pp. 1-22, 2023.

[20] S. Zakaria, S. M. A. Manaf, M. T. Amron, and M. T. M. Suffian, "Has the world of finance changed? A review of the influence of artificial intelligence on financial management studies," Information Management and Business Review, vol. 15, no. 4, pp. 420-432, 2023.

[21] U. Haldar, G. T. Alam, H. Rahman, M. A. Miah, P. Chakraborty, A. S. M. Saimon, et al., "AI-Driven Business Analytics for Economic Growth Leveraging Machine Learning and MIS for Data-Driven Decision-Making in the US Economy," Journal of Posthumanism, vol. 5, no. 4, pp. 932–957, 2025.