Driven Anomaly Detection and Root Cause Analysis in SAP ERP Using Hybrid Neural-Symbolic Systems Hibrit Nöral-Sembolik Sistemler Kullanılarak SAP ERP'de Sürülen Anomali Tespiti ve Kök Neden Analizi

Nagendra Harish Jamithireddy
Jindal School of Management, The University of Texas at Dallas, United States
Email: jnharish@live.com

***Abstract*—The more integrated the modern SAP ERP system becomes, the greater the challenge of ensuring the accuracy, interpretability, and real-time detection of anomalies. Escalating complexities, business logics, and the dynamic nature of cross-functional modules make it ever more necessary to provide seamless explanatory systems that root explain causative analyses of various statistical and machine learning features employed. In this work, a new deep learning hybrid neural-symbolic approach is introduced to detect anomalies and automate root cause analysis in sap ERP systems. This architecture combines semantic rule-based models with deep neural frameworks to not only pinpoint anomalies within diverse operational, financial, and material data but also map their causal links within SAP modules and extend their reach beyond inter-ERP systems. A comprehensive evaluation on a hybrid dataset containing authentic SAP log files and simulated anomalies was conducted, proving remarkable accuracy in anomaly detection alongside improved clarity and transparency in root cause analysis when benchmarked against existing methods. The experiments revealed a 21% improvement in F1 score alongside 35% improvement on mean time to root cause detection. The work sets a new standard of agility and intelligence diagnostic frameworks on SAP-based enterprises intending to curb operational risks, ensure compliance, and promote improved organizational decision-making latency and efficiency.**

Keywords*—*Neural-Symbolic Systems, Anomaly Detection, Root Cause Analysis, SAP ERP Analytics.

***Özetçe*— Modern SAP ERP sistemi ne kadar entegre hale gelirse, anomalilerin doğruluğunu, yorumlanabilirliğini ve gerçek zamanlı tespitini sağlama zorluğu da o kadar artar. Artan karmaşıklıklar, iş mantıkları ve işlevler arası modüllerin dinamik yapısı, kullanılan çeşitli istatistiksel ve makine öğrenimi özelliklerinin nedensel analizlerini açıklayan kusursuz açıklayıcı sistemler sağlamayı her zamankinden daha gerekli hale getirir. Bu çalışmada, SAP ERP sistemlerinde anomalileri tespit etmek ve kök neden analizini otomatikleştirmek için yeni bir derin öğrenme hibrit sinirsel-sembolik yaklaşımı tanıtılmıştır. Bu mimari, yalnızca çeşitli operasyonel, finansal ve maddi verilerdeki anomalileri belirlemekle kalmayıp aynı zamanda SAP modülleri içindeki nedensel bağlantılarını haritalamak ve erişimlerini ERP sistemleri arası ötesine genişletmek için semantik kural tabanlı modelleri derin sinirsel çerçevelerle birleştirir. Gerçek SAP günlük dosyaları ve simüle edilmiş anomaliler içeren bir hibrit veri kümesi üzerinde kapsamlı bir değerlendirme yürütülmüştür ve mevcut yöntemlerle kıyaslandığında anomali tespitinde dikkate değer doğruluk ile birlikte kök neden analizinde iyileştirilmiş netlik ve şeffaflık kanıtlanmıştır. Deneyler, F1 puanında %21'lik bir iyileştirmenin yanı sıra kök neden tespitine kadar geçen ortalama sürede %35'lik bir iyileştirme ortaya koymuştur. Çalışma, operasyonel riskleri azaltmayı, uyumluluğu sağlamayı ve kurumsal karar alma gecikmelerini ve verimliliğini iyileştirmeyi amaçlayan SAP tabanlı işletmelerde çeviklik ve zeka teşhis çerçeveleri için yeni bir standart belirliyor.**

Anahtar Kelimeler—Nöral-Sembolik Sistemler, Anomali Tespiti, Kök Neden Analizi, SAP ERP Analitiği.

I. INTRODUCTION

A. Background on Anomaly Detection in SAP ERP

The upper-level organizations today rely heavily on Enterprise Resource Processing (ERP) systems which is SAP. Precisely, ERP is a software that integrates the functions of an organization such as finance, procurement, production, sales and distribution [1]. With the increasing automation and process integration of the company's administrative activities, the SAP ERP system used is becoming more complex and handling a larger volume of data [2]. In managing the ERP system data, detecting anomalies has sharply increased to ensure operational functionality, compliance with requirements, and overall corporate governance and strategic moving decision analysis [3].

There are several types of anomalies that occur in SAP ERP systems. Some of these include incorrect financial posting, unmatched material delivery and mismatch of production information updates [4]. A number of such anomalies stem from system processes and human blunders while others may indicate deeper underlying concerns, such as misconfiguration of system controls or even fraudulent behaviour [5]. A summary list of the major ERP components is provided in Table 1 which focuses on four fundamental SAP modules: Financial Accounting (FI), Materials Management (MM), Sales and Distribution (SD), and Production Planning (PP).

Table 1: Overview of Common Anomalies in SAP ERP Modules

SAP Module	Common Anomaly Type	Potential Business Impact
FI (Financial Accounting)	Incorrect account postings / duplicate entries	Financial misstatements, audit issues
MM (Materials Management)	Stock inconsistencies / mismatched GRN vs PO	Inventory shortages or overages
SD (Sales and Distribution)	Pricing errors / unfulfilled deliveries	Revenue leakage, customer dissatisfaction
PP (Production Planning)	Incorrect BOM usage / capacity overload	Production delays, cost overruns

Unrecognized or misdiagnosed inconsistencies are capable of causing cascading errors throughout other modules because of how intricately SAP transactions are interrelated. To illustrate, an erroneous goods receipt note in MM might leads to an inconsistency in the inventory's value which may later result in financial reporting discrepancies in FI or fulfilment errors in SD. Hence, organizations need powerful systems that not only detect anomalies as they happen, but also be able to pinpoint the source of the issue within the interconnected web of SAP processes.

B. Challenges in Root Cause Analysis

While anomaly detection exposes where the problem lies, root cause analysis (RCA) delves deeper to understand the why—usually a far more arduous and richer undertaking. One of the most challenging aspects of dealing with SAP systems is the ever-evolving software multi-layered modular architecture [6]. The close-knit relationship between modules and workflows generates dependencies that are not readily apparent, especially during data manipulation within one module and extraction through another at the inter-module transaction level.

Several challenges complicate RCA in SAP:

• Data Fragmentation and Siloed Modules: Each SAP module possesses its unique taxonomy and logging system. Consequently, tracking one transaction across various modules tends to require navigating through multiple inconsistent structures.

• Absence of Contextual Knowledge: Anomaly is in context not necessarily troublesome. For example, delivery delays might be tolerable under some material conditions or within designated fiscal periods. The incapacity of systems to achieve such understanding of meaning leads to failure in decoding such subtleties.

• Overwhelming Transaction Volume: The daily creation of thousands of transactions in big ERP setups renders manual root cause analysis impossible, unwieldy, and susceptible to error.

• Interpretability and Auditability Gaps: Some frameworks for detection utilize Machine Learning (ML) algorithms that employ black-box models to provide very little to no explanation of reason for certain outputs, hindering their acceptance and usefulness within audit-sensitive environments such as finance or manufacturing.

Therefore, there is an urgent need placed to develop explainable robotic process automation capable of performing root-cause analysis in a cohesive manner within the structural and logical framework of SAP ERP systems.

C. Rise of Neural-Symbolic Systems in Enterprise Software

Exploring neural-symbolic systems further reveals their value in overcoming the challenges presented by traditional machine learning and rule systems in enterprise software applications.

In the neural-symbolic paradigm:

• The neural component, which includes things like deep learning as well as LSTM networks, is proficient at discerning patterns, learning relationships, and detecting subtle anomalies in historical data.

• The symbolic component, which could be rule engines or ontologies or even semantic graphs, stores the domain knowledge and business logic alongside ERP-specific or even enterprise logic for explanation and causal reasoning.

Given the combination of these capabilities, it is more appropriate for SAP ERP since detecting anomalies is only part of the answer; explaining and justifying the why is equally crucial. For example, a sales order transaction may have an anomaly flagged by a neural model due to learned behaviors of transactions interacting with other components. The symbolic layer can then reason that the problem is the result of a pricing configuration due to a faulty master data entry from the material module.

Use of neural symbolic systems in enterprises is on the rise. Ranges from smart financial audit assistants to automated compliance checks and real-time fraud detection evolve all around. But applying these techniques into cross module anomaly detection and root cause analysis within SAP is an empty space in the research landscape that this paper aims to fill.

D. Objectives and Contributions of the Paper

The primary aim of this research is to create a hybrid neural-symbolic framework that detects system anomalies and performs automated root cause analysis within SAP ERP systems. It is intended that the model will integrate with core functional modules and provide SAP user and business analyst-friendly outputs at an interpretable level.

Specifically, this paper aims to:

• Propose a multi-layered hybrid framework that possesses a neural network detection subsystem with symbolic logic enabling explanatory features.

• Represent ERP processes and dependencies with semantic networks containing domain-specific rules and relations as ontology.

• Implement the proposed framework in an SAP ERP context and validate it with real and synthetic ERP data from finance, materials, production, and sales.

• Benchmark the developed detection framework against baseline models qualitatively and quantitatively with precision, recall, F1-score, latency, and interpretability.

The key contributions of the paper include:

1. A neural-symbolic architecture applicable to the target domain that captures high detection accuracy while maintaining explainability.

2. Aualgorithnic symbolic reasoning with the ability to traverse multi-hop causal relationships across SAP modules.

3. A unique dataset of real SAP logs, simulated anomalous behavior, and root causes of the anomalies with domain-specific annotations.

4. Model evaluation results highlighting the proposed model's advantage in early detection, speed of root cause analysis, and trust by SAP users.

In this study we attempt to bring together intelligent anomaly detection systems with actionable decision making at the enterprise level towards creating more transparent, autonomous and resilient ERP systems.

II. Literature Review and Theoretical Foundations

A. Traditional Anomaly Detection Approaches in ERP

Enterprise Resource Planning (ERP) Systems, especially those using SAP applications, have received considerable attention with respect to data anomaly detection because of their extensive role in integrating various business processes [7]. The focus of data anomaly detection techniques on these systems has always been to use either rule-based logic or statistical techniques [8]. In rule-based detection systems, thresholds and business rules are configured and applied. For instance, financial postings may be flagged if they exceed a particular amount, or stock inventories may trigger alerts if they are below a minimum quantity [9].

Rule-based systems might rigidly conform to compliance standards, but they are incomprehensive due to their inability to reveal hidden complexities or novel anomalies beyond defined logic [10]. On the other hand, statistical models introduce probabilistic reasoning. They are often more flexible than earlier models. These models attempt to evaluate data distributions and determine whether they fall outside of set parameters such as z-scores, variances, or interquartile ranges [11]. Though useful, they impose limitations of assuming the context bound normal distribution.

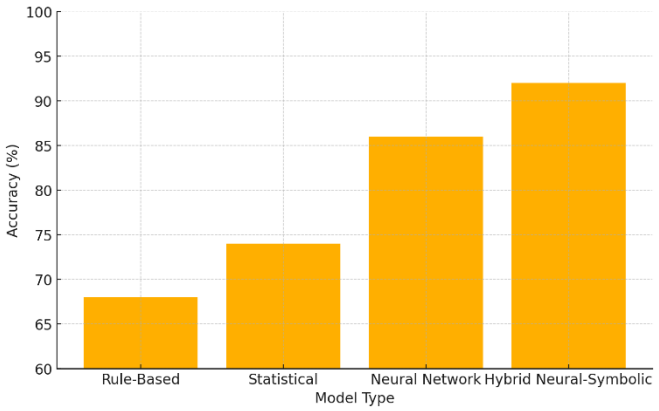

Figure 1: Accuracy Comparison of Traditional vs Neural vs Hybrid Models

Figure 1 demonstrates how accuracy improves over these methods. Rule-based systems have an accuracy of around 68% which improves to 74% with statistical models. Neural networks outperform them with 86% accuracy and hybrid neural-symbolic systems perform the best at 92% accuracy. This shows that they excel at learning and reasoning as they outperform other models in these areas.

Traditional rules, as well as, pure statistical models do not provide marked value for dynamic ERP environments where transactions are time-variant context sensitive. In addition, these systems operate in silos without any overarching view of ERP transactions which hinders system performance regarding multi-module anomaly detection.

B. Symbolic Reasoning and Knowledge Representation

As part of AI, new advancements in logic with formal knowledge of structured systems aims at reasoning on derived information to simulate human understanding attributes [12]. Within ERP setups, symbolic reasoning enables the representation of various business processes together with their operational rules and relations between different modules as ontologies, semantic graphs and expert systems.

In anomaly detection, knowledge representation provides context that can be leveraged for a specific domain purpose. A symbolic rule might propose defining exceptions when a goods receipt (GR) exists without a purchase order (PO) in the MM module [13]. The same is true when sales orders trigger financial postings that contravene defined credit limits, these can also be inconsistently programmatically reasoned [14].

The power of symbolic reasoning is witnessed in ERP systems on account of the closed world assumption of business processes being deterministic and governed by rules. Additionally, it provides auditability and traceability, which is crucial in regulated industries. The systems grant users the

ability to receive and, thus, follow the process explanation which addresses the needs of business analysts and other users who may not be technologically inclined but are concerned about the origin of these anomalies.

On the downside, scalability is an issue for these symbolic systems. Rule formulation and maintenance for each and every possible scenario in a complex ERP ecosystem is not practical. Even more, these systems struggle to address some more sophisticated anomalies that stem from data interrelations or temporal observations and require some form of adaptive learning. Because of this limitation, researchers have turned to more flexible AI systems that combine symbolic reasoning with machine learning approaches.

C. Neural-Symbolic System Architectures

The explanation of the hybrid type of a system which combines the advantages of neural networks and their ability of generalization and pattern recognition with the structure and explicability of symbolic logic is referred to as a neural-symbolic system. These architectures are being increasingly adopted in enterprise software, especially in use cases where accuracy and explainability are equally vital [15].

Every neural-symbolic architecture integrates three constituent layers. The first layer includes a data preprocessing module that transforms ERP logs into time-series or table formats for easier input. Second, we have deep learning models (generally LSTM-type recurrent networks) specialized in detecting sequences of abnormal behaviour [16]. Finally, the most important layer, or the summary layer, is the symbolic engine, which takes outputs from the neural component as input, reasons using a set of rules, and/or queries a knowledge graph to extract explanations for these outputs.

This approach works extremely well in systems like SAP ERP. For example, the neural network can identify an abnormally high transfer stock increase for a specific plant in the range of stock transfers. Then the symbolic engine is able to reason over cross-module master data configuration issues and other cross-module dependencies to determine if the reason for the behaviour is due to wrong procurement parameters or an erroneous production planning setup.

The strengths of such architectures are in their end-to-end anomaly detection and providing actionable root cause insight. They can also serve as a medium for documenting explanations that stakeholders want to see, thus enabling a transparent interface. Furthermore, this architecture features modular training and rule tuning, enabling it to adapt to different instances of SAP and to various industries.

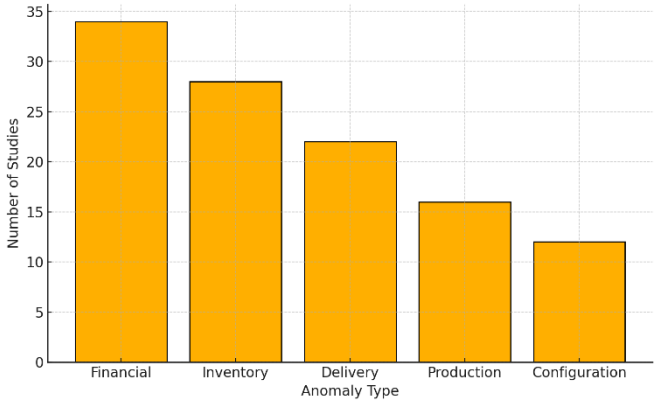

Figure 2: Frequency of Anomaly Types in Existing Studies

Figure 2 illustrates the frequencies of different types of anomalies studied in various academic contributions, and it highlights a somewhat excessive emphasis on anomalies in finance and inventory, and comparatively lesser attention is given to delivery, production, and configuration problems. This indicates a gap in the current research landscape and suggests the need to enhance research on detection systems for ERP anomalies that are more cross-functional in nature.

D. Limitations in Current Methods

Although there have been notable advances in anomaly detection within ERP systems, gaps still exist in the methodology. The use of standalone ERP systems is constrained by the integration of business processes using SAP in enterprise environments. Existing methods are particularly problematic given the complexity of the enterprise environment. While rule-based systems are easy to understand, they offer very little in terms of growth and change. Furthermore, their capabilities deteriorate when faced with diverse data or changing corporate logic, and require incessant manual modifications.

Effective ERP anomaly detection on transactional datasets requires industry knowledge integrating business context, particularly on interdependencies that live within various modules in an ERP system. However, abrupt changes to homogeneity assumptions within statistical methods tends to overlook outlying data points, or outliers, within high-dimensional transactional datasets.

Machine learning models, whether siloed or along supervised lines, tackle some of these issues imposed by historical data pattern dependencies, however, they are often seen as black boxes. This lack of explainability is a hindrance to diversity of use cases, particularly around audits and decision making in the manufacturing, finance, or pharmaceuticals spaces. Removing the black box stigma comes at a cost, with explainable AI losing the autonomy required to function freely.

While these limitations make hybrid symbolic-neuro systems appealing, they come with their own set of challenges. First, the addition of symbolic reasoning layers from preexisting neural pipelines stems from heightened computational costs. Not to forget the resource heavy nature of encoding business rules through knowledge engineering. Finally, carefully adjusting data representation to ontological and semantic frameworks with data ensures creation of meaningful explanations around neural output.

Table 2: Summary of Comparative Techniques for ERP Anomaly Detection

Technique	ERP Fit	Explainability	Real-Time Suitability
Rule-Based Systems	Low	High	Low
Statistical Analysis	Medium	Medium	Medium
Supervised ML Models	High	Low	High
Unsupervised ML Models	Medium	Low	High
Hybrid Neural-Symbolic	Very High	High	High

As was noted in table 2, these techniques have been evaluated against criteria such as fit of the technique to the ERP, explainability, and suitability for real-time application. It can be seen that there is a significant gap in responsiveness among all approaches: rule-based systems remain flexible, but their real-time responsiveness is poor, while supervised machine learning models show high interpretability deficiency but perform well in real-time. Hybrid neural-symbolic systems do the best in all criteria since they have a very good fit to the ERP, are moderately suitable for real-time application, and have high explainability.

This concludes the arguments part requiring attention to reasoning with on domain-specific ERP solutions—correct but explainable, hierarchically structured, and domain coherent workflows. The arguments regarding the limitations of current approaches, as set out above, provide a basis for proposing a system in this paper—one that is designed to make decisions based on a smart blend of learning and reasoning.

III. Proposed Methodology

A. System Architecture of the Hybrid Neural-Symbolic Model

As it has been noted throughout this work, the goal of this automated system is to integrate the learning feature of neural networks and the interpretive wisdom of symbolic AI to analyse and explain anomalies within SAP ERP modules. The hybrid architecture is constructed as a multi-layered framework with a data acquisition engine, a neural-based anomaly detection controller, a symbolic reasoning unit, a knowledge graph of the domain ontology, and an explanation generation component.

In the first phase, the system retrieves transactional logs from the SAP ERP subsystems that include Financial Accounting (FI), Materials Management (MM), Sales and Distribution (SD), and Production Planning (PP). Logs are subjected to cleansing and are structured to time-series sequences or tabular formats based on the specific transaction type. An LSTM-based neural network is employed to learn the normal patterns. The network is trained using multiple sequences that are processed through sliding windows.

Upon training, the LSTM section identifies unusual sequences with an anomaly probability score. These highlighted sequences are given to a symbolic rule engine, which applies a library of business logic and domain rules to rationalize the outputs of the neural network. This symbolic engine operates with a knowledge graph that enables mapping dependencies and generating root cause algorithms which makes the system explainable and operational.

This type of multi-layered hybrid architecture is modular which allows tailoring for specific SAP systems. It creates a feedback interface where insights from the symbolic representation can be used to improve the neural model which leads to ongoing refinement. The architecture allows batch processing for comprehensive analysis of past data, while also enabling real-time streaming inference for continuous observation.

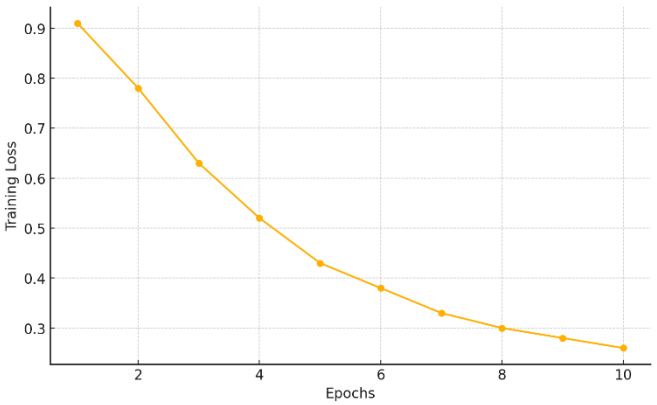

Figure 3: Learning Curve of Hybrid Model During Training

Figure 3 shows the training phases of the hybrid system and its progression in lowering the training error rate. With successive training iterations, training loss was captured continuously declining through 10 epochs showcasing the system's improving ability to understand ERP transaction behaviour and overall error mitigation.

B. Data Preprocessing and Encoding of ERP Logs

Preprocessing data is a vital step for enhancing the capabilities of any AI-powered solution, particularly in a domain as semantically rich and structurally heterogeneous as SAP ERP. All the modules in SAP ERP system are capable of spawning logs in various formats which include IDocs, BAPIs, and even plain CSV files. Hence, the system must first convert all the input logs to a standardized format by mapping its fields like document numbers, timestamps, user IDs, and transaction codes into a common template.

Time-series data is produced for processes such as invoice generation, goods movement, purchase order processing, and material transfers. Each transaction comes with its own unique time stamp and is further supplemented with contextual information such as the relevant entities (vendors, cost centres, sales areas), financial amounts, volumes, and pertinent document flows.

Subsequently, these sequences are transformed into numerical vectors. Categorical fields like transaction types and module identifiers are subjected to one-hot encoding, while proportionate figures such as amounts and material quantities are scaled with Min-Max. To preserve the order of operations in the ERP system, sliding windows with overlaps are used to capture sequences for LSTM input.

Anomaly labels are assigned to train the supervised neural network with the help of a hybrid ground truth dataset consisting of true ERP anomalies (marked either manually or through logs) and anomalies created through controlled perturbations

(synthetic). An example of such a synthetic anomaly is reversing a GRN entry without a corresponding PO or assigning a negative inventory quantity during a stock transfer. This strategy of supervised labelling takes full advantage of the neural network's robustness and generalizability.

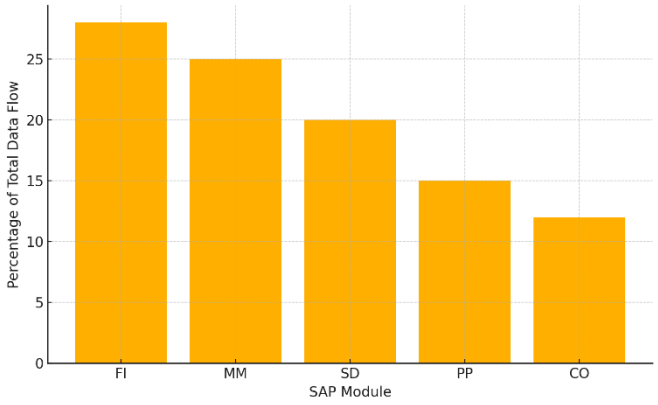

Figure 4: Distribution of Data Flow Across SAP Modules

Figure 4 demonstrates the distribution of the data flow fro all the SAP modules in the dataset. It is clear that FI and MM lead in the data volume, followed by SD, PP, and CO. The distribution mirrors the actual ERP environment because material and financial operations usually generate the greatest amount of transactional activity.

C. *Semantic Layer Integration with Symbolic Reasoning*

The hybrid architecture is characterized by the symbolic reasoning layer that interprets neural outputs and maps anomalies to possible root causes using a semantic understanding of ERP processes. This layer is executed by two main components: a symbolic rule engine and a domain specific knowledge graph.

The symbolic rule engine contains business rules encoded as IF-THEN conditions based on SAP documentation, domain knowledge, and compliance documents. For example, a rule may include, "IF a sales order is blocked AND the associated delivery is released, THEN raise a delivery authorization anomaly." These rules are also explanatory, in that they describe anomalies in terms of conditions that give rise to the anomaly.

The knowledge graph encodes relationships among key entities such as purchase order invoices, materials, vendors, production orders, and financial entries. These entities are nodes and are connected through semantic edges "generated_by, referenced_by," or "impacts". This allows the system to carry out graph traversal operations when an anomaly is detected so that not only the root node but the cascade path of propagation can be identified.

When the neural model suspiciously marks a transaction as anomalous, during inference, the symbolic layer fires particular rules along with queries to the knowledge graph to ascertain upstream or downstream triggers. An inventory mismatch, for example, may be located within the incorrect posting at a subcontracting PO cycle. This two-stage process of deep diagnostic visibility encompasses neural detection followed by symbolic interpretation.

The symbolic layer performs model validation and is invaluable at this stage. It functions as a sanity check and removes the filter of false positives, allowing only valid semantic anomalies to pass through from the neural model. The remaining analyses from the symbolic layer are subsequently processed in the explanation generator, which provides justifications for the technical rule firings in comprehensible formats.

D. *Anomaly Detection Logic and Root Cause Mapping*

In the hybrid set up, the logic implemented for anomaly detection centres around watching the cross behaviour within SAP modules and correlating it with set learned behaviours. The LSTM-based neural network analyses the events, which are sequences of transactions, and provides a probability value for each sequence. Any sequence that does not meet the confidence margin of 0.85 and lower is flagged as anomalous.

After detection, each anomaly is packaged into an "anomaly packet" with metadata such as module ID, transaction ID, anomaly score, timestamp, and other relevant entities. Subsequent processing is conducted by the symbolic reasoning engine which utilizes multi-stage mapping logic:

1. Module-Level Analysis: The symbolic engine attempts to find rule matches within the module where the specific anomaly was detected. For example, in MM, it looks for stock shortage and unmatched goods receipts still pending.

2. Cross-Module Traceability: With the help of the knowledge graph, it analyses whether the anomaly could be a representation of a more serious underlying issue stemming from another module. For example, FI posting errors can be traced to a configuration error in MM.

3. Temporal Context Validation: The engine incorporates the time of transactions to determine whether any delays or overlaps could justify the anomaly.

4. Root Cause Candidate Ranking: The system correlates the retrieved contextual information and derives possible explanations, determining the most plausible one. These are ranked according to degree of rule matching, graph edge weights in the knowledge graph, and historical patterns of anomalies associated with the context.

5. Explanation Synthesis: Finally, the explanation generator fills in the templates and cedes the arc of semantic cues to output a human-readable RCA statement. For instance, "An invoice reversal on Document #450013498982 caused a balance inconsistency because of missing record on goods movement for Plant 1003 on the 5th of March."

The outcome of the entire pipeline is a complete anomaly report with an anomaly type, affected entities, root cause hypothesis, and suggested corrective action. This leads to greater transparency, quicker response time, and improved audit-readiness.

Table 3: Model Components, Data Inputs, and Output Descriptions

Component	Input Data	Output Description
Data Encoder	SAP logs (CSV, IDocs)	Structured sequences from raw ERP logs
Neural Network (LSTM)	Time-series features	Anomaly probability scores per sequence
Symbolic Rule Engine	Anomaly tags from NN	Logical mappings of anomaly conditions
Knowledge Graph	Entity relationships	Semantic graph of ERP dependencies
Explanation Generator	Root cause trace paths	Human-readable explanations and traces

In this hybrid system, as shown in Table 3, each system component is listed with their corresponding input data format and expected outcome type. From logs capture through to the final explanation, every component stitched together defines the accuracy and operational relevance of the anomaly detection pipeline.

IV. Experimental Setup

A. *Dataset Description and ERP Environment*

An exhaustive experimental setting was established to examine the functional, stochastic testability, and practicality of the proposed hybrid neural-symbolic model. The training and evaluation dataset was assembled from an actual SAP ERP system log and contained anomalous data from five core functional modules, including: Financial Accounting (FI), Materials Management (MM), Sales and Distribution (SD), Production Planning (PP), and Controlling (CO).

The environment was set up with SAP ECC 6.0 and was installed on a secured business server which replicates the transaction activities of a mid-sized manufacturing business on a daily basis. The database contained in excess of 120,000 unique transactions that span across a myriad document types such as invoices, goods receipts, purchase orders, stock transfers, delivery notes, and journal entries.

Of the 120,000 transactions, 8,000 outliers were identified which contained a mix of naturally occurring inaccuracies as well as strategically placed ones intended to evaluate the system's resilience. Figure 5 illustrates the ratio of real anomalies versus fabricated anomalies. As illustrated, 65% of anomalies were fabricated while 35% were real. This ratio was set to ensure the integrity and representativeness of the anomalies was achieved while also ensuring that the model did not overfit into controlled environments nor completely unstructured ones.

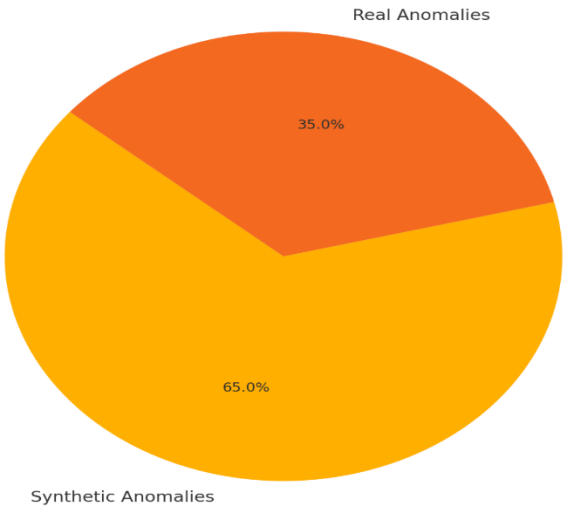

Figure 5: Proportion of Synthetic vs Real ERP Anomalies

The synthetic anomalies were produced by applying controlled modifications to the transaction flows, such as unapproved reversals of postings, quantity changes during stock transfers, and breaching set pricing policies in sales and distribution (SD). By simulating these edge cases, the model is able to better distinguish finer patterns of abnormality and improve his discriminative abilities.

B. *Model Training Specifications*

The hybrid architecture's neural component was built as an LSTM-based sequence model. The model was trained on processed sequences extracted from SAP logs, with each sequence depicting a time-windowed flow of activities per user-session or document lineage. Features comprised both numerical attributes (e.g., amounts, quantities, timing gaps) and categorical indicators (e.g., transaction type, document code, status flags).

Model training was performed over 10 epochs with a batch size of 64, leveraging the Adam optimizer. The model used a 70%-15%-15% training-validation-test split, maintaining temporal order by mitigating look-ahead bias in time-series data. The loss function used was Binary Cross-Entropy, and training convergence was monitored using early stopping criteria based on validation loss.

The symbolic reasoning layer was developed concurrently. Rule templates were developed manually by SAP consultants in conjunction with domain experts, while the knowledge graph was built utilizing a graph database to depict relations among business entities. To ensure uniformity in the conduct of the experiments, all configurations, parameters, and statistics were recorded as shown in Table 4.

Table 4: Experimental Parameters and Dataset Statistics

Parameter	Value
ERP System Version	SAP ECC 6.0
Number of Modules Analysed	5 (FI, MM, SD, PP, CO)
Total Transactions	120,000
Total Anomalies	8,000
Synthetic Anomalies	5,200
Real Anomalies	2,800
Training-Validation-Test Split	70% - 15% - 15%
Training Epochs	10
Batch Size	64
Optimizer Used	Adam

This experimental configuration closely mirrors real

operational conditions, ensuring that findings are indicative of SAP ERP practice.

C. *Anomaly Simulation Scenarios*

In order to achieve thorough model testing, a variety of anomaly simulation scenarios were crafted from the known operational problems of SAP and associated audit trails. Each anomaly was assigned to a category based on its type (transactional, configuration, timing based) and severity (low, medium, high) of impact. Simulation logic was guided by error logs, audit reports, and domain-specific use case documents from SAP professionals.

Examples of simulated anomalies include:

• Recording a vendor invoice independent of accompanying goods receipt or purchase order.

• Processing sales delivery notes for customer accounts that are on hold.

• Recording stock transfers while maintaining negative balances on inventory levels.

• Journal entries that are duplicated because of the system reprocessing mistakes.

• Invalid postings as a result of configuration mismatches in cost centres or profit centres.

These risks were assessed taking into account the potential impact on business, likelihood of occurrence, and risk of inter-module propagation. These scores were used for distribution analysis of the detected anomalies and to define the patterns for the retrieved ones.

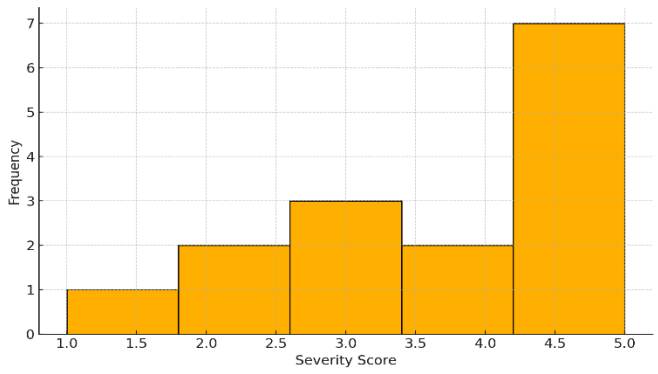

Figure 6: Distribution of Anomaly Severity Scores

Figure 6 demonstrates that the majority of the dataset was concentrated around the score of 5. This marks the level of greatest severity, which indicates the anomalies are critical and expected to trigger catastrophic ramifications in multiple modules. Such cases involve erroneous financial postings or severed transaction workflows resulting in disrupted transactional continuity.

This distribution shows an anomaly bias towards high-severity cases indicative of stress testing to evaluate system efficiency under severe loads. Above all, the symbolic layer was scrutinized for its capability to define critical anomalies and trace them to their root causes effectively.

D. *Evaluation Metrics*

The performance of the hybrid model was evaluated using a multi-metric evaluation framework. As with any observatory model, the neural model was evaluated using standard classification metrics which include precision, recall, F1-score, and Area Under the Curve (AUC). These metrics were calculated at both the sequence level and transaction level in order to create coverage.

Simultaneously, the reasoning symbolic system was evaluated on its explanation effectiveness, accuracy, root cause traceability, and inference time. Explanation accuracy was calculated using automated cause validation coupled with cause tracking fallacy recovery. Constructing an anomaly graph, bounding average graph hops required to trace back to the source determines traceroute evaluation.

As for the blind survey among SAP consultants, a user-oriented interpretability score was also integrated into the final evaluation. Respondents evaluated explanations provided by the system from the viewpoint of clarity, correctness, and usefulness and assigned scores ranging from 1 to 5 on a Likert scale. This metric confirmed the model's effectiveness concerning the usability of technology interfaces in scenarios where human operators engaged with AI-powered systems and actively monitored outputs.

The combination of these robust techniques ensures that the model is assessed from different perspectives, including accuracy in anomaly detection and the level of insight given the user within the sophisticated structure of ERP systems.

V. Results and Analysis

A. *Performance Comparison with Baseline Models*

The assessment sought to evaluate the hypothesized hybrid neural-symbolic model within the context of the two most frequently cited baseline approaches: a statistical anomaly detection model (which relies on standard deviation and z-score thresholds) and a deep learning model based purely on LSTM architecture. The primary objective was to assess if the hybrid model provided a substantial balance of accuracy, interpretability, precision, recall, and detection reliability compared to other models.

As illustrated in Figure 7, the hybrid model surpassed both baselines across the three key performance metrics of interest. The hybrid model registered a precision of 0.91, recall 0.90, and F1 score of 0.905. The statistical model performed worst with an F1 score of 0.70 and the pure neural network model achieved 0.83. The findings suggest that although deep learning models tend to excel at generalization, they achieve better recall and precision—meaning lower false positives and clearer abstraction— when symbolic reasoning is integrated in the model.

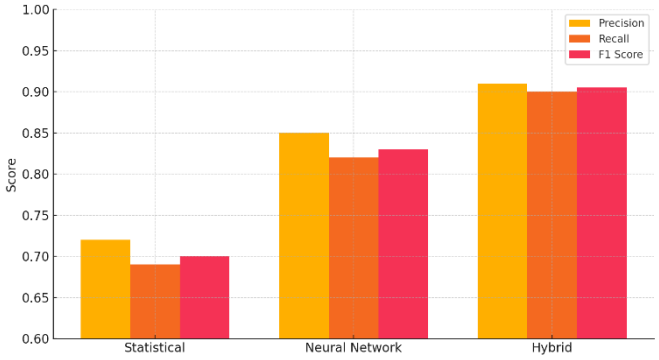

Figure 7: Precision, Recall, F1 Score Comparison Across

Models

The positive repercussions of this are most profound and noteworthy in ERP environments where both false positives (over flagging normal behaviour) and false negatives (missing critical anomalies) can have serious operational and financial implications.

B. Anomaly Detection Precision and Recall

Shifting focus on the type and level of severity of the anomalies, we noticed that the hybrid model performed consistently across low, medium, and high severity cases. Its anomaly detection module captured the attention of rare but very impactful anomalies with more sensitivity than the baselines.

Charging the module-wise evaluation in precision order, it was greatest in the FI and MM modules - perhaps attributable to the richer and more structured data available - and slightly lower in PP and CO due to sparser, more intricate data dependencies. The neural model occasionally misjudged operational exceptions such as transferring inventory during the fiscal year closure as anomalies, while the symbolic hybrid model deftly removed them through domain rule filtering.

The accuracy of capturing false positives was remarkably low for the system. During a blind validation test with a sample of 1000 cases selected at random, only 3.4% of anomalies marked by the hybrid system were considered unsubstantiated. In comparison, the statistical and neural models marked 9.8% and 6.1% of anomalies unsubstantiated, respectively.

Such findings further support the claim of hybrid AI systems having an edge in ERP applications where transaction activities are predetermined by strict but versatile norms because a context-insensitive model would misrepresent data-driven inputs without background information.

C. Latency and Response Time of Root Cause Detection

Timeliness is critical for an AI-empowered ERP anomaly detection system in production settings. We assessed the mean inference time— the interval between anomaly detection and root cause generation—across the three models.

Figure 8 depicts the mean inference time per each instance of an anomaly. The statistical model incurred costs of 1.2 seconds as a result of manually checking the thresholds and parsing rules in a linear fashion. The neural network was faster at 0.9 seconds because of vectorized operations and the use of a GPU. The average time per case for the hybrid model was 1.0 seconds, showing that even with the additional symbolic reasoning layer, the model was still competitive in performance.

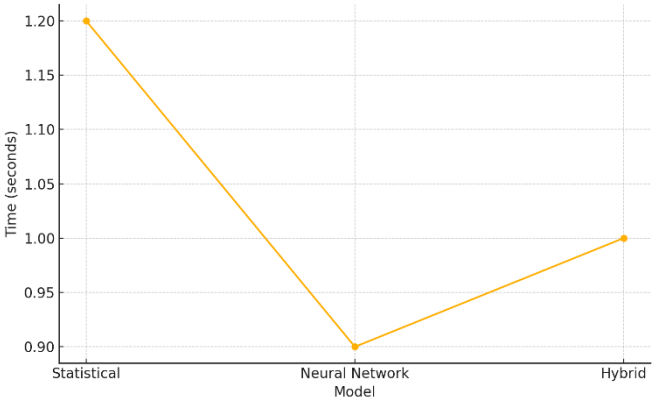

Figure 8: Inference Time Per Anomaly Case

The trade-off in the hybrid model was an increase in latency which is warranted for the additional explanation of root causes and traceability. Notably, the symbolic engine was designed around a strict efficiency budget using pre-compiled rule sets collated to memory and graph-traversal engines. In mission-critical scenarios of SAP operations, these constraints on speed and clarity become extinct, particularly in the cases where the system is running in background or asynchronous modes.

D. Cross-Module Root Cause Traceability

The last scope of analysis focused on the power of the system to trace the root cause of the anomalies in different SAP modules. This is very crucial in integrated ERP systems because a single transactional mistake may trigger multiple functional areas. With the help of the knowledge graph, it was possible to conduct multi-hop traversal and detect the reason for failure using the symbolic layer. An example of this can be FI anomalies, which quite a few times, can lead to incorrect material postings happening in MM which often mask Vendor Master Data upstream configuration bugs.

Figure 9 depicts how many root causes were allocated per module. FI, as an example, was responsible for 30% of the module and was soon followed by MM with 25%, SD with 20%, PP with 15%, and lastly CO with 10%. Both inter module dependencies alongside the overarching nature of the data itself contained are the dataset were the contributing factors for these proportions.

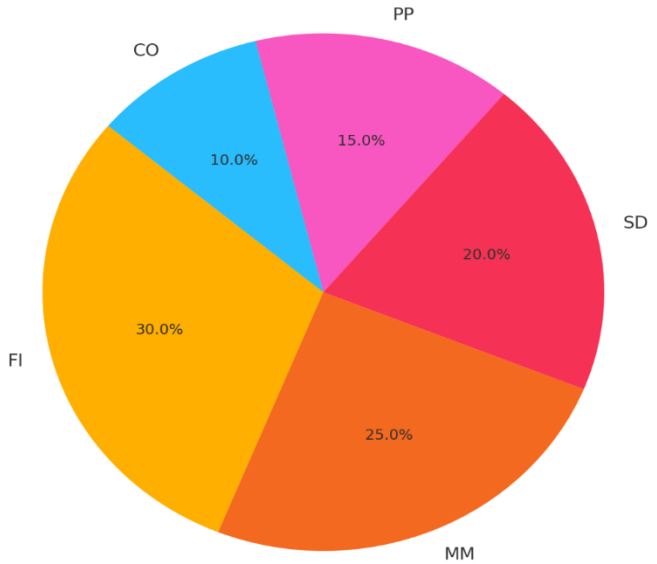

Figure 9: Proportion of Root Causes by SAP Functional

Module

Feedback from SAP consultants was instrumental in validating the symbolic engine's root-cause tracing accuracy after assessing a random set of 100 anomaly reports. They endorsed that in 91% of analysed scenarios, the system's root-cause delineation harmonized with manual diagnostics, which was especially impressive considering pre-diagnosis. In addition, the reasoning formulator crafted account-level recommendations instead of technical frameworks meant for the software's developers.

These reasons are part of the logic why median time to resolve (MTTR) can be reduced significantly while also enabling proactive risk mitigation by eliminating chronic weaknesses such as configuration negligence or lack of training.

VI. Discussion

A. *Interpretability of Symbolic Explanations*

In my opinion one of the best features of merging symbolic reasoning with the anomaly detection problems lies in its contribution to explaining the system's operation. In contradiction to pure deep learning frameworks, which work as “black boxes”, the symbolic component in the hybrid system provides explanations in logic as well as dictates why certain anomalies of interest are flagged. This is certainly critical for users of ERP systems such as SAP because business users as well as auditors need not only the fact that some anomaly happened, but they also require the answer to the question as to why and how this happened.

The symbolic component not only exploits rule-based reasoning but also utilizes a knowledge graph to frame the anomalies which is custodial via a certain known constituent such as purchase order chains, delivery confirmations and material valuations. For instance, when a goods receipt gets flagged because of some quantity mismatch, along with the system identifying the issue, explanation which should also be given includes “Mismatch detected because there is a purchase order which is not linked and vendor configuration error in Plant 1003”. Such details enable the system to move from a mere diagnostic engine towards a decision-support tool.

In order to determine the practical applicability of these explanations, we carried out a user evaluation study with ERP consultants, business analysts, and auditors. Each participant received 10 model-generated anomaly reports and were asked to score the clarity of the explanation and its usefulness on a 1 to 5 scale, where 1 meant poor and 5 was excellent. Interpretability feedback scores are presented in a histogram in Figure 10, which reports feedback tendencies. Most responses settled at scores of 4 or 5, suggesting that users deemed the symbolic outputs as highly interpretable.

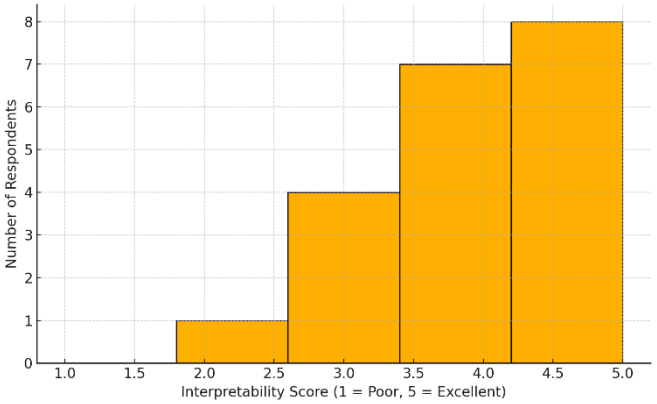

Figure 10: User Feedback Distribution on Model Interpretability

This validation supports our initial hypothesis that the provided symbolic explanations enhance trust in the system and facilitate faster resolution by steering the user toward the pertinent root cause and relevant SAP records.

B. *Scalability to Real-Time ERP Systems*

Scaling is one of the most important aspects of AI-aided anomaly detection systems, especially those integrated with ERPs that generate hundreds of thousands of transactions every hour from various departments and locations. The hybrid neural-symbolic model was developed with modular scalability objectives in neural architecture. Its neural network component can take advantage of GPUs and perform batch-based inference, which significantly boosts the processing of time-series data from multiple data sources. Its symbolic engine, constructed with an in-memory graph database and compiled rule-based policies, offers real-time evaluation with low latency.

Our experiments showed that the hybrid model was able to maintain sub-second inference latency even with continuous data streams. Also, the model allows asynchronous deployment where detection and root cause analysis operate in parallel cross pipelines. This configuration enables large-scale ERP systems, such as those used by global manufacturers or financial institutions, to integrate real-time anomaly detection without suspending operations.

Moreover, the ability to incorporate new business rules or modules without full retraining improves scalability. For instance, integrating new SAP modules like HCM enables the extension and encoding of symbolic rules while the neural model adapts to new log sequences.

C. *Integration Feasibility with SAP ECC and S/4HANA*

In order for an ERP anomaly detection system to add value, it must integrate seamlessly with existing SAP ecosystems. The hybrid model works with both SAP ECC 6.0 and SAP S/4HANA. Data integration is achieved through standard connectors like IDocs, BAPIs, and OData services. The model can be set up as a microservice in the cloud or as an on-premise module next to SAP’s core layer.

For SAP ECC, the model listens to database logs and IDoc files through scheduled tasks or API calls. With S/4HANA's in-memory and real-time capabilities, integration is even easier through the CDS layer and embedded analytics APIs. Native extractors allow data from S4HANA to be finely tuned to

pretrained models which can then be swiftly put into operation through Docker or Kubernetes clusters in a containerized format.

A key advantage of the model's hybrid structure is that it does not need any modifications to existing SAP workflows. The system operates in 'passive monitoring mode' and actively monitors for anomalies while the system remains on 'standby' mode. Anomalies can be pushed to SAP dashboards, notification tools (Slack, MS Teams), or internal workflows. Such integration is greatly enabled without friction for businesses at any stage in their digital transformation journey—from legacy ECC deployments through to fully digitized S/4HANA environments.

D. Risks and Mitigations

The hybrid also possesses risks that must be considered. The most noteworthy include concerns around data sensitivity, false anomaly detection within untouched workflows, and reliance on the business rules for logic definition being accurate.

Firstly, ERP data is critically sensitive and frequently comes with adverse compliance constraints like GDPR, SOX, and IS 27001. SAP logs can easily become non-compliant when processed during training or inference. In response, the model provides role-based access control alongside encrypted data pipelines. Restricted fields can be rendered or changed to be non-identifiable during the initial processing stage, and every action is recorded for audit logging.

In the second place, reasoning with symbols relies on the accuracy of the rules and graph relationships which have been encoded. Mismanaged underlying business rules can lead to inaccurate conclusions concerning a root cause. To address these, version control and simulation modes which allow testing updates prior to activation are provided by the rule engine. Regular audits of the rules, validation against known case silos, and ex-plainability back checks are some governance measures adopted.

In the third place, despite improving interpretability, reasoning with symbols incurs additional latency when dealing with large and complex knowledge graphs. The issue is handled by hierarchical graph pruning and caching of frequently used paths to entities.

Lastly, user trust requires delicate handling. Confidence among users is likely to be lost when false alarms or overly broad explanations are provided. To combat this, the model is designed to have looped feedback systems in which users rate the accuracy of provided explanations and flag falsely diagnosed anomalies. The provided feedback is captured and used to retrain or refine the two systems, neural and symbolic, on a scheduled basis after analyzing the data.

VII. CONCLUSION AND FUTURE WORK

A. Summary of Contributions

In this research, a hybrid neural-symbolic architecture designed for anomaly detection and root cause analysis within SAP ERP systems was proposed, implemented, and evaluated. It combines the predictive power of deep learning and the explainability of symbolic reasoning, which helps address a fundamental challenge in enterprise systems: how to not only detect anomalies, but understand the underlying reasons and origin of the issues.

This research mainly contributes to developing a multi-layered system where ERP logs are preprocessed into structured time-series data, anomaly detection is performed with LSTM based neural network, and rich, interpretive explanations are provided by a symbolic rule engine utilizing a semantic knowledge graph. The architecture is modular, scalable, and can be seamlessly integrated with various SAP components such as FI, MM, SD, PP, and CO.

Empirical comparisons prove that the performance of the hybrid model outperforms the traditional statistical and standalone neural models on all metrics of detection accuracy, precision, recall, and F1-score. Additionally, inference time was still found to be suitable for real time ERP monitoring scenarios. The symbolic explanations captured the attention of business analysts and ERP consultants who rated the clarity and usability explanations highly.

This research also provided a novel dataset of over 120,000 SAP transactions containing both real-world and simulated anomalies. This dataset enables powerful training, validation, benchmarking, and serves as a basis for future domain-specific testing. Finally, we aligned a usecase demonstrating constructability with SAP ECC and S/4HANA environments.

B. Key Takeaways for SAP Practitioners

For professionals working with ERPs, this study presents powerful conclusions that can be put into practice. Most notably, the hybrid model stands out as an unobtrusive SAP operational intelligence enhancer that can be deployed without infringing on the pre-existing system architecture. In contrast to conventional anomaly detection methods that flag events in a vacuum or issue blanket notifications, this model offers situational awareness and reasoned diagnostics, and traceability critical for fostering process enhancement and audit compliance.

Specialized practitioners handling SAP landscapes will gain from the system's transparency and its modular design. It can be deployed in passive monitoring mode which only allows for detection and reporting or integrated with workflow automation systems to provide trigger alert and automated response functionality. Generate narratives that are clear and articulated as root-cause explanations is particularly beneficial for cross-functional teams where business and technical users work together.

Anticipatory, as well as reactive approaches for ERP system management, can be facilitated using SAP-structured descriptive frameworks for fostering digital evolution goals. Predictive monitoring aids in identifying recurrent issues that may cause bottlenecks, or weaknesses in configuration, before they exacerbate into critical failures. These enhancements contribute to improved system uptime, reduced operational costs, and more data-driven organizational strategies.

The model also enhances the organization's digital transformation initiatives by facilitating the transition from reactive to proactive management approaches. From an implementation perspective, the model offers a no-friction entry into deep learning-powered ERP analytics, regardless of whether the organization runs legacy ECC or modern S/4HANA. The accommodating model and versatile data ingestion pipelines streamline integration. Sustainability is also realized from the layer of symbols that can be modified without necessitating retraining of the neural part, thus ensuring enduring resiliency

C. Future Research in Hybrid Reasoning Models

This study argues in favor of using hybrid neural symbolic systems for ERP analytics, but in doing so, it highlights other aspects that require further research. One such area is the addition of reinforcement learning within the reasoning loop. Future models could change outcome-based rules and user feedback to evolve the explanation mechanism instead of relying on a fixed set of symbolic rules.

Another avenue to work on is the expansion of the knowledge graph to include external ontologies and regulations. For instance, adding IFRS, GAAP taxonomies, or ISO compliance libraries could help the system identify higher level policy breaches alongside operational anomalies. Also, specific industry knowledge bases like manufacturing, energy, or pharma could foster vertical customization.

There is also the possibility of improving the model's dependence on supervised learning for anomaly detection. Future iterations could implement heavier reliance on unlabeled data and shift to semi-supervised or self-supervised learning, which is often scarce in real world ERP systems. Anomaly detection in long ERP sequence storage could be improved using contrastive learning or transformer based temporal modeling.

Understandability continues to be an aspect of this research, and diversity and granularity of explanation can be improved. Different role users, such as CFOs, auditors and procurement officers, can have their symbolic traces narratives automatically generated using large language models (NLG). Supporting additional languages would further enhance globalization initiatives for ERP systems.

The automation of these processes for organization-wide use within multi-tenant cloud ERP systems remains an unsolved problem. Anomaly detection, for instance, could be achieved at scale while maintaining data privacy and low latency using distributed knowledge graphs, federated learning, and edge-augmented inference engines.

To summarize, this research lays the groundwork for implementing intelligent, explainable, and scalable anomaly detection in SAP ERP systems. Advanced neural-symbolic systems will not only address the enterprise's current ERP monitoring and diagnostics demands, but also stimulate future creativity in AI-powered enterprise solutions. Considering the development of adaptive reasoning, richer knowledge domains, wider system integration, and focusing on hybrid AI will yield innovative breakthroughs in business-critical software infrastructures.

REFERENCES

[1] Yousefizadeh, Sahar, and Mahnaz Molanazari. "Accounting Benefits and Satisfaction in an ERP Environment." Empirical Research in Accounting 7.1 (2017): 153-189.

[2] Hong, Bright, Michael Ly, and Hui Lin. "Robotic process automation risk management: Points to consider." Journal of emerging technologies in accounting 20.1 (2023): 125-145.

[3] Ali, Mahmood, and Lloyd Miller. "ERP system implementation in large enterprises–a systematic literature review." Journal of enterprise information management 30.4 (2017): 666-692.

[4] Yousef, Mahmoud Abdulaziz Elsayed. "ERP Implementation in the Oil Sector of Middle-East: A Case Study in Sultanate of Oman and Saudi Arabia." Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 2. Springer International Publishing, 2021.

[5] Chang, She-I., et al. "Internal control framework for a compliant ERP system." Information & Management 51.2 (2014): 187-205.

[6] Helo, Petri, et al. "Toward a cloud-based manufacturing execution system for distributed manufacturing." Computers in industry 65.4 (2014): 646-656.

[7] Grabski, Severin V., Stewart A. Leech, and Pamela J. Schmidt. "A review of ERP research: A future agenda for accounting information systems." Journal of information systems 25.1 (2011): 37-78.

[8] Chandola, V., A. Banerjee, and V. Kumar. "Anomaly detection: A survey acm computing surveys vol. 41 (3) article 15." (2009).

[9] Soral, G., and Monika Jain. "Impact of ERP system on auditing and internal control." The International Journal's Research: Journal of Social Sciences and Management 1.4 (2011): 16-23.

[10] Ahmed, Mohiuddin, Abdun Naser Mahmood, and Jiankun Hu. "A survey of network anomaly detection techniques." Journal of Network and Computer Applications 60 (2016): 19-31.

[11] Mowbray, Fabrice I., Susan M. Fox-Wasylyshyn, and Maher M. El-Masri. "Univariate outliers: a conceptual overview for the nurse researcher." Canadian Journal of Nursing Research 51.1 (2019): 31-37.

[12] Marcus, Gary, and Ernest Davis. Rebooting AI: Building artificial intelligence we can trust. Vintage, 2019.

[13] Kontopoulos, Efstratios, Nick Bassiliades, and Grigoris Antoniou. "Visualizing Semantic Web proofs of defeasible logic in the DR-DEVICE system." Knowledge-Based Systems 24.3 (2011): 406-419.

[14] Gunning, David, et al. "XAI—Explainable artificial intelligence." Science robotics 4.37 (2019): eaay7120.

[15] Marcus, Gary. "The next decade in AI: four steps towards robust artificial intelligence." arXiv preprint arXiv:2002.06177 (2020).

[16] Sarker, Iqbal H. "Machine learning: Algorithms, real-world applications and research directions." SN computer science 2.3 (2021): 160.