Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Sharing Cyber Threat Warnings Across Banks in Real Time Without Exposing Private Data: A Privacy-Preserving Federated Threat Intelligence Framework

Saiprakash Kodela

Independent Researcher, USA

ARTICLE INFO

Received: 03 Oct 2024

Revised: 17 Nov 2024

Accepted: 28 Nov 2024

ABSTRACT

Banking databases contain sensitive customer, transaction, payment, credit, identity, and regulatory information. Conventional database access-control mechanisms can determine whether a user has permission to access a resource, but they are less effective when an authorized employee, administrator, contractor, or compromised account uses legitimate privileges for an unauthorized purpose. Machine-learning systems can identify unusual behavior, although their predictions are probabilistic and may be difficult to explain or enforce consistently. This paper proposes a neuro-symbolic artificial intelligence framework for policy-aware database access control and insider-threat detection in banking systems. The framework combines neural behavioral models with symbolic authorization rules, attribute-based policies, separation-of-duty requirements, business-purpose restrictions, data classifications, and transaction context. The neural component analyzes database queries, authentication activity, access frequency, record volume, temporal patterns, device posture, and peer-group behavior. Its outputs are transformed into risk predicates that are evaluated by a symbolic policy engine. The engine produces graduated decisions, including permit, permit with masking, result limitation, step-up authentication, human review, temporary suspension, and denial. The proposed architecture preserves deterministic security rules while allowing access decisions to respond to changing behavioral risk. It also creates an explanation trace that records the policies, attributes, risk indicators, and evidence associated with each decision. A design-science methodology, formal decision model, banking use cases, threat analysis, and evaluation framework are presented. The paper argues that neuro-symbolic integration can improve continuous authorization, contextual insider-threat detection, explainability, and regulatory auditability. However, the system should support human security decision-making rather than independently determine employee guilt or impose disciplinary action.

Keywords: neuro-symbolic artificial intelligence, banking security, database access control, insider threat, behavioral analytics, attribute-based access control, zero trust, explainable AI

INTRODUCTION

Banking operations depend on databases that process customer identities, deposits, withdrawals, loans, card transactions, payment instructions, credit assessments, fraud indicators, and regulatory records. These databases are accessed by employees, database administrators, applications, service accounts, third-party providers, auditors, and automated banking services. The diversity of users, devices, applications, and business processes creates a complex authorization environment. Access that is appropriate for one employee, customer, purpose, location, device, or transaction may be inappropriate under different circumstances. Banking institutions therefore require access-control mechanisms that consider not only the identity of a user but also the context and purpose of each request.

Traditional access control primarily determines whether a subject is permitted to perform a particular action on an object. Role-based access control assigns permissions to organizational roles such as teller, credit officer, relationship manager, payment approver, fraud investigator, and database administrator (Sandhu et al., 1996). This approach simplifies permission management because users receive access through their assigned roles rather than through individually configured permissions.Attribute-based access control provides greater flexibility by

evaluating attributes associated with subjects, protected resources, requested actions, and environmental conditions (Hu et al., 2019). For example, a bank may permit a relationship manager to view the records of customers assigned to that manager when the request is made for an approved customer-service purpose, during authorized working hours, and from a managed device. Attribute-based policies can therefore provide more contextual and fine-grained decisions than role-based permissions alone.

Although role-based and attribute-based controls are necessary, they cannot always distinguish legitimate use from malicious or inappropriate use of valid permissions. An employee may have permission to access customer information but use that permission to browse high-profile customer accounts, collect information for identity theft, or export customer records before leaving the organization. Similarly, a database administrator may possess the technical ability to query sensitive tables even when the access is unrelated to an approved maintenance activity.

Insider threats are especially difficult to detect because the actor may be properly authenticated and may possess legitimate knowledge of organizational systems and procedures. An insider may be a malicious employee, a negligent employee, a contractor, a privileged administrator, or an external attacker using compromised employee credentials. Unlike many external attacks, insider activity may initially appear similar to legitimate organizational behavior. Insider-threat detection must therefore examine not only whether an action is technically permitted but also whether it is behaviorally unusual and inconsistent with the user's responsibilities.

Behavioral analytics can assist by learning normal patterns of user activity and detecting significant deviations. Previous studies have applied Gaussian mixture models, Bayesian networks, and deep-learning techniques to insider-threat detection (Al Tabash & Happa, 2018; Elmrabit et al., 2020; Nasir et al., 2021). These approaches demonstrate that user activities, access patterns, and organizational risk factors can be analyzed to identify potentially suspicious behavior.

However, unusual behavior does not automatically indicate malicious intent. An employee may access an unusually large dataset during an audit, regulatory investigation, system migration, fraud investigation, or emergency response. A purely statistical anomaly detector may therefore generate false positives when it does not understand the business purpose or policy context of an activity. Operational studies also show that insider-threat detection requires interpretable alerts, proportionate investigations, and suitable organizational governance rather than model accuracy alone (Erola et al., 2022).

The central problem is therefore not simply the detection of anomalies. A banking security system must determine whether an unusual action is consistent with the user's permissions, business purpose, customer relationship, transaction responsibilities, device status, security context, and approved exceptions. This requires the integration of behavioral risk assessment with explicit authorization policy.

The limitations of purely rule-based and purely neural systems suggest the need for a combined approach. Rule-based systems provide consistency, predictability, and understandable explanations, but they are normally limited to situations anticipated by policy designers. Neural systems can identify complex and previously unknown behavioral patterns, but their outputs may be probabilistic, difficult to interpret, and unsuitable as the sole basis for consequential access decisions.

Neuro-symbolic artificial intelligence combines neural learning with symbolic knowledge representation and reasoning. Neural models identify patterns in large and heterogeneous datasets, while symbolic systems represent explicit rules, relationships, constraints, and logical inferences. Neural-symbolic reasoning has been investigated as a method for combining learning and symbolic knowledge within a unified computational framework (Garcez et al., 2009). Logic Tensor Networks provide a more recent example in which first-order logical knowledge is integrated with differentiable machine-learning methods (Badreddine et al., 2022).

In a banking database environment, a neural component may determine that a user's query sequence is unusual, occurs outside normal working patterns, involves an abnormal number of customer records, or differs significantly from the behavior of comparable employees. A symbolic component can then evaluate whether the user has a valid role, an assigned customer relationship, an approved business purpose, a trusted device, an appropriate

authentication level, and the required transaction authority. The resulting decision is based on both learned behavioral evidence and explicit organizational policy.

This paper proposes a Neuro-Symbolic Policy-Aware Database and Insider-Threat framework, referred to as NS-PADIT. The framework is intended to answer the following research question:

How can neural behavioral analysis and symbolic policy reasoning be integrated to provide explainable, continuous, and context-sensitive database access control for banking systems?

The paper makes four principal contributions. First, it presents an architecture that connects behavioral analytics directly to a database policy decision point. Second, it defines a decision model in which mandatory symbolic policies cannot be overridden by neural predictions. Third, it proposes graduated enforcement actions rather than relying only on permit-or-deny decisions. Fourth, it provides an evaluation framework for measuring detection effectiveness, false-positive burden, authorization latency, system robustness, and explanation quality.

The proposed work is conceptual and follows a design-science approach. It does not claim that the architecture has already been deployed in a banking institution or that experimental results have been obtained. The numerical thresholds and performance criteria presented in the paper are evaluation targets that should be validated using synthetic banking workloads, public insider-threat datasets, and controlled organizational pilots.

LITERATURE REVIEW

Role-based access control is one of the most established approaches to organizational authorization. It simplifies administration by associating permissions with organizational roles rather than assigning permissions separately to each employee (Sandhu et al., 1996). This model is suitable for relatively stable banking functions, but role definitions may become excessively broad when employees support multiple products, branches, customers, or temporary projects.

A role may indicate that a user is a relationship manager, for example, but it may not identify which customers the manager is responsible for, whether the requested access has a valid business purpose, or whether the request originates from a trusted device. As organizations become more dynamic, role-based permissions alone may provide insufficient contextual control.

Attribute-based access control determines authorization by evaluating attributes of the subject, protected resource, requested action, and environment against applicable policy rules (Hu et al., 2019). Subject attributes may include role, department, clearance, branch, employment status, and customer assignment. Resource attributes may include data classification, jurisdiction, record type, and sensitivity. Environmental attributes may include time, location, authentication strength, device compliance, and incident status. NIST defines ABAC as a logical access-control method in which authorization decisions are made by evaluating these categories of attributes against policy (Hu et al., 2019).

ABAC provides greater flexibility than role-only models but depends on accurate attributes and carefully designed policies. Incorrect, incomplete, missing, or outdated attributes can result in inappropriate access decisions. Complex policy sets may also contain contradictions, redundant rules, unreachable rules, or unintended exceptions.

Research has investigated techniques for improving ABAC policy administration. Karimi et al. (2022) demonstrated that authorization logs can be analyzed to extract candidate attribute-based access-control policies. Such methods may reduce the manual effort required for policy development. However, automatically extracted policies must still be reviewed because historical access records may contain excessive privileges, policy violations, or temporary exceptions that should not become permanent authorization rules.

Formal validation is also important in complex access-control environments. Caserio et al. (2022) proposed a formal validation approach for XACML 3.0 policies. Formal analysis can assist in detecting policy conflicts, incomplete rules, unexpected decisions, and incorrect combinations of authorization conditions. XACML provides a standardized language for representing attribute-based access-control policies and authorization decisions (OASIS, 2013).Usage control extends authorization beyond the initial access decision by considering ongoing conditions and obligations during resource use (Park & Sandhu, 2004). Under a usage-control model, authorization may continue

to depend on changing attributes after access has begun. This concept is relevant to long-running database sessions because a user's risk level, device status, transaction context, privilege level, or employment status may change during an active session.

For example, a database session may begin from a trusted device under normal conditions. During the session, the user may escalate privileges, access an unusually sensitive table, initiate a large export, or lose authorized employment status. Continuous evaluation allows the security system to modify or terminate access when important conditions change.Zero-trust architecture similarly rejects implicit trust based solely on network location or a previous authentication event. NIST explains that zero trust focuses on protecting individual resources and requires authentication and authorization to be performed before access to a resource is established (Rose et al., 2020). A user or device should not be considered trustworthy merely because it is connected to an internal banking network.

Zero-trust principles support continuous and context-sensitive authorization. Identity, device posture, application identity, resource sensitivity, and other contextual information should influence access decisions. In a database environment, this means that successful authentication at the beginning of a session should not automatically authorize every subsequent query or data export.

Insider-threat research has investigated statistical, probabilistic, and deep-learning approaches. Al Tabash and Happa (2018) applied Gaussian mixture models and sensitivity profiles to model employee behavior and identify unusual activities. Their approach demonstrates how deviations from expected behavior can be used as indicators of potential insider risk.

Elmrabit et al. (2020) proposed a Bayesian-network approach for insider-threat risk prediction. Bayesian networks allow relationships among technical, organizational, and behavioral risk factors to be represented probabilistically. This is useful because insider risk rarely depends on a single event and may instead emerge from a combination of weak indicators.

Nasir et al. (2021) developed a deep-learning-based approach for detecting insider threats from behavioral data. Deep-learning models can identify complex relationships among user activities and may detect patterns that are difficult to represent through manually written rules. However, their predictions may be difficult for security analysts to interpret, particularly when the model does not provide a clear explanation of the events that influenced its output.

Practical deployment creates additional challenges. Erola et al. (2022) examined lessons from deploying an insider-threat detection tool in three multinational organizations. Their findings demonstrate that effective deployment requires organizational procedures, analyst interpretation, suitable alert presentation, and proportionate investigation processes. A technically accurate model may still be ineffective if it produces excessive alerts or fails to explain why a user was considered suspicious.

Public insider-threat datasets are useful for model development and comparison, but they do not fully represent banking systems. General-purpose datasets may not include SQL query structures, customer assignments, payment workflows, separation-of-duty rules, regulatory purposes, privileged database maintenance, or emergency banking operations. Banking-specific evaluation data are therefore necessary before an insider-threat model can be considered suitable for operational use.

Neuro-symbolic AI provides a possible method for integrating learned behavioral patterns with explicit access-control rules. Early neural-symbolic research demonstrated how neural networks and symbolic reasoning could be combined to support learning and logical inference (Garcez et al., 2009). More recent work on Logic Tensor Networks shows how first-order logical knowledge can be incorporated into differentiable learning systems (Badreddine et al., 2022).

In the proposed framework, neural models do not directly make final database authorization decisions. Instead, neural outputs are converted into controlled symbolic facts, such as unusual_time, bulk_read, query_novelty_high, or possible_exfiltration. The symbolic policy engine evaluates these facts together with deterministic permissions, prohibitions, business purposes, separation-of-duty requirements, obligations, and approved exceptions.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 1- Comparison of Relevant Security Approaches

Approach	Main advantage	Main limitation	Adaptability	Explanation
Role-based access control	Simple role administration	Broad and static permissions	Low	High
Attribute-based access control	Context-sensitive authorization	Complex policy and attribute management	Moderate	High
Rule-based threat detection	Predictable alerts	Limited detection of unknown patterns	Low	High
Neural anomaly detection	Learns complex behavior	False positives and limited transparency	High	Low to moderate
Neuro-symbolic access control	Combines learning and policy	Higher implementation complexity	High	High

The literature indicates that behavioral analytics and policy enforcement are usually implemented as separate functions. Access-control systems decide whether a request satisfies policy, while insider-threat systems generate alerts after analyzing activity. This separation creates a delay between detection and enforcement and may produce alerts without sufficient business context.

NS-PADIT addresses this gap by converting validated neural risk indicators into symbolic facts. These facts do not replace the authorization policy. Instead, they are evaluated together with deterministic permissions, prohibitions, obligations, and exceptions. This structure is intended to retain the adaptability of neural learning while preserving the consistency and auditability of symbolic access control.

RESEARCH METHOD

The study follows a design-science research method. Design science is appropriate because the objective is to construct and evaluate a technical artifact that addresses an identified organizational security problem. The artifact consists of an architecture, a formal decision model, policy rules, an explanation mechanism, and an evaluation plan.

The first research activity is problem identification. The identified problem is that static banking database permissions do not adequately detect misuse by authorized users, while standalone anomaly-detection systems cannot consistently translate probabilistic risk into explainable authorization decisions.

The second activity is requirement definition. The proposed framework must satisfy least privilege, continuous authorization, purpose limitation, separation of duties, behavioral risk analysis, explanation, auditability, privacy protection, and operational resilience. It must also avoid allowing a machine-learning model to override mandatory policy or independently determine employee misconduct.

The third activity is artifact design. The architecture is divided into data collection, neural analysis, symbolic reasoning, enforcement, explanation, and audit components. Banking-specific entities such as customers, accounts, payments, cases, change tickets, devices, employees, and service accounts are represented in the policy knowledge base.The fourth activity is demonstration through representative use cases. The use cases include unauthorized customer browsing, bulk data extraction, privileged database administration, payment approval, emergency access, and compromised service accounts.The fifth activity is evaluation. The proposed system should be compared with role-based access control, static attribute-based access control, rule-based insider detection, and neural-only

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

anomaly detection. Evaluation should use both public insider-threat datasets and a synthetic banking database environment.

The framework assumes that the bank has centralized identity management, reliable database auditing, time synchronization, secure policy administration, and tamper-resistant log storage. It also assumes that identity, employment, customer-assignment, device, and transaction attributes are available from authoritative systems.

The framework does not assume that every anomaly represents malicious activity. A high anomaly score is treated as evidence requiring policy interpretation. Mandatory rules remain deterministic. For example, a suspended employee cannot be permitted to access production data simply because the requested action resembles the employee's historical behavior.

The design also follows the principle of proportional response. Moderate or uncertain risk may result in masking, reduced result volume, step-up authentication, or human review. Denial and session suspension are reserved for mandatory policy violations or sufficiently strong combinations of risk and context.

PROPOSED NEURO-SYMBOLIC FRAMEWORK

NS-PADIT contains a telemetry layer, neural intelligence layer, symbolic reasoning layer, enforcement layer, and evidence layer. The telemetry layer collects database audit events, authentication data, identity attributes, device status, employee status, customer assignments, payment workflow information, approved change tickets, and recent security events.

The neural intelligence layer contains specialized models rather than a single unrestricted model. A sequence model analyzes the order of user actions. A query-semantic model analyzes normalized SQL structure, selected tables, sensitive columns, predicates, joins, aggregation, and export operations. A peer model compares the user with employees performing similar functions. A graph model represents relationships among users, devices, applications, databases, customers, accounts, and transactions.

The symbolic reasoning layer evaluates explicit banking policies. These include role permissions, customer-assignment rules, business-purpose restrictions, separation-of-duty requirements, device requirements, working-hour conditions, transaction-value limits, emergency-access rules, and risk-response rules.

The enforcement layer applies the resulting decision at a database proxy, application programming interface, database-native security control, or service authorization gateway. The evidence layer stores the decision, model version, policy version, request fingerprint, supporting attributes, and explanation trace.

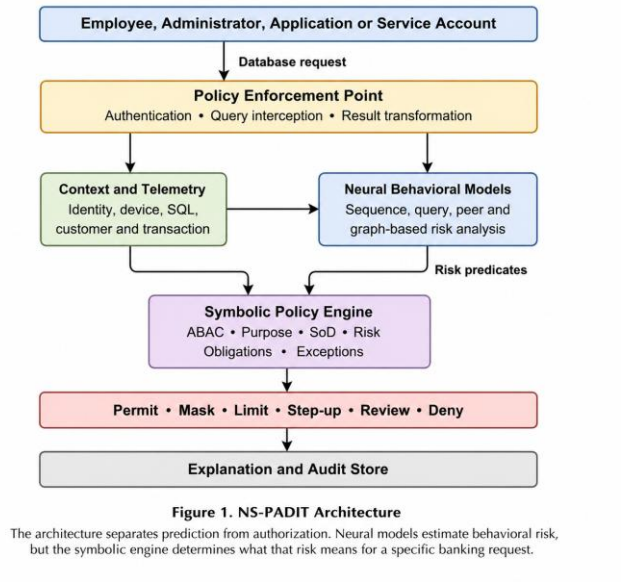

Figure 1- NS-PADIT Architecture

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The architecture separates prediction from authorization. Neural models estimate behavioral risk, but the symbolic engine determines what that risk means for a specific banking request.

The telemetry collected by NS-PADIT must be relevant to security and proportionate to the protected resource. Database events should record the authenticated identity, application, source device, query fingerprint, affected tables, accessed columns, result size, number of records, execution time, export destination, and authorization outcome.

Sensitive SQL literals should be tokenized or pseudonymized before being transferred to the neural environment. For example, a customer number may be replaced with a stable pseudonymous identifier. This allows repeated access patterns to be analyzed without unnecessarily exposing the original identifier to model developers.

Table 2 Inputs Used by Neural and Symbolic Components

Input category	Examples	Neural purpose	Symbolic purpose
Identity	Role, department, branch, status	User and peer profile	Permission and eligibility
Device	Managed state, certificate, risk	Session-risk estimation	Trusted-device requirement
Time	Hour, day, shift, holiday	Behavioral rhythm	Approved access period
SQL query	Tables, columns, joins, operation	Query novelty	Object and action policy
Volume	Rows, bytes, frequency	Exfiltration detection	Export limitations
Customer context	Assignment, branch, active case	Relationship deviation	Purpose authorization
Payment context	Initiator, approver, value	Sequence analysis	Separation of duties
Change context	Ticket, maintenance window	Administrative anomaly	Approved-change policy
Incident context	Case number, emergency state	Risk adjustment	Break-glass exception

The neural outputs are converted into a controlled symbolic vocabulary. Example predicates include unusual_time, query_novelty_high, bulk_read, peer_deviation_high, untrusted_device, customer_relationship_absent, and possible_exfiltration.

A grounding component performs this conversion. It applies approved thresholds, confidence limits, and temporal conditions. For example, an unusually large query may become the predicate bulk_read only when the number of rows exceeds both a policy-defined threshold and the user's validated historical baseline.

Every grounding rule must be versioned and tested. Otherwise, a small modification to a neural threshold could silently alter access-control behavior. Model changes, feature changes, threshold changes, and symbolic policy changes should therefore be managed through a common change-control process.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The framework should also distinguish between human accounts and service accounts. Human models may use role, shift, customer assignment, and peer behavior. Service-account models should focus on application identity, source host, expected stored procedures, query frequency, schema access, and data destination.

INSIDER-THREAT DETECTION MODEL

The insider-threat component analyzes behavior at the event, session, user, peer-group, and organizational levels. Event-level analysis identifies unusual individual queries. Session-level analysis examines sequences such as authentication, privilege escalation, sensitive-table access, bulk selection, compression, and export. User-level analysis compares current behavior with the user's historical pattern. Peer-level analysis compares the user with employees performing similar duties.

Temporal modeling is important because behavior depends on working hours, shift patterns, payroll periods, reporting cycles, month-end processing, and emergency events. Song et al. (2024) showed that both absolute and relative time information can contribute to insider-threat detection.

The framework does not treat a single score as conclusive. It combines independent indicators such as abnormal query structure, unusual time, excessive record volume, absent customer relationship, new device, unexpected location, and recent privilege escalation. Risk should increase when multiple independent indicators support the same hypothesis.

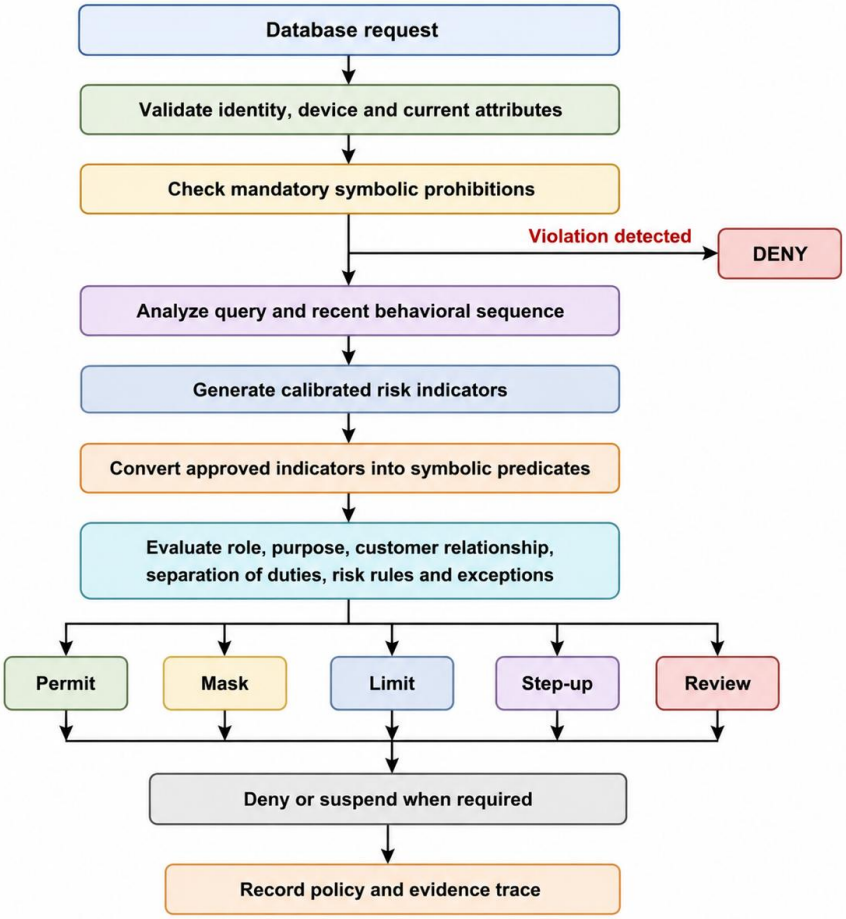

Figure 2. Continuous Authorization Process

Figure 2 Continuous Authorization Process

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The system continuously updates risk during an active session. A session that begins normally may become suspicious after an unusual privilege change, a large query, access to an unrelated customer population, or an attempted export.

Feedback from completed investigations may be used for model improvement, but unresolved alerts should not automatically become malicious training labels. Otherwise, analyst assumptions could be reproduced as model bias. Confirmed benign exceptions should also be recorded so that repeated legitimate activity does not continuously generate unnecessary alerts.

BANKING APPLICATION SCENARIOS

A relationship manager normally accesses customers assigned to the manager's portfolio. If the manager views a small number of unrelated customers, the system may request a case number or stronger authentication. If the manager queries thousands of unrelated customers outside working hours from an unmanaged device, the symbolic engine can deny access because customer relationship, purpose, device, and risk conditions are not satisfied.

A fraud investigator may legitimately access customers outside the investigator's ordinary branch or portfolio. In this situation, an active fraud case provides a valid purpose exception. The access may be permitted with enhanced logging and a limited retention period.

Database administrators require extensive technical privileges, but those privileges should not provide unrestricted access to customer content. An administrator altering an index during an approved maintenance window may be permitted. The same administrator selecting customer identity records without a ticket may be denied even if the SQL command is technically possible.

Payment systems require strict separation of duties. A user who creates a payment should not approve the same payment. Neural analysis may also identify unusual approval timing, unexpected beneficiary patterns, repeated failed approvals, or coordination between accounts. However, deterministic separation-of-duty policy remains the principal control.

Table 3- Illustrative Decisions in Banking Scenarios

Scenario	Policy context	Behavioral evidence	Decision
Manager reads assigned customer	Valid purpose and trusted device	Normal behavior	Permit
Manager reads unrelated customers at midnight	No customer relationship	High time and volume anomaly	Deny and review
Fraud analyst accesses case-linked account	Approved investigation	Unusual but explained	Permit with logging
Administrator changes index with ticket	Approved maintenance	Minor novelty	Permit in monitored session
Administrator reads payment data without ticket	No valid purpose	Moderate anomaly	Deny
User initiates and approves payment	Separation-of-duty violation	Risk score irrelevant	Deny

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Reporting user exports excessive identity data	Report permission exists	Bulk and novelty indicators	Mask, limit, and review
Service account accesses new restricted table	No approved application path	High query novelty	Suspend and investigate

Emergency access requires special treatment. During an operational incident, users may perform unfamiliar actions. A break-glass policy should require strong authentication, an incident ticket, time-limited access, manager notification, full monitoring, and post-event review. Neural novelty should not automatically block an authenticated emergency response when all symbolic exception conditions are satisfied.

EVALUATION FRAMEWORK

The proposed architecture should be evaluated using a combination of the CMU CERT insider-threat dataset, a synthetic banking database workload, and a controlled shadow-mode deployment. The public dataset supports comparison with previous studies, while the synthetic environment provides banking-specific tables, roles, customer assignments, payment workflows, and SQL queries.

The synthetic dataset should contain normal behavior for tellers, relationship managers, credit officers, fraud investigators, payment approvers, auditors, administrators, applications, and service accounts. Threat scenarios should include customer browsing, low-and-slow extraction, bulk export, compromised credentials, administrator misuse, policy-compliant emergency access, payment collusion, and service-account compromise.

The framework should be compared with four baselines: static role-based access control, static attribute-based access control, rule-based anomaly detection, and neural-only insider-threat detection. An additional ablation evaluation should remove individual components such as the temporal model, peer comparison, customer-relationship rules, device predicates, and explanation mechanism.

Table 4- Proposed Evaluation Measures

Evaluation area	**Metric**
Threat detection	Precision, recall, F1 score, AUPRC and AUROC
Operational workload	False alerts per 1,000 user-days
Early intervention	Time from malicious activity to restriction
Policy security	Percentage of prohibited requests permitted
User disruption	Percentage of legitimate requests denied
Performance	Median, 95th and 99th percentile decision latency
Scalability	Requests processed per second
Explanation	Fidelity, completeness and policy coverage
Robustness	Performance under drift, missing data and evasion
Auditability	Reproduction of decisions by model and policy version

Accuracy alone should not be emphasized because insider events are rare. Precision-recall analysis and false alerts per user-day provide a more useful representation of operational performance. Time-based testing should be used to measure how the system responds to behavioral drift.

Journal of Information Systems Engineering and Management
2024, 9(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Target performance values must be established with banking stakeholders. A possible engineering objective is a 95th-percentile decision time below 50 milliseconds for cached, low-risk read requests and below 200 milliseconds for high-risk requests requiring full analysis. These are proposed targets rather than measured results.Analyst studies should evaluate whether explanations reduce investigation time and improve agreement. Security analysts should receive the applicable policy, relevant attributes, grounded risk indicators, confidence values, supporting events, and model version. The study should also test whether analysts can identify incorrect or incomplete explanations.

DISCUSSION

The main advantage of NS-PADIT is the separation of responsibilities between neural and symbolic components. Neural models identify complex patterns that are difficult to encode manually. Symbolic policies determine whether those patterns justify a particular access-control action. This prevents the machine-learning model from becoming an unrestricted decision maker.

The framework can also reduce false positives. An unusual query may be legitimate when connected to an approved investigation, emergency ticket, customer complaint, or system migration. Symbolic context allows the system to recognize these situations instead of treating every deviation as malicious.

Graduated enforcement is another important benefit. Binary denial may interrupt legitimate banking activity, while alert-only systems may allow harmful actions to continue. Masking, query limitation, step-up authentication, monitored execution, and human review provide intermediate controls.

The architecture nevertheless creates several risks. The neural component may be manipulated through data poisoning, adversarial behavior, gradual imitation of normal activity, or compromised training data. NIST identifies poisoning, evasion, privacy compromise, and model extraction among important adversarial machine-learning risks. Training data should therefore be protected, versioned, validated, and excluded when associated with unresolved incidents.

The symbolic component may also fail if policies are incomplete, contradictory, or based on outdated organizational information. Formal policy testing should identify conflicts, unreachable rules, excessive permissions, and unsafe exceptions. Historical access logs should not be treated as automatically legitimate because they may reflect previous over-permission.

Employee privacy is a significant governance concern. Monitoring should be limited to security-relevant information, supported by lawful authority, documented purposes, controlled retention, and restricted investigator access. Sensitive personal characteristics unrelated to information security should not be included in behavioral models.

A high anomaly score must not be treated as proof of malicious intent. Security alerts represent investigative evidence, not a conclusion about employee guilt. Employment action, disciplinary decisions, or legal referrals require authorized human review and corroboration.

The system must also be resilient. A failure of the policy engine should not create unrestricted database access. Mandatory security policies should be cached securely, decision services should be redundant, and fallback behavior should depend on resource sensitivity. Critical writes and sensitive exports should fail securely, while predefined low-risk operations may use recent signed decisions during a temporary service interruption.

CONCLUSION

This paper proposed a neuro-symbolic framework for policy-aware database access control and insider-threat detection in banking systems. The framework addresses a limitation of conventional authorization: a user may possess legitimate permission while using that permission in an abnormal, prohibited, or harmful manner. It also addresses a limitation of standalone machine learning: an anomaly score may lack the policy context and explanation required for a defensible access decision.

NS-PADIT combines database telemetry, neural behavioral analysis, symbolic policies, continuous authorization, graduated enforcement, and explanation traces. Neural models analyze temporal patterns, query semantics, access volume, peer behavior, device context, and entity relationships. Their outputs are converted into controlled risk predicates. A symbolic engine evaluates those predicates together with role, purpose, customer relationship, data classification, separation-of-duty, transaction, and emergency-access rules.

Mandatory prohibitions cannot be overridden by neural predictions. Uncertain risk results should normally produce reversible responses such as masking, result limitation, stronger authentication, or human review. Denial and suspension are appropriate when explicit policy violations or sufficiently strong risk combinations are present.

The proposed framework may improve contextual detection, reduce false-positive alerts, and provide more transparent security decisions. Its effectiveness must nevertheless be demonstrated through realistic banking workloads, public insider-threat datasets, policy testing, analyst studies, and carefully controlled pilots.

Future research should investigate privacy-preserving behavioral analytics, graph-based collusion detection, formal verification of neural-to-symbolic grounding, federated learning, adaptive policy simulation, and secure access control for autonomous banking agents. Further work is also needed to determine how explanation quality affects investigator accuracy and employee procedural fairness.

The framework should be understood as a decision-support and access-enforcement mechanism rather than an automated employee-judgment system. Technical accuracy, policy correctness, privacy, resilience, and human oversight are equally necessary. When these requirements are addressed together, neuro-symbolic AI offers a promising approach for protecting sensitive banking databases against both unauthorized access and misuse of legitimate privileges.

REFERENCES

[1] Al Tabash, K., & Happa, J. (2018). Insider-threat detection using Gaussian mixture models and sensitivity profiles. *Computers & Security, 77*, 838–859. https://doi.org/10.1016/j.cose.2018.03.006

[2] Badreddine, S., d'Avila Garcez, A., Serafini, L., & Spranger, M. (2022). Logic tensor networks. *Artificial Intelligence, 303*, 103649. https://doi.org/10.1016/j.artint.2021.103649

[3] Basel Committee on Banking Supervision. (2021). *Principles for operational resilience*. Bank for International Settlements.

[4] Caserio, C., Lonetti, F., & Marchetti, E. (2022). A formal validation approach for XACML 3.0 access control policy. *Sensors, 22*(8), 2984. https://doi.org/10.3390/s22082984

[5] Elmrabit, N., Yang, S.-H., Yang, L., & Zhou, H. (2020). Insider threat risk prediction based on Bayesian network. *Computers & Security, 96*, 101908. https://doi.org/10.1016/j.cose.2020.101908

[6] Erola, A., Agrafiotis, I., Goldsmith, M., & Creese, S. (2022). Insider-threat detection: Lessons from deploying the CITD tool in three multinational organisations. *Journal of Information Security and Applications, 67*, 103167. https://doi.org/10.1016/j.jisa.2022.103167

[7] Garcez, A. d'Avila, Lamb, L. C., & Gabbay, D. M. (2009). *Neural-symbolic cognitive reasoning*. Springer. https://doi.org/10.1007/978-3-540-73246-4

[8] Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., & Scarfone, K. (2019). *Guide to attribute based access control definition and considerations* (NIST Special Publication 800-162, Update 2). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-162

[9] Karimi, L., Aldairi, M., Joshi, J. B. D., & Abdelhakim, M. (2022). An automatic attribute-based access control policy extraction from access logs. *IEEE Transactions on Dependable and Secure Computing, 19*(4), 2304–2317. https://doi.org/10.1109/TDSC.2021.3054331

[10] Nasir, R., Afzal, M., Latif, R., & Iqbal, W. (2021). Behavioral based insider threat detection using deep learning. *IEEE Access, 9*, 143266–143274. https://doi.org/10.1109/ACCESS.2021.3118297

[11] OASIS. (2013). *eXtensible Access Control Markup Language Version 3.0*. OASIS Standard.

[12] Park, J., & Sandhu, R. (2004). The UCONABC usage control model. *ACM Transactions on Information and System Security, 7*(1), 128–174. https://doi.org/10.1145/984334.984339

[13] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero trust architecture* (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

[14] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. *Computer, 29*(2), 38–47. https://doi.org/10.1109/2.485845