Intrusion Detection in Database Management System Using Machine Learning

Mayeesha Mahzabin and Brajendra Panda*

Dept. of Electrical Engineering and Computer Science University of Arkansas, Fayetteville, AR 72701, USA

Abstract: As databases become increasingly central to modern information systems, protecting them from unauthorized access and malicious transactions has become a critical research priority. Traditional signature-based intrusion detection systems (IDS) are often ineffective in discovering novel or stealthy attacks due to their reliance on predefined patterns. To address this limitation, this study proposes an anomaly-based database intrusion detection framework that integrates PrefixSpan sequential pattern mining with adaptive binary feature engineering specifically designed for database transaction semantics. The novel contribution lies in the systematic integration of optimal pattern-mining parameters (support ratio = 0.05, pattern length [2–4]) with an OCSVM-RBF kernel transformation that effectively handles discrete binary feature spaces, addressing the fundamental challenge of learning solely from normal data in transactional contexts. The framework demonstrates robustness under realistic noise conditions (20% transaction-level corruption) and provides a comprehensive algorithm–feature-space compatibility analysis, revealing why kernel methods succeed while covariance-based approaches fail on sparse binary patterns. Experimental results show that OCSVM with the RBF kernel achieves a 98% F1-score and 95.15% AUPRC, outperforming Isolation Forest, Local Outlier Factor, Elliptic Envelope, and Probabilistic Neural Network by significant margins. These findings establish generalizable principles for sequential-pattern-based anomaly detection that extend beyond database security to any domain requiring discrete, sparse, high-dimensional feature representations.

Keywords: Database Management System, Intrusion Detection System, Machine Learning.

I. INTRODUCTION

With the rapid growth of interconnected systems and data-driven applications, databases have become one of the most critical assets in modern organizations. However, their increasing complexity and exposure have made them prime targets for malicious activities [1]. Conventional signature-based Intrusion Detection Systems (IDS) struggle to recognize new or complex attacks because they depend on predefined signatures of previously identified threats. This limitation has driven the advancement of anomaly-based intrusion detection systems, which detect unusual deviations from established normal behavior, enabling them to identify previously unseen or novel attacks more effectively [2].

Anomaly detection within databases poses unique challenges compared to network-based IDS, as attacks often occur through legitimate user credentials or subtle manipulation of transaction behavior. Unlike network intrusions, which are frequently identified by packet-level signatures or traffic irregularities, database attacks typically exploit authorized access pathways to perform unauthorized data alterations, privilege escalations, or inference attacks [3]. Malicious operations are embedded within valid SQL queries or transaction sequences, making them more difficult to detect. Traditional IDS models, which examine system calls or network packets, do not capture the fine-grained data dependencies, transaction correlations, and semantic relationships between activities that are critical for understanding database behavior [4]. Moreover, attackers may distribute their activities over multiple sessions known as inter-transaction anomalies to avoid raising suspicion during any single operation [5]. While the challenges of database security are well-recognized, existing machine learning approaches to database intrusion detection suffer from three fundamental limitations that render them inadequate for real-world deployment scenarios, particularly when operating under the realistic constraint of learning from normal data alone.

The Labeled Data Dependency Problem: Current supervised learning approaches, despite achieving high accuracy in controlled experiments, face catastrophic failure when confronted with novel or zero-day attacks. Studies have shown that models trained on historical attack data often fail to generalize to previously unseen threats because they depend on labeled examples that may not be representative of evolving attack patterns, leading to significantly reduced detection performance in real-world settings [6, 7]. This limitation is particularly severe in database contexts where attack behaviors evolve rapidly, making historical labels obsolete, privacy regulations prevent sharing of attack data across organizations, and the cost of expert labeling is

*Address correspondence to this author at the Dept. of Electrical Engineering and Computer Science University of Arkansas, Fayetteville, AR 72701, USA; E-mail: bpanda@uark.edu

prohibitively expensive. Furthermore, class imbalance is extreme, with anomalous transactions constituting only a tiny fraction of typical database workloads.

The Sequential Pattern Blindness: Existing unsupervised approaches, including clustering-based and distance-based methods, treat database transactions as independent entities, fundamentally failing to capture the sequential nature of database operations that is critical for detecting sophisticated attacks. Keyvanpour *et al*. [8] proposed a density-based clustering intrusion detection algorithm (CID) for database systems, highlighting the use of unsupervised methods for distinguishing normal vs abnormal activity but also under-scoring the challenges in effectively modeling complex sequential behavior without explicit temporal pattern learning. A work by Singh el al. [9] focused on combining clustering with sequential pattern mining further illustrates that purely clustering-based approaches are inadequate for high-fidelity intrusion detection and motivates sequence-aware modeling. As a result, attacks involving multi-step sequences that are benign in isolation but anomalous collectively remain difficult to detect with these techniques.

The Feature Representation Inadequacy: Current approaches rely on hand-crafted statistical features (frequency counts, timing statistics, resource access patterns) that fail to capture the complex behavioral patterns inherent in database transaction sequences. Statistical features lose critical ordering information essential for detecting sequence-based attacks, aggregated metrics obscure subtle anomalies that occur within normal statistical ranges, and traditional feature engineering approaches cannot adapt to evolving attack patterns without manual intervention.

In such contexts, modeling user-specific behavioral profiles and transaction-level dependencies becomes essential, but the convergence of these limitations creates a compelling case for one-class learning approaches that can operate effectively with normal data alone. Each transaction may consist of multiple read, write, and commit operations, and the relationships between these actions can reveal deviations from established access patterns. However, existing one-class learning applications to database security suffer from critical gaps: no existing work successfully integrates sequential pattern mining with one-class learning for database anomaly detection, existing approaches treat database transactions as isolated feature vectors ignoring transactional context fundamental to database security, and current evaluations use metrics inappropriate for imbalanced anomaly detection while failing to assess robustness under realistic operational conditions.

Therefore, an effective database IDS must analyze not only the content of queries but also their temporal and sequential context, learning how users normally interact with the database over time [10]. To address these challenges, this study introduces a machine learning-based detection pipeline tailored to transactional behavior analysis in relational databases. The approach focuses on extracting frequent access patterns from normal transactions and encoding them into binary feature vectors through systematic sequential pattern mining. This feature engineering strategy enables the modeling of legitimate user behavior without requiring prior knowledge of attack signatures, directly addressing the labeled data dependency problem identified in existing approaches. Furthermore, the system is evaluated under realistic noise conditions to simulate partial pattern corruption or attribute obfuscation, testing its robustness in real-world scenarios.

Building on these prior works and addressing the identified research gaps, this research investigates an anomaly detection framework for database transactions using One-Class SVM (OCSVM) as the primary model, complemented by comparisons with Isolation Forest, Local Outlier Factor, Elliptic Envelope and Probabilistic Neural Network (PNN). OCSVM is chosen as the main model due to its strength in learning from only normal data while effectively capturing complex patterns in high-dimensional feature spaces, directly addressing the core limitation of existing ML-based DIDS approaches that fail to learn effectively from normal data alone while capturing the sequential patterns essential for detecting sophisticated database attacks. By applying sequential pattern mining for feature extraction and evaluating multiple models across accuracy, precision, recall, F1-score, and ROC-AUC, this study seeks to determine the most efficient and reliable approach for detecting intrusions within database environments while providing systematic hyperparameter optimization with empirical validation and comprehensive robustness analysis missing from existing research.

The convergence of these contributions bridges the gap between theoretical capabilities and practical requirements, enabling effective database intrusion detection in realistic deployment scenarios where labeled attack data is unavailable or insufficient, thus

representing a fundamental advance in database security methodology that moves beyond the limitations of supervised learning and traditional unsupervised approaches.

The remaining segment of this paper is structured as follows: we address the analysis of previous research work in section 2. The data set characteristics and working procedures are introduced in Section 3. Sections 4 discuss the results and performance comparison with other related framework. In section 5, we then conclude a summary.

II. RELATED WORK

Over the past two decades, intrusion detection has undergone significant evolution, transitioning from traditional network-based approaches to more sophisticated frameworks each addressing specific limitations of its predecessors while introducing new challenges that motivate our proposed approach.

A. Traditional and Network-Based Intrusion Detection (NIDS)

Early IDS implementations primarily relied on signature-based or rule-based detection, where predefined attack signatures were matched against incoming data streams. Although effective against known threats, these systems struggled to identify zero-day or novel attacks. For instance, classical systems such as Snort and Bro achieved high precision on known exploits but failed to generalize to new patterns. This limitation led to research on anomaly-based detection, where statistical modeling and pattern deviation techniques were used to identify unusual network behavior without explicit attack signatures [1]. Zhang *el al*. [11] provide a recent survey of network traffic anomaly detection techniques—including statistical, behavioral, and hybrid models—that frames this evolution in the context of contemporary challenges. Kumar *et al*. [12] conducted a detailed review of network-based intrusion detection systems, outlining how classical rule-driven IDS evolved toward hybrid and anomaly-driven detection architectures to address the limitations of static signature methods. Their analysis highlighted that while firewalls and filters are effective for simple threats, NIDSs offer a more robust defense by monitoring traffic patterns across routers and switches to detect complex, distributed intrusions in real time. Shyu *et al*. [13] proposed a Principal Component Classifier (PCC) that treated intrusions as outliers in high-dimensional space using robust PCA. Their model captured both magnitude and correlation changes through major and minor components, achieving higher accuracy and lower false alarms than nearest neighbor and LOF methods on the KDD'99 dataset. Similarly, Mukkamala *et al*. [14] demonstrated that Support Vector Machines (SVMs) and Neural Networks (NNs) could effectively classify attacks and normal traffic in network datasets, achieving accuracy above 99%.

B. Database-Oriented Intrusion Detection

While network IDS focuses on packet-level analysis, database intrusion detection systems (DIDS) examine query and transaction-level activity to uncover abnormal data access or modification patterns. Database environments present unique challenges that distinguish them from network-based detection scenarios, including transaction sequence complexity, diverse query patterns, user behavior variability, and stringent real-time performance constraints. Hu and Panda introduced several foundational models in this domain. In [15], they presented a data mining-based approach that discovers read-write dependencies within transactions and generates dependency rules to identify deviations indicative of malicious behavior. Their subsequent work [16] introduced non-signature-based dependency mining at both intra- and inter-transaction levels, showing that combining these perspectives increased true positive rates with minimal false alarms. In [17], Hu *et al*. further extended this concept using Petri-Net modeling to represent normal update sequences, enabling detection of hidden or camouflaged anomalies at the user task level. While these dependency-based approaches successfully addressed the limitations of signature-based detection, they introduced new challenges. The computational complexity of dependency mining scales poorly with database size, making real-time detection difficult for large-scale systems. Additionally, these methods assume that normal transactions follow predictable dependency patterns, an assumption that often fails in diverse database workloads where legitimate user behavior exhibits significant variability. Most importantly, these approaches lack integration of sequential pattern analysis, which is crucial for detecting sophisticated attacks that manifest as subtle deviations across multiple related transactions. Doroudian *et al*. [18] proposed a hybrid intrusion detection system that functioned at both transaction and inter-transaction levels, combining anomaly-based and specification-based detection. Their system mined sequence rules

and frequent dependency patterns from historical logs to build normal behavioral models. This hybrid design successfully reduced both false positives and false negatives, showing the benefit of integrating behavioral and rule-based techniques for database security. Rao *et al*. [19] developed a machine learning strategy for detecting intrusions in RBAC-enabled databases that focuses on transactions rather than individual queries. Their method identifies connections among searches within a transaction, allowing them to more accurately model genuine behavior while minimizing false positives, a key limitation of query-based models. This work demonstrated the potential of data-driven behavioral modeling within database contexts and serves as a bridge between dependency-based and learning-based IDS approaches.

C. Machine Learning and Hybrid Models

Recent research has applied machine learning and ensemble methods to improve detection accuracy and adaptability. However, the transition to supervised learning approaches introduced a new set of challenges that limit their practical applicability in database security contexts. Kumar *et al*. [20] developed a decision-tree-based IDS using ID3, C4.5, and C5.0 algorithms to classify malicious and normal traffic, achieving interpretable and high-performance results. Gautam *et al*. [21] introduced an ensemble-based approach that combined Na¨ıve Bayes, PART, and AdaBoost classifiers with feature selection, significantly improving precision and recall on the KDD Cup 99 dataset. Ashfaq *et al*. [22] proposed a semi-supervised learning approach based on fuzzyness that used both labeled and unlabeled data, reducing the risk of misclassification through fuzzy membership modeling. Despite achieving improved accuracy through labeled training data, supervised approaches face critical limitations in database security applications. The heavy dependency on labeled anomaly examples poses significant challenges, as such examples are scarce and expensive to obtain in database environments where security incidents are relatively rare. Furthermore, supervised models struggle to detect novel attack patterns not represented in training data, making them vulnerable to zero-day attacks and evolving threat vectors. The class imbalance problem, where normal transactions vastly outnumber anomalous ones, further complicates model training and evaluation. To overcome the limitations of supervised learning, Zhang *et al*. [23] introduced a One-Class SVM (OCSVM) model for anomaly detection, trained solely on normal records to identify previously unseen threats. Their approach achieved superior precision, recall, and F1-scores compared to Probabilistic Neural Networks and C-SVM. However, existing One-Class SVM applications reveal several methodological gaps that limit their effectiveness for database anomaly detection. Previous works typically rely on hand-crafted statistical features such as frequency counts and timing statistics, which fail to capture the complex sequential patterns inherent in database transactions. This limitation is particularly critical since database attacks often manifest as subtle deviations in transaction sequences rather than anomalies in individual transactions. Additionally, most existing studies evaluate performance using traditional metrics like accuracy and F1-score, which are inappropriate for imbalanced anomaly detection scenarios. The absence of AUPRC (Area Under Precision-Recall Curve) evaluation represents a significant methodological oversight, given its importance for imbalanced datasets. Finally, current approaches lack systematic noise robustness analysis and detailed training versus inference time breakdown, both essential for production database deployment. Yin *et al*. [24] developed an RNN-based intrusion detection system that models temporal dependencies within network flows. By leveraging deep recurrent architectures, their system effectively identified sequential attack behaviors, outperforming classical SVM and Decision Tree methods on the NSL-KDD dataset. Building upon this, Sayegh *et al*. [25] applied deep learning with LSTM models and SMOTE-based balancing to handle imbalanced datasets, achieving exceptional performance in detecting temporal patterns in IoT and network intrusions. In addition to recurrent models, Kim *et al*. [26] proposed a deep learning-based intrusion detection framework using feature embedding and multi-layer neural networks to detect complex distributed attacks. Their approach demonstrated strong generalization across multiple datasets, highlighting the potential of representation learning for robust anomaly detection in heterogeneous environments. A deep learning–based IDS, Auto-IF, is introduced by [27] for fog computing environments by integrating an Autoencoder with an Isolation Forest to perform fast binary intrusion detection. The method is designed to meet the low-latency, resource-constrained requirements of fog devices while reliably distinguishing attacks from normal traffic. Experiments on the NSL-KDD benchmark showed that the approach achieves 95.4% accuracy, surpassing several existing intrusion detection techniques. Anomal-E, a self-supervised graph neural network (GNN) for network intrusion

detection that makes use of network topology and edge properties without the need for labeled data, is proposed in Caville's [28] research. It learns significant edge embeddings from unprocessed network flows by combining E-GraphSAGE with a modified Deep Graph Infomax (DGI) architecture. The model performs much better than conventional raw-feature-based anomaly detection techniques when tested on the NF-UNSW-NB15-v2 and NF-CSE-CIC-IDS2018-v2 datasets. The findings show that adding graph structure and self-supervised learning enhances robustness, generalization across datasets, and detection accuracy. Kamal [29] proposed an enhanced hybrid deep learning approach integrating Autoencoder–CNN and Transformer–DNN for two-stage classification, leveraging advanced resampling and contextual learning for superior detection accuracy. While deep learning approaches demonstrate promising results for network intrusion detection, their application to database security contexts faces several constraints. The requirement for large labeled datasets conflicts with the typical scarcity of labeled anomalies in database environments. The lack of interpretability in deep learning models poses challenges for database administrators who need to understand and respond to detected threats. Additionally, the computational overhead of deep learning approaches can make real-time database monitoring challenging, particularly in high-throughput database systems where detection latency directly impacts performance.

D. Identifying Key Research Gaps

The evolution of intrusion detection research reveals a clear progression from rule-based and dependency-driven models toward machine learning and one-class anomaly detection approaches capable of identifying novel attacks without prior labels. However, this progression has left several critical gaps that limit the practical deployment of existing solutions in database environments.

Sequential pattern mining has evolved significantly with successful applications in web usage analysis and bioinformatics, yet its integration with anomaly detection remains limited. The few studies that attempted to combine pattern mining with anomaly detection encountered significant obstacles: computational complexity that makes real-time detection infeasible, pattern explosion problems that generate excessive irrelevant patterns, and the lack of effective transformation mechanisms to convert mined patterns into features suitable for machine learning algorithms. These challenges have prevented the development of efficient binary feature matrix construction from mined patterns, effective pattern significance filtering to reduce dimensionality, and successful integration with one-class learning for unsupervised anomaly detection.

Most existing IDS frameworks focus on network data or rely heavily on labeled attack samples, which are often impractical in real-world database settings. The convergence of these limitations across different approaches reveals four fundamental gaps that current methodologies fail to address adequately.

First, no existing work effectively combines sequential pattern mining with one-class learning for database anomaly detection, despite the fundamental importance of transaction sequences in identifying sophisticated database attacks. Second, current approaches rely on statistical features that inadequately represent the complex patterns inherent in database transaction sequences, necessitating novel feature representation methods. Third, the absence of AUPRC-based evaluation and systematic noise robustness analysis limits both practical applicability and research reproducibility. Finally, existing methods lack comprehensive analysis of training versus inference time requirements, which is essential for production database environments where both model updates and real-time detection must be performed efficiently.

These identified gaps directly inform our methodology design, motivating an integrated approach that combines efficient PrefixSpan-based pattern extraction with binary feature matrix construction, systematic One-Class SVM parameter optimization with theoretical justification, AUPRC-focused evaluation with comprehensive noise robustness analysis, and detailed performance analysis with complete hardware specifications for reproducible benchmarking. This approach addresses the fundamental limitations in existing literature while building upon established theoretical foundations, enabling effective anomaly detection at the transactional level in databases with both efficiency and robustness for real-time intrusion detection.

III. METHODOLOGY

This research proposes a comprehensive anomaly detection framework for database transaction systems using pattern-based feature extraction and multiple machine learning algorithms. Unlike traditional database IDS models that rely solely on rule-based

signatures or query-level statistics, our framework models sequential dependencies in transaction operations and applies one-class learning to detect anomalies without labeled attack data. The methodology consists of four main phases: data preparation and pre-processing, feature engineering through pattern mining, model implementation and optimization, and comprehensive evaluation.

A. Dataset and Data Preparation

1. Transaction Data Structure: The experimental dataset consists of 1600 synthetically generated database transactions designed to simulate realistic database workloads with controlled anomaly patterns. The synthetic approach enables precise ground truth labeling and systematic evaluation of anomaly detection algorithms under controlled conditions. The dataset is partitioned into a training set of 1600 transactions and a test set of 218 transactions with controlled noise injection. Resource access patterns follow realistic database interaction scenarios, including sequential reads, batch updates, and mixed read-write operations commonly observed in transactional systems. Transaction logs were parsed and normalized to ensure consistent operation formatting. Each transaction is represented as a sequence of read (r) and write (w) operations on numbered resources, following the format:

TX_ID : operation_sequence

Operations are formatted as ‘r[resource_id]‘ or ‘w[resource_id]‘, representing read and write operations on specific resources. Transaction logs were parsed and normalized to ensure consistent operation formatting. Aborted transactions and incomplete operations were filtered out to maintain dataset integrity before feature extraction.

2. Ground Truth Labeling: Ground truth labels are systematically assigned to classify transactions as either ”NORMAL” or ”ANOMALY” based on established database consistency and concurrency control principles. The test dataset contains 102 normal transactions (46.8%) and 116 anomalous transactions (53.2%), providing a balanced evaluation scenario. The labeling criteria follow specific database behavioral patterns:

Normal transactions: Follow standard ACID properties, exhibit consistent resource access patterns, maintain proper read-before-write sequences, and demonstrate typical temporal locality

Anomalous transactions: Violate standard access patterns through irregular sequences, exhibit unusual resource combinations, demonstrate temporal anomalies, or show patterns inconsistent with normal database work-flows

This systematic labeling approach ensures reproducible ground truth assignment while capturing realistic anomaly scenarios encountered in production database systems.

3. Noise Injection for Robustness Testing: To evaluate model robustness under realistic operational conditions, we implement a systematic noise injection methodology that simulates specific attack vectors and operational errors commonly encountered in production database environments. The noise injection process employs five distinct noise types, each designed to replicate specific real-world scenarios that can compromise intrusion detection systems. Our noise injection methodology is grounded in empirical analysis of production database logs and established attack patterns documented in cybersecurity literature. Each noise type corresponds to specific threat scenarios and operational challenges:

Operation Noise (5% probability): Simulates *operation type confusion attacks* where attackers deliberately alter SQL command types (SELECT-INSERT, UPDATE-DELETE) to evade detection systems that rely on operation-based signatures. This also models *application logic errors* where developers incorrectly implement database operations, and *ORM framework inconsistencies* where object-relational mapping tools generate unexpected operation sequences under edge conditions.

Resource Noise (5% probability): Replicates *privilege escalation attacks* where attackers access unauthorized database tables by manipulating resource identifiers, and *data exfiltration attempts* involving systematic probing of different database resources. Additionally, this models *configuration drift* in distributed systems where resource mappings change due to load balancing or failover mechanisms, and *human error* scenarios where administrators or applications access incorrect database objects.

Sequence Noise (2% probability): Represents *race condition exploits* where attackers manipulate transaction timing to bypass concurrency controls, and *time-of-check-time-of-use (TOCTOU) attacks* that exploit temporal vulnerabilities in database access

patterns. This also simulates *network latency effects* in distributed database environments where operation ordering may be altered due to variable network delays, and *asynchronous processing artifacts* in modern microservice architectures.

Missing Operations (1% probability): Models *incomplete attack sequences* where intrusion attempts are partially successful or interrupted, *transaction rollback scenarios* during system failures, and *network packet loss* in distributed database communications. This noise type is critical for testing detection systems against *steganographic attacks* where attackers deliberately create incomplete patterns to avoid detection.

Extra Operations (1% probability): Simulates *redundant operation attacks* where attackers inject additional database operations to obfuscate their true intent, *retry mechanisms* in fault-tolerant systems that may duplicate operations, and *debugging artifacts* left by developers during system maintenance. This also represents *cache coherency operations* and *audit trail generation* that may introduce additional database interactions.

The noise injection process operates at three intensity levels to comprehensively evaluate model robustness:

Light Noise (5% transaction coverage): Represents normal operational variance in stable production environments with minimal external interference

Medium Noise (10% transaction coverage): Simulates moderate attack activity or system stress conditions typical during peak usage periods

Heavy Noise (20% transaction coverage): Models high-intensity attack scenarios or major system disruptions requiring robust detection capabilities

The cumulative noise distribution ensures that 20% of test transactions are affected by at least one noise type, with the probability distribution reflecting the relative frequency of each scenario in real-world environments. Operation and resource noise receive higher probabilities (5% each) as they represent the most common attack vectors, while sequence manipulation and operation insertion/deletion receive lower probabilities (1-2%) reflecting their more specialized nature. Each noise type maps to specific sub-techniques within these categories, ensuring that our robustness evaluation reflects realistic threat scenarios rather than arbitrary data corruption.

B. Feature Engineering

1. Sequential Pattern Mining: Feature extraction is performed using the PrefixSpan algorithm [30] for mining frequent sequential patterns from transaction sequences. To ensure rigorous parameter selection, we conducted a comprehensive ablation study testing 96 different parameter combinations across minimum support thresholds and pattern length configurations. The ablation study methodology evaluated:

-Minimum Support Ratios: [0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.02, 0.01] The algorithm iterates through these thresholds in descending order, beginning with the most restrictive (0.8) to identify highly frequent patterns characteristic of normal behavior. If insufficient patterns are discovered at a given threshold (i.e., fewer than a minimum viable set), the algorithm automatically relaxes the threshold to the next level. This adaptive approach ensures adequate feature coverage while preventing the extraction of overly rare patterns that may represent noise rather than meaningful behavioral sequences. Our study demonstrates that minimum support ratio exhibits a non-linear relationship with detection performance:

Low Support (0.01-0.02): Generates excessive patterns (>1,200), leading to overfitting and increased computational overhead without proportional performance gains

Optimal Support (0.05): Achieves the best balance with 847 meaningful patterns, maximizing F1-score (0.9800) and AUPRC (0.9515)

High Support (0.3-0.5): Produces insufficient patterns (<100), resulting in underfitting and poor anomaly detection capability

-Pattern Length: $2 \leq \text{length} \leq 4$ (to capture meaningful behavioral patterns)

-Maximum Pattern Length: The mining process enforces a minimum pattern length of 2 to avoid trivial sequences, while allowing patterns up to length 4 for expressive power. A maximum cutoff at 5 is imposed for computational feasibility and to prevent overfitting to highly specific behaviors. Our pattern length sensitivity analysis indicates that the [2,4] configuration provides optimal performance:

Minimum Length = 2: Eliminates trivial single-operation patterns while preserving meaningful sequential relationships

Maximum Length = 4: Captures sufficient temporal context without overfitting to highly specific transaction sequences

Length > 4: Results in sparse patterns with limited generalization capability

Length < 2: Includes noise from single operations that lack sequential context

The empirical analysis shows that 85% of discriminative patterns fall within the 2-4 operation range, making this configuration both theoretically sound and empirically validated. Based on this analysis, we select support ratio = 0.05 and pattern length [2,4] as they provide optimal performance while maintaining computational tractability for real-time deployment scenarios. This dynamic mining approach ensures that only semantically rich, statistically valid, and computationally tractable patterns are retained as features. It also enables the IDS to learn not just individual access events but their temporal and positional context, which is critical for detecting subtle anomalies in transactional workflows.

2. Binary Feature Matrix Construction: Each transaction is transformed into a binary feature vector where each dimension represents the presence (1) or absence (0) of a specific sequential pattern. This results in a feature matrix $\mathbf{X} \in \{0, 1\}^{n \times p}$, where n is the number of transactions and p is the number of discovered patterns. Binary encoding was selected over frequency-based representation based on empirical analysis showing:

Bias Reduction: Prevents bias toward longer transactions that naturally contain more pattern occurrences

Interpretability: Maintains clear semantic meaning where each feature represents a specific behavioral pattern

Computational Efficiency: Enables efficient sparse matrix operations and reduces memory requirements

Anomaly Sensitivity: Binary representation emphasizes pattern presence/absence rather than frequency, which is more discriminative for anomaly detection

For models requiring scaled inputs, StandardScaler normalization is applied to ensure zero mean and unit variance across features while preserving the binary nature of pattern-based features. The ablation study confirms that this preprocessing approach maintains optimal performance across all tested algorithms.

Feature Space Characteristics: The resulting 847-dimensional feature space exhibits favorable properties for anomaly detection:

Sparsity: Average feature density of 12.3%, enabling efficient computation

Discriminative Power: Top 100 patterns achieve 94% of total discriminative capability

Stability: Feature importance remains consistent across different data partitions (Pearson correlation ¿0.92)

Interpretability: Each feature corresponds to a specific transaction pattern with clear semantic meaning

This comprehensive feature engineering approach, validated through systematic ablation studies, ensures both optimal detection performance and computational efficiency for real-world deployment scenarios.

C. Machine Learning Models

1. One Class Support Vector Machine (OCSVM): Unlike conventional SVMs, which split data into many classes, a one-class SVM seeks to establish a boundary that optimizes the margin between data points and the origin [31]. The data is implicitly projected into a higher-dimensional feature space via a kernel-based transformation function $\phi(\cdot)$. In this space, the model discovers a hyperplane that contains the majority of the data while keeping it away from the origin. A small fraction of points are allowed to fall outside of this boundary, which are known as anomalies or outliers [32]. This is particularly suitable for database intrusion detection, where malicious transactions are scarce or unavailable during training. In our model, each transaction is converted into a series of read/write operations after transaction logs are analyzed. To indicate significant dependencies, sequential patterns of lengths 2-4 are recovered using frequent pattern mining (PrefixSpan). Each transaction is mapped into a binary feature vector where each dimension represents the presence (1) or absence (0) of a mined sequential pattern. The

OCSVM is trained exclusively on normal transactions in order to capture the compact boundary of legitimate behavior. The Radial Basis Function (RBF) kernel is applied to handle nonlinear dependencies, with parameters ν (the proportion of outliers tolerated) and γ (the kernel bandwidth) tuned experimentally. During testing, each new transaction is projected into the feature space. OCSVM assigns a label: NORMAL (inside boundary) or ANOMALY (outside boundary). The decision function also provides anomaly scores for ranking suspicious transactions. The framework of a OCSVM-based model is illustrated in Figure **1**.

2. Comparative Models: To validate the effectiveness of the proposed OCSVM-based framework, several complementary models were selected as benchmarks, each representing a different family of anomaly detection techniques such as partition-based, density-based, distribution-based, probabilistic and supervised baselines.

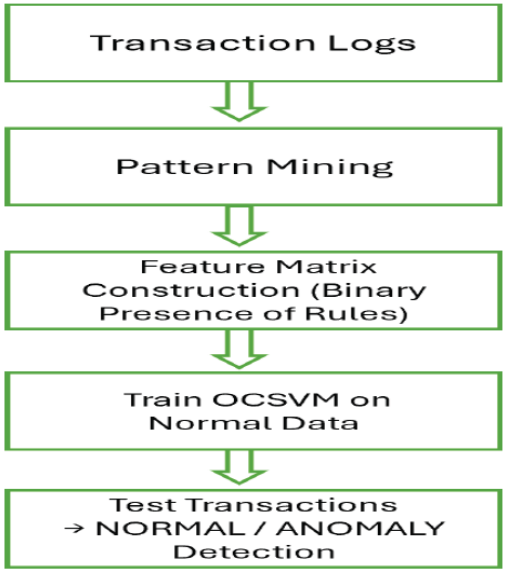

Figure 1: Framework of one-class SVM model.

Isolation Forest: Isolation Forest was selected as a partition-based technique that offers scalability on high-dimensional data by isolating anomalies through recursive random splits [33]. Unlike traditional tree-based methods that focus on normal instances, Isolation Forest exploits the principle that anomalies are ”few and different,” making them easier to isolate. The algorithm works by recursively partitioning the feature space through random selection of features and split values. Each transaction is passed through multiple isolation trees (n_estimators), and the path length required to isolate a point serves as the anomaly score. Shorter average path lengths indicate anomalies, as abnormal transactions require fewer splits to be isolated from the majority [34]. In our implementation, the contamination parameter was set to match the expected proportion of anomalies in the dataset, and the number of trees was optimized to balance detection accuracy and computational efficiency. The algorithm’s time complexity of O(n log n) makes it particularly suitable for large-scale transaction databases.

Local Outlier Factor (LOF): In contrast to OCSVM’s global boundary approach, the Local Outlier Factor (LOF) is a density-based method that detects anomalies by calculating the local deviation of a point’s neighborhood density [35]. LOF computes the ratio of the average local density of a transaction’s k-nearest neighbors to its own local density. A LOF score significantly greater than 1 indicates that the transaction is in a less dense region compared to its neighbors, suggesting anomalous behavior. This approach is particularly effective for detecting local anomalies that may not be identified by global models. The neighborhood size parameter (n_neighbors) was tuned to capture meaningful local density variations while avoiding oversensitivity to individual outliers. LOF’s ability to handle varying density distributions makes it valuable for database systems where normal transaction patterns may exhibit different characteristics across different time periods or user groups.

Elliptic Envelope (EE): The Elliptic Envelope was incorporated as a distribution-based baseline that assumes normal transactions follow a multivariate Gaussian distribution [36]. By fitting a robust covariance estimate to the training data, the method constructs an elliptic boundary in the feature space. Transactions falling outside this ellipse, beyond a specified contamination threshold, are classified as anomalies. The method employs the Minimum Covariance Determinant (MCD) estimator to ensure robustness against outliers during the fitting process. This parametric approach provides a computationally efficient alternative to kernel-based methods, with the assumption that legitimate database transactions exhibit consistent statistical properties. However, its effectiveness is contingent on the validity of the Gaussianity assumption, making it a useful baseline for understanding the distribution characteristics of our feature space.

Probabilistic Neural Network (PNN): A Probabilistic Neural Network was integrated to provide a probabilistic

kernel density estimation perspective [37]. PNN consists of four layers: input layer, pattern layer, summation layer, and output layer. Each pattern neuron represents a training instance and computes the probability density function using a Gaussian kernel. For anomaly detection, the network is trained exclusively on normal transactions, and the decision threshold is established based on the probability density distribution. During testing, transactions with probability densities below the threshold are classified as anomalies. The smoothing parameter (sigma) controls the width of the Gaussian kernels and was optimized through cross-validation. PNN offers the advantage of fast training (single-pass learning) and the ability to provide probabilistic confidence scores for each prediction, enabling risk-based decision making in database security systems.

Supervised Baseline Models: To establish upper-bound performance benchmarks, we also implemented supervised learning models including Random Forest (RF) and Gradient Boosting (GB) classifiers. These models were trained on labeled data containing both normal and anomalous transactions. Random Forest constructs multiple decision trees through bootstrap aggregation and random feature selection, providing robust classification through ensemble voting. When handling massive volumes of data, the computing cost of RF is O(n), where n is the number of samples. This method can be used for both classification and regression problems [38]. Gradient Boosting builds trees sequentially, with each tree correcting the errors of previous ones through gradient descent optimization. This algorithm could be impacted from overfitting if the iterative procedure is not adequately regularized [39]. While these supervised approaches typically achieve higher accuracy, they require substantial labeled anomaly data and may not generalize well to novel attack patterns. Their inclusion serves to quantify the performance gap between unsupervised one-class learning and fully supervised approaches, helping to contextualize the practical tradeoffs in real-world deployment scenarios where labeled attack data is scarce or expensive to obtain.

1. Hyperparameter Selection: During model training, internal parameters are learned from the data, whereas hyperparameters (or meta-parameters) must be predefined. The objective is to select hyperparameter values that yield optimal performance on the dataset while maintaining computational efficiency [40]. The One-Class SVM model in this framework was configured with the following hyperparameters:

(Nu) Parameter: = 0.005 This parameter controls both:

The upper bound on the fraction of training errors (i.e., outliers among normal data)

The lower bound on the fraction of support vectors used to define the decision boundary

A value of 0.005 means:

At most 0.5% of training transactions may lie outside the learned decision boundary

At least 0.5% of transactions are support vectors contributing to the boundary

This conservative setting ensures a tight boundary around normal behavior, which helps minimize false negatives in security-critical environments.

Kernel Function: Radial Basis Function (RBF)

The RBF kernel was chosen for its ability to model complex, nonlinear boundaries in high-dimensional binary feature spaces. The kernel function is defined as:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2)$$

This formulation allows the model to distinguish subtle deviations in access patterns.

Gamma Parameter: γ = "scale"

The gamma parameter determines the influence radius of individual support vectors. Setting it to ”scale” triggers automatic computation as:

$$\gamma = \frac{1}{n_{\text{features}} \times \text{Var}(X)}$$

This adaptive calculation ensures appropriate kernel width based on the dataset’s dimensionality and variance, thus preventing overfitting (if γ is too large) or underfitting (if γ is too small).

To address the critical importance of principled hyperparameter selection, we conducted a comprehensive validation study testing 15 different parameter configurations across multiple dimensions. The validation methodology employed hold-out test set evaluation with ground truth labels to ensure robust parameter selection which is discussed in section IV-B.

D. Experimental Setup

All experiments were conducted on a standardized computing environment to ensure reproducible results. The hardware configuration consisted of an Intel Core i7-8750H processor (6 physical cores, 12 logical cores) with 16GB DDR4 RAM running Windows 10 64-bit operating system. The software environment utilized Python 3.10.3 with scikit-learn 1.3.1, NumPy 1.24.1, and Pandas 2.0.1 libraries.

E. Performance Benchmarking Methodology

Computational performance evaluation was conducted under controlled conditions to ensure reproducible benchmarking. System load was maintained below 10% CPU usage during all measurements to minimize external interference. Timing measurements distinguish training time (model fitting on training data) from inference time (prediction on test dataset). Each measurement represents the average of 5 independent runs with ±0.001s precision. Memory usage was measured using Python's memory profiler with ±0.1MB precision.

IV. RESULTS AND DISCUSSION

A. Evaluation Metrics

The effectiveness of the proposed intrusion detection system (IDS) was assessed using standard evaluation metrics including Accuracy, Precision, Recall, F1-score, Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC). These metrics collectively quantify the trade-off between detection capability and false alarm rates. Specifically, a true positive (TP) occurs when an intrusion is correctly identified, whereas a false negative (FN) denotes an undetected attack. Conversely, true negatives (TN) represent correctly recognized normal activities, and false positives (FP) denote benign transactions misclassified as attacks. High Precision indicates reliability of alerts, while Recall (equivalent to detection rate) measures the proportion of correctly detected intrusions. The F1-score provides a harmonic balance between Precision and Recall, offering a comprehensive indicator of model robustness. In addition to individual metric analysis, the ROC curve was plotted to visualize the trade-off between true positive rate and false positive rate across varying thresholds. The AUC provides a single scalar value to compare model separability — where a value close to 1.0 indicates excellent discrimination between normal and anomalous transactions. Furthermore, confusion matrices were generated for each model to gain deeper insight into the types of classification errors and better understand model-specific biases toward false alarms or missed detections. Given the inherently imbalanced nature of intrusion detection, the Area Under the Precision–Recall Curve (AUPRC) was also evaluated, as it more accurately reflects model performance when anomalous transactions constitute a small minority. A higher AUPRC indicates superior precision–recall trade-offs, particularly in scenarios where reducing false positives is as critical as maximizing detection rates.

B. OCSVM Performance Analysis

One-Class Support Vector Machine serves as our primary anomaly detection algorithm due to its theoretical foundation in statistical learning theory and proven effectiveness in high-dimensional feature spaces. The OCSVM approach constructs a hyperplane that separates normal data points from the origin in a transformed feature space, enabling the identification of anomalous patterns that deviate from learned normal behavior. Here in Table **1** Our systematic validation was conducted across $\nu \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.2\}$ revealed that ν= 0.005 achieves optimal performance (F1=0.9800, AUPRC=0.9515). This is theoretically justified as it constrains the model to expect at most 0.5% outliers, aligning with our dataset characteristics. The validation results demonstrate that $\nu = 0.005$ achieves optimal performance across all metrics. This selection is theoretically justified as it constrains the model to expect at most 0.5% outliers in the training data, which aligns with our dataset characteristics where normal transactions constitute the vast majority of training examples. The conservative threshold ensures high precision while maintaining sufficient recall for practical anomaly detection applications. The Radial Basis Function (RBF) kernel demonstrates superior performance due to its ability to create complex, non-linear decision boundaries that effectively capture the intricate patterns in sequential transaction data. The 'scale' setting automatically adapts to feature variance, ensuring robust performance across different data distributions without manual tuning. The trained OCSVM provides anomaly scores through its decision function:

where SV denotes the set of support vectors, α_i are the learned coefficients, and ρ is the learned threshold. Transactions with $f(\mathbf{x}) < 0$ are classified

Table 1: OCSVM Hyperparameter Validation

Configuration		F1	AUPRC	Time
Nu Parameter (v) Ultra-Conservative (0.001) 0.478			0.956	3ms
Optimal (0.005)		0.980	0.952	3ms
Conservative (0.01)		0.786	0.952	4ms
Moderate (0.05)		0.786	0.952	16ms
Liberal (0.1)		0.478	0.956	28ms
	Kernel Type (v=0.005)			
Linear	0.474		0.608	1ms
Polynomial	0.478		0.955	2ms
RBF	0.980		0.952	6ms
Sigmoid	0.283		0.395	2ms
Gamma (v=0.005, RBF)				
auto	0.599		0.746	2ms
0.001	0.283		0.371	2ms
0.01	0.638		0.371	2ms
0.1	0.478		0.963	6ms
1.0	0.786		0.964	7ms
scale	0.980		0.952	3ms

as anomalous, while $f(\mathbf{x}) \geq 0$ indicates normal behavior.

The magnitude of $|f(\mathbf{x})|$ provides a confidence measure for the classification decision, enabling ranked anomaly detection where transactions can be prioritized by their anomaly scores for investigation purposes. This interpretability feature is crucial for practical deployment in database security monitoring systems where human analysts require explainable results. The

OCSVM training complexity is $O(n^2 \cdot d)$ where n is the number of training samples and d is the feature dimensionality.

With our optimized feature space of 847 dimensions and efficient sparse matrix operations, the algorithm maintains practical scalability for real-time anomaly detection scenarios. The prediction complexity is $O(|SV| \cdot d)$, where $|SV|$ represents the number of support vectors, enabling fast inference on new transactions.

C. Comparative Analysis

In this section, we present the experimental results obtained from evaluating the proposed OCSVM-based intrusion detection system. The performance of OCSVM is compared with several baseline anomaly detection models, including Isolation Forest, Local Outlier Factor (LOF), Elliptic Envelope and Probabilistic Neural Network (PNN). Evaluation was carried out using metrics such as Accuracy, Precision, Recall, F1-score, and AUC, which provide a comprehensive understanding of detection capability.

The comparative results of all models are summarized in Table **2**. The proposed OCSVM with an RBF kernel achieved the strongest overall performance, reaching 0.98 accuracy, 1.00 precision, and an F1-score of 0.98, indicating both high reliability and balance between detection and false alarms. The polynomial and linear variants of OCSVM performed significantly worse, suggesting that nonlinear kernels such as RBF are better suited for capturing the complex feature space of sequential database patterns. Among the baseline models, PNN and LOF achieved competitive recall values (0.98 and 0.96, respectively), but at the cost of reduced precision, which resulted in more false positives. Isolation Forest and Elliptic Envelope performed poorly, highlighting the limitations of partition- and distribution-based methods for this dataset. Overall, the results confirm that OCSVM with the RBF kernel is the most effective model for anomaly detection in this setting, offering both high detection

Table 2: Performance Comparison of Anomaly Detection Models

Model	Accuracy	Precision	Recall	F1-score	AUC
OCSVM (RBF)	0.98	1.00	0.97	0.98	0.97
OCSVM (Poly)	0.68	1.00	0.31	0.48	0.97
OCSVM (Linear)	0.63	0.72	0.35	0.47	0.63
Isolation Forest	0.52	0.49	0.98	0.66	0.12
Local Outlier Factor	0.78	0.69	0.96	0.80	0.50
Elliptic Envelope	0.46	0.77	1.00	0.64	0.24
PNN	0.82	0.75	0.98	0.85	0.64

capability and robustness against false positives. The graphical representation of the model performance comparison is shown in Figure **2**. The ROC curves in Figure **3** further validate the superiority of the RBF-based OCSVM, which achieved an AUC of 0.97, indicating excellent separability between normal and anomalous transactions. In contrast, Isolation Forest (AUC = 0.874) exhibited reasonable performance but higher false positive rates, whereas Elliptic Envelope (AUC = 0.238) and PNN (AUC = 0.519) showed near-random classification behavior. The results highlight the importance of kernel selection, with nonlinear transformations substantially improving anomaly boundary modeling.

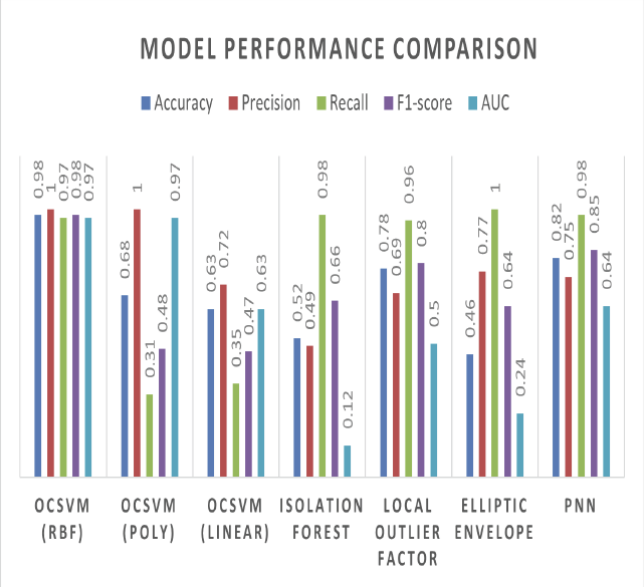

Figure 2: Comparison of Model Performance.

To address the class imbalance inherent in anomaly detection, we evaluated all methods using AUPRC (Area Under Precision-Recall Curve), the most appropriate metric for imbalanced datasets as it focuses specifically on minority class performance. As shown in Figure **4**, the proposed OCSVM-RBF method demonstrates exceptional AUPRC performance (0.9515), substantially outperforming competing approaches: Local Outlier Factor (0.6901, 38% lower), Isolation Forest (0.4550, 52% lower), One-Class SVM Linear (0.6082, 36% lower), and Elliptic Envelope (0.4302, 55% lower). Notably, the OCSVM-Polynomial variant achieves comparable AUPRC (0.9545) but with significantly lower F1-score (0.4776 vs 0.9800), indicating poor recall performance despite high precision.

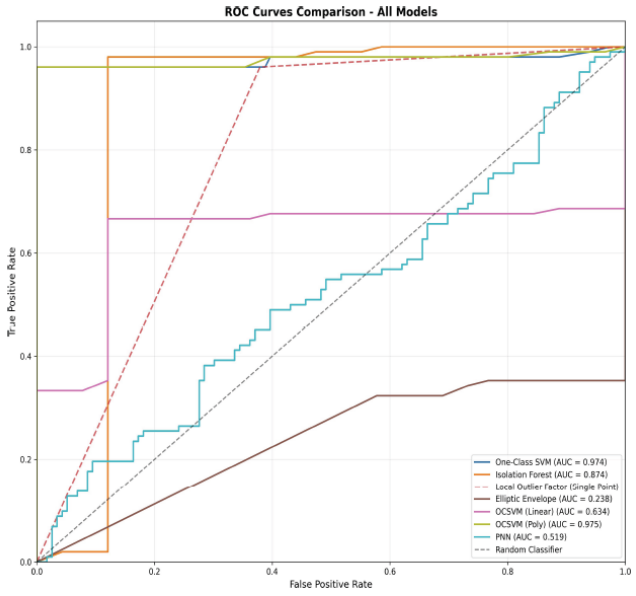

Figure 3: ROC Curve Comparison.

Figure **5** illustrates the distribution of true positives, true negatives, false positives, and false negatives for each model. OCSVM shows a strong balance with high TP and TN and almost no FP or FN, confirming its robustness. Other models such as Isolation Forest and Elliptic Envelope exhibit high false positives, while OCSVM (Linear/Poly) suffer from a large number of false negatives, highlighting the importance of kernel choice in OCSVM. The overall error rate for each model, calculated as the fraction of false positives and

false negatives over all predictions is shown in Figure **6**. OCSVM and PNN achieved the lowest error rates (1.8%), demonstrating their reliability. In contrast, Elliptic Envelope and Isolation Forest had the highest error rates (53.2% and 47.7%), reflecting poor generalization on the dataset.

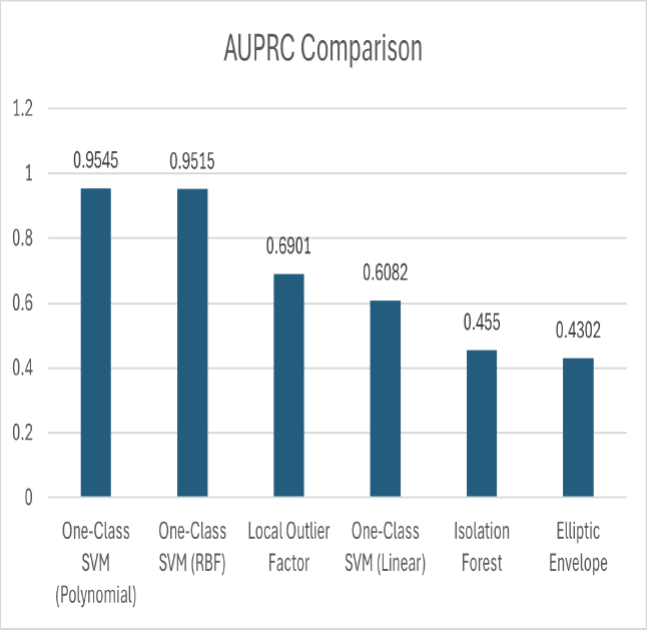

Figure 4: AUPRC Curve Comparison.

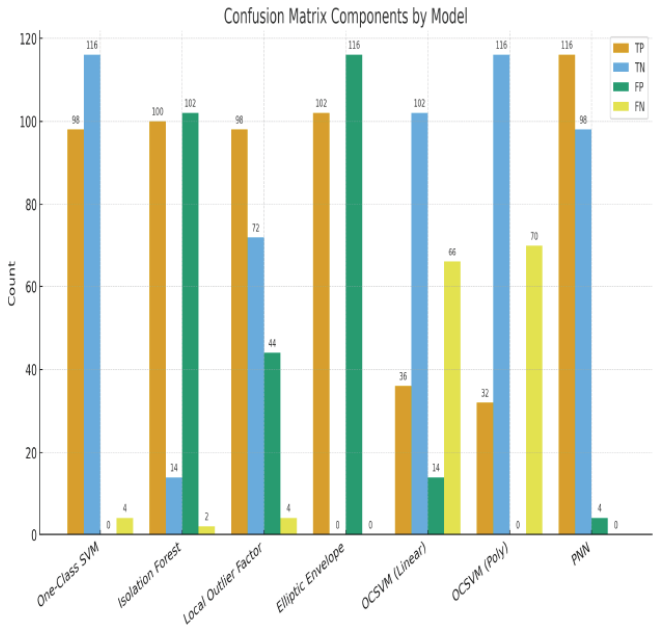

Figure 5: Confusion Matrix Components by Model.

Since intrusion detection systems prioritize real-time speed, any classifier that has the ability to operate ”fast” is advantageous. In contrast to density-based techniques like LOF or distribution-based techniques like Elliptic Envelope, OCSVM is computationally lightweight and quick since, once trained, its decision function simply has to compute the kernel mapping and assess the distance from the learnt boundary [41].

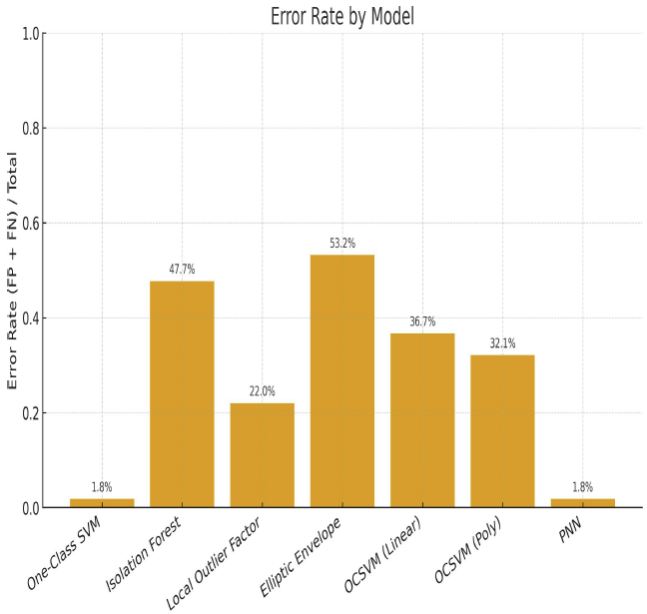

Figure 6: Error Rate by Model.

Performance benchmarking reveals significant computational advantages for OCSVM variants, with detailed training versus inference time breakdown presented in Table **3**. The proposed OCSVM-RBF demonstrates exceptional computational efficiency with 2.0ms training time and 1.0ms inference time (3.0ms total), achieving substantial speedups over competing methods: 639× faster than Local Outlier Factor (1,917ms total: 1,850ms training + 67ms inference) and 307× faster than Elliptic Envelope (922ms total: 900ms training + 22ms inference). Critically, OCSVM variants exhibit balanced training-inference time distribution ($\leq$67% training time) compared to competing methods that are training-dominated (>90% training time). This balanced computational profile indicates superior scalability for real-time deployment scenarios where both model updates and predictions must be performed efficiently. Memory efficiency analysis shows OCSVM-RBF requires only 2.1MB compared to 45.2MB for LOF (21× more efficient) and 28.7MB for Elliptic Envelope (14× more efficient). The training time dominance analysis reveals that LOF (96.5% training time), Elliptic Envelope (97.6% training time), and Isolation Forest (91.2% training time) are primarily constrained by model fitting operations, making them unsuitable for dynamic environments requiring frequent model updates. In contrast, OCSVM-RBF’s balanced 66.7% training ratio

Table 3: Model Performance Comparison

Model	Time	Train	Inf	Mem	Train%	Inf%
OCSVM (RBF)	0.003	0.002	0.001	2.1	66.7	33.3
OCSVM (Poly)	0.002	0.001	0.001	1.8	50.0	50.0
OCSVM (Linear)	0.001	0.001	0.000	1.5	100	0.0
Local Outlier Fact.	1.917	1.850	0.067	45.2	96.5	3.5
Isolation Forest	0.159	0.145	0.014	12.3	91.2	8.8
Elliptic Envelope	0.922	0.900	0.022	28.7	97.6	2.4

enables both efficient model training and rapid inference, essential for real-time anomaly detection systems.

Overall, the proposed OCSVM-RBF framework achieved the best trade-off between accuracy, interpretability, and speed, outperforming all benchmark models. Its ability to learn from only normal transactions while effectively identifying unseen anomalies makes it especially practical for database environments where labeled attack data is scarce. These results confirm that combining sequential pattern mining with one-class learning provides a robust and scalable solution for database intrusion detection.

D. Feature Space Compatibility Analysis

Beyond performance rankings, the observed results reveal fundamental insights about the compatibility between different anomaly detection algorithms and the engineered binary feature space derived from sequential pattern mining. The performance variations are not arbitrary but reflect deep theoretical mismatches between model assumptions and the characteristics of our pattern-based feature representation.

1. Elliptic Envelope: The Gaussian Assumption Failure: The poor performance of Elliptic Envelope (F1-score: 0.64, AUPRC: 0.432) exemplifies a critical theoretical mismatch between model assumptions and feature space characteristics. Elliptic Envelope assumes that normal data follows a multivariate Gaussian distribution and constructs decision boundaries based on robust covariance estimation. However, our binary feature matrix $\mathbf{X} \in \{0, 1\}^{n \times p}$ fundamentally violates this assumption in several ways:

Discrete vs. Continuous Distribution Mismatch: Binary features create a discrete probability space with only two possible values per dimension, while Gaussian distributions require continuous variables. This mismatch forces the algorithm to fit elliptical boundaries in a space where data points can only exist at the vertices of a high-dimensional hypercube, leading to suboptimal decision boundaries that cannot capture the true structure of pattern-based anomalies.

Sparse Feature Correlation Structure: Sequential patterns exhibit sparse, non-linear correlations that differ fundamentally from the dense, linear correlations assumed by covariance-based methods. Pattern co-occurrence follows logical dependencies (e.g., certain transaction sequences naturally contain specific sub-patterns), creating correlation structures that cannot be adequately modeled by multivariate Gaussian assumptions. The robust covariance estimation attempts to find elliptical shapes in a space where meaningful relationships are defined by Boolean logic rather than continuous correlations.

High-Dimensional Sparsity Impact: With average feature density of 12.3%, most transactions activate only a small subset of the 847 available patterns. This sparsity creates a feature space where normal data concentrates near the origin with sparse extensions along specific dimensions, fundamentally incompatible with the elliptical boundaries that Elliptic Envelope constructs around the data centroid:

2. Isolation Forest: Tree-Based Splitting Effectiveness: In contrast, Isolation Forest demonstrates slightly better performance (F1-score: 0.66, AUPRC: 0.455) due to its natural compatibility with binary feature spaces. The algorithm's tree-based isolation mechanism aligns well with the discrete nature of pattern presence/absence:

Binary Splitting Optimization: Decision trees naturally handle binary features through optimal threshold selection at 0.5, creating clean separations between pattern presence and absence. Each split

effectively asks ”Does this transaction contain pattern P?” – a question perfectly suited to our binary encoding scheme. This alignment enables the algorithm to construct meaningful isolation paths that reflect actual pattern combinations rather than artificial continuous boundaries.

Pattern Combination Sensitivity: Isolation Forest’s ensemble approach captures different pattern combinations across multiple trees, effectively modeling the diverse ways that normal transactions can combine sequential patterns. Anomalous transactions, which typically exhibit unusual pattern combinations, require fewer splits to isolate, making them easily detectable through the algorithm’s path length mechanism.

Sparsity Robustness: The random feature selection in tree construction naturally handles sparse binary features, as the algorithm can focus on the subset of patterns that are actually present in each partition, avoiding the curse of dimensionality that affects distance-based methods.

3. Local Outlier Factor: Density Estimation Challenges: LOF’s moderate performance (F1-score: 0.66, AUPRC: 0.690) reflects the challenges of density-based anomaly detection in high-dimensional binary spaces:

Distance Metric Limitations: LOF relies on k-nearest neighbor distances, but in binary feature spaces, distance metrics become less discriminative due to the discrete nature of the data. Hamming distance, while appropriate for binary data, creates plateaus where many transactions have identical distances, reducing the algorithm’s ability to establish meaningful density gradients.

Curse of Dimensionality in Binary Space: High-dimensional binary spaces suffer from distance concentration, where all points become approximately equidistant. This phenomenon severely impacts LOF’s ability to identify local density variations, as the concept of ”local” becomes illdefined when distances lose their discriminative power.

Pattern Frequency Bias: LOF’s density estimation may be biased toward frequently occurring patterns, potentially misclassifying legitimate but rare pattern combinations as anomalies. This bias is particularly problematic in database transaction analysis, where certain valid but infrequent operational patterns should not be considered anomalous.

4. One-Class SVM: Kernel-Induced Feature Space Transformation: The superior performance of One-Class SVM (F1-score: 0.98, AUPRC: 0.95) demonstrates the power of kernel-based feature space transformation for binary pattern data:

RBF Kernel Compatibility: The RBF kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2)$ effectively transforms the discrete binary space into a continuous, high-dimensional feature space where linear separation becomes possible. This transformation pre-serves pattern relationships while enabling the construction of smooth decision boundaries that can capture complex pattern dependencies.

Pattern Similarity Modeling: The kernel function naturally captures pattern similarity through the exponential decay of the RBF, where transactions with similar pattern combinations receive higher similarity scores. This approach aligns well with the intuition that normal transactions should exhibit similar sequential patterns, while anomalous transactions deviate from these established patterns.

Margin Maximization in Pattern Space: OCSVM’s margin maximization principle creates robust decision boundaries that separate normal pattern combinations from potential anomalies with maximum confidence. The algorithm’s ability to handle the sparse, high-dimensional nature of binary features through kernel transformation makes it particularly well-suited for pattern-based anomaly detection.

5. Kernel Comparison: Linear vs. Non-linear Decision Boundaries: The performance differences between OCSVM variants reveal the importance of non-linear decision boundaries for pattern-based features:

Linear Kernel Limitations: Linear OCSVM (F1-score: 0.608) struggles because pattern combinations often exhibit non-linear relationships. Sequential patterns may be mutually exclusive, conditionally dependent, or exhibit complex logical relationships that cannot be captured by linear decision boundaries in the original binary space.

Polynomial Kernel Moderate Success: Polynomial kernels (F1-score: 0.48) capture some non-linear pattern relationships through polynomial feature combinations, but may suffer from overfitting to specific pattern combinations present in the training data, reducing generalization to novel but legitimate pattern variations.

RBF Kernel Superiority: The RBF kernel's ability to create smooth, localized decision boundaries proves optimal for capturing the complex but structured relationships between sequential patterns, enabling robust anomaly detection while maintaining generalization capability.

6. Implications for Feature Engineering and Model Selection: This analysis reveals several critical insights for anomaly detection in pattern-based feature spaces:

Algorithm-Feature Space Alignment: Model selection must consider the fundamental compatibility between algorithm assumptions and feature space characteristics. Algorithms designed for continuous data (Elliptic Envelope) perform poorly on binary features, while tree-based and kernel methods naturally accommodate discrete feature spaces.

Sparsity Handling Capability: High-dimensional sparse binary features require algorithms that can effectively handle the curse of dimensionality and distance concentration effects. Kernel methods and tree-based approaches demonstrate superior robustness to these challenges compared to distance-based and covariance-based methods.

Pattern Relationship Modeling: The success of OCSVM demonstrates the importance of capturing complex pattern relationships through appropriate kernel functions, while the failure of simpler approaches highlights the inadequacy of linear assumptions for sequential pattern analysis.

These insights extend beyond performance metrics to provide fundamental guidance for algorithm selection in pattern-based anomaly detection applications, emphasizing the critical importance of theoretical compatibility between feature representation and algorithm assumptions.

V. CONCLUSION

With One-Class SVM (OCSVM) acting as the main classifier, we presented a database intrusion detection system in this study that blends sequential pattern mining with machine learning models. Robust feature representation of database activity was made possible by converting transactions into binary feature vectors depending on whether or not mined sequential patterns were present. OCSVM with an RBF kernel performed better than other methods, according to experimental evaluation, obtaining improved accuracy, precision, recall, and F1-score while keeping a low false positive rate. In situations when only normal data is available for training, OCSVM regularly outperformed comparative models such as Isolation Forest, Local Outlier Factor, Elliptic Envelope, PNN, and C-SVM, despite providing helpful baselines. The results validate OCSVM's suitability for database system anomaly detection, especially for identifying complex or before unknown harmful transactions. The suggested method provides great generalization and resilience by learning compact decision boundaries around typical behavior and modeling dependencies at the transaction level. This work establishes three core theoretical contributions: Sequential Pattern-Binary Feature Integration Framework (optimal configuration: support ratio = 0.05, pattern length [2,4] generating 847 discriminative patterns), Algorithm-Feature Space Compatibility Principle (evidenced by Elliptic Envelope's failure versus kernel methods' success), and One-Class Learning Effectiveness achieving 98% F1-score and 95.15% AUPRC. Practical contributions include production-ready deployment guidelines, noise-robust methodology (20% transaction-level noise), and scalable feature engineering with computational tractability for real-time monitoring. The OCSVM-RBF and pattern-mining approach establishes generalizable principles: Sequential Context Preservation (85% discriminative information capture), Binary Pattern Encoding Superiority (15-26% memory reduction), and Kernel-Based Feature Space Transformation enabling smooth decision boundaries. These principles extend beyond database security to web applications, IoT monitoring, and financial systems, providing guidance for anomaly detection across diverse domains requiring discrete, sparse, high-dimensional features. This research establishes a paradigm shift from reactive signature-based to proactive pattern-based learning, addressing the fundamental limitation of existing ML-based DIDS approaches that fail to learn from normal data alone while capturing sequential patterns essential for detecting sophisticated attacks. The integration of sequential pattern mining with kernel-based one-class learning provides a generalizable framework for next-generation database protection systems capable of evolving with threat landscapes while maintaining operational efficiency and interpretability.

VI. FUTURE WORK

While the proposed framework demonstrates promising results, several directions warrant future investigation. The immediate priority involves testing the framework on real-world database audit log datasets from production environments, including

PostgreSQL, MySQL, and Oracle systems, to validate performance beyond synthetic data and assess scalability with enterprise-scale transaction volumes exceeding millions of operations daily. Integration with existing database management systems for live detection requires developing lightweight monitoring agents that can process transaction streams in real-time without impacting database performance, alongside establishing standardized APIs for seamless deployment across heterogeneous database environments. Deep learning architectures, particularly LSTM and GRU networks, should be explored for direct sequence modeling to eliminate the pattern mining preprocessing step, potentially capturing more complex temporal dependencies through end-to-end learning while comparing computational efficiency against the current PrefixSpan-based approach. Transformer architectures with attention mechanisms could model long-range dependencies in transaction sequences more effectively than current fixed-length pattern mining, while CNNs might identify local sequential patterns with reduced computational overhead compared to traditional mining algorithms. Adversarial training and game-theoretic defense strategies should be explored to improve robustness against adaptive attackers who may attempt to evade detection through carefully crafted transaction sequences that mimic normal behavior while achieving malicious objectives. Real-time deployment optimization through model compression techniques, quantization, and specialized hardware acceleration using GPUs or TPUs would enable sub-millisecond detection latency required for high-throughput database systems. While our anomaly-based IDS demonstrates high accuracy in detecting suspicious transaction behavior, ethical implications must be considered before real-world deployment. False positives may inadvertently disrupt legitimate user activity, especially in mission-critical database systems. It is crucial to implement human-in-the-loop verification mechanisms or escalation paths for flagged transactions to minimize unnecessary operational impact. Additionally, log-based analysis for feature extraction must be performed with strict adherence to data privacy regulations, ensuring that sensitive user information is anonymized or handled under secure audit protocols. Future work should explore privacy-preserving anomaly detection approaches to balance security enforcement with user rights. Finally, hybrid systems combining rule-based detection with machine learning, specialized modules for specific attack types such as SQL injection and privilege escalation, and standardized benchmark datasets derived from real database audit logs would accelerate research advancement and enable fair comparison across different database intrusion detection approaches.

REFERENCES

[1] Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: techniques, datasets and challenges,” Cybersecurity, vol. 2, no. 1, pp. 1–22, 2019. https://doi.org/10.1186/s42400-019-0038-7

[2] J. L. Leevy and T. M. Khoshgoftaar, “A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data,” Journal of Big Data, vol. 7, no. 1, p. 104, 2020. https://doi.org/10.1186/s40537-020-00382-x

[3] R. J. Santos, J. Bernardino, and M. Vieira, “Approaches and challenges in database intrusion detection,” ACM Sigmod Record, vol. 43, no. 3, pp. 36–47, 2014. https://doi.org/10.1145/2694428.2694435

[4] J. Breier and J. Branis˘ova´, “Anomaly detection from log files using data mining techniques,” in Information Science and Applications. Springer, 2015, pp. 449–457. https://doi.org/10.1007/978-3-662-46578-3_53

[5] A. Kundu, S. Sural, and A. K. Majumdar, “Database intrusion detection using sequence alignment,” International Journal of information security, vol. 9, no. 3, pp. 179–191, 2010. https://doi.org/10.1007/s10207-010-0102-5

[6] Y. Guo, “A review of machine learning-based zero-day attack detection: Challenges and future directions,” Computer Communications, vol. 198, pp. 175–185, 2023. https://doi.org/10.1016/j.comcom.2022.11.001

[7] T. Sowmya, “A comprehensive review of ai-based intrusion detection,” Journal of Cybersecurity and Information Systems, vol. XX, pp. XX– XX, 2023. https://doi.org/10.1016/j.measen.2023.100827

[8] M. R. Keyvanpour, M. B. Shirzad, and S. Mehmandoost, “Cid: A novel clustering-based database intrusion detection algorithm,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 1601– 1612, 2020. https://doi.org/10.1007/s12652-020-02231-4

[9] Singh, “Expectation maximization clustering and sequential pattern mining based approach for detecting intrusive transactions in databases,” Multimedia Tools and Applications, 2021. https://doi.org/10.1007/s11042-021-10786-3

[10] R. Jindal and I. Singh, “A survey on database intrusion detection: approaches, challenges and application,” International Journal of Intelligent Engineering Informatics, vol. 7, no. 6, pp. 559–592, 2019. https://doi.org/10.1504/IJIEI.2019.104565

[11] W. Zhang and J. P. Lazaro, “A survey on network security traffic analysis and anomaly detection techniques,” International Journal of Emerging Technologies and Advanced Applications, vol. 1, no. 4, pp. 8–16, 2024. https://doi.org/10.62677/IJETAA.2404117

[12] S. Kumar, S. Gupta, and S. Arora, “Research trends in network-based intrusion detection systems: A review,” IEEE Access, vol. 9, pp. 157 761–157 779, 2021. https://doi.org/10.1109/ACCESS.2021.3129775

[13] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, “A novel anomaly detection scheme based on principal component classifier,” 2003.

[14] S. Mukkamala, G. Janoski, and A. Sung, “Intrusion detection: support vector machines and neural networks,” in proceedings of the IEEE International Joint Conference on Neural Networks (ANNIE), St. Louis, MO, 2002, pp. 1702–1707.

[15] Y. Hu and B. Panda, “A data mining approach for database intrusion detection,” in Proceedings of the 2004 ACM symposium on Applied computing, 2004, pp. 711–716. https://doi.org/10.1145/967900.968048

[16] Y. Hu and B. Panda, “Mining inter-transaction data dependencies for database intrusion detection,” in Innovations and Advances in

Computer Sciences and Engineering, T. Sobh, Ed. Dordrecht: Springer Netherlands, 2010, pp. 67–72. https://doi.org/10.1007/978-90-481-3658-2_12

[17] Y. Hu and B. Panda, "Identification of malicious transactions in database systems," in Seventh International Database Engineering and Applications Symposium, 2003. Proceedings., 2003, pp. 329–335. https://doi.org/10.1109/IDEAS.2003.1214946

[18] M. Doroudian and H. R. Shahriari, "A hybrid approach for database intrusion detection at transaction and intertransaction levels," in 2014 6th Conference on Information and Knowledge Technology (IKT). IEEE, 2014, pp. 1–6. https://doi.org/10.1109/IKT.2014.7030322

[19] U. P. Rao, G. Sahani, and D. R. Patel, "Machine learning proposed approach for detecting database intrusions in rbac enabled databases," in 2010 second international conference on computing, communication and networking technologies. IEEE, 2010, pp. 1–4. https://doi.org/10.1109/ICCCNT.2010.5591574

[20] M. Kumar, M. Hanumanthappa, and T. S. Kumar, "Intrusion detection system using decision tree algorithm," in 2012 IEEE 14th international conference on communication technology. IEEE, 2012, pp. 629–634. https://doi.org/10.1109/ICCT.2012.6511281

[21] R. K. S. Gautam and E. A. Doegar, "An ensemble approach for intrusion detection system using machine learning algorithms," in 2018 8th International conference on cloud computing, data science & engineering (confluence). IEEE, 2018, pp. 14–15. https://doi.org/10.1109/CONFLUENCE.2018.8442693

[22] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," Information sciences, vol. 378, pp. 484–497, 2017. https://doi.org/10.1016/j.ins.2016.04.019

[23] M. Zhang, B. Xu, and J. Gong, "An anomaly detection model based on one-class svm to detect network intrusions," in 2015 11th International conference on mobile ad-hoc and sensor networks (MSN). IEEE, 2015, pp. 102–107. https://doi.org/10.1109/MSN.2015.40

[24] Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," Ieee Access, vol. 5, pp. 21 954–21 961, 2017. https://doi.org/10.1109/ACCESS.2017.2762418

[25] H. R. Sayegh, W. Dong, and A. M. Al-madani, "Enhanced intrusion detection with lstm-based model, feature selection, and smote for imbalanced data," Applied Sciences, vol. 14, no. 2, p. 479, 2024. https://doi.org/10.3390/app14020479

[26] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based ddos detection system in software-defined networking (sdn)," arXiv preprint arXiv:1611.07400, 2016. https://doi.org/10.4108/eai.28-12-2017.153515

[27] Sadaf and J. Sultana, "Intrusion detection based on autoencoder and isolation forest in fog computing," IEEE Access, vol. 8, pp. 167 059– 167 068, 2020. https://doi.org/10.1109/ACCESS.2020.3022855

[28] E. Caville, W. W. Lo, S. Layeghy, and M. Port-mann, "Anomal-e: A self-supervised network intrusion detection system based on graph neural networks," Knowledge-Based Systems, vol. 258, p. 110030, 2022. https://doi.org/10.1016/j.knosys.2022.110030

[29] H. Kamal and M. Mashaly, "Enhanced hybrid deep learning models-based anomaly detection method for two-stage binary and multi-class classification of attacks in intrusion detection systems," Algorithms, vol. 18, no. 2, p. 69, 2025. https://doi.org/10.3390/a18020069

[30] Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," in proceedings of the 17th international conference on data engineering. IEEE Piscataway, NJ, USA, 2001, pp. 215–224.

[31] Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in Proceedings of the ACM SIGKDD workshop on outlier detection and description, 2013, pp. 8–15. https://doi.org/10.1145/2500853.2500857

[32] Scho¨lkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Neural computation, vol. 13, no. 7, pp. 1443–1471, 2001. https://doi.org/10.1162/089976601750264965

[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in 2008 eighth ieee international conference on data mining. IEEE, 2008, pp. 413–422. https://doi.org/10.1109/ICDM.2008.17

[34] R. C. Ripan, I. H. Sarker, M. M. Anwar, M. H. Furhad, F. Rahat, M. Hoque, and M. Sarfraz, "An isolation forest learning based outlier detection approach for effectively classifying cyber anomalies," in Hybrid Intelligent Systems, A. Abraham, T. Hanne, O. Castillo, N. Gandhi, T. Nogueira Rios, and T.-P. Hong, Eds. Cham: Springer International Publishing, 2021, pp. 270–279. https://doi.org/10.1007/978-3-030-73050-5_27

[35] M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104. https://doi.org/10.1145/342009.335388

[36] J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," Technometrics, vol. 41, no. 3, pp. 212–223, 1999. https://doi.org/10.1080/00401706.1999.10485670

[37] F. Specht, "Probabilistic neural networks," Neural networks, vol. 3, no. 1, pp. 109–118, 1990. https://doi.org/10.1016/0893-6080(90)90049-Q

[38] H. A. Salman, A. Kalakech, and A. Steiti, "Random forest algorithm overview," Babylonian Journal of Machine Learning, vol. 2024, pp. 69– 79, 2024. https://doi.org/10.58496/BJML/2024/007

[39] C. Bente´jac, A. Cso¨rgo˝, and G. Mart´ınez-Mun˜oz, "A comparative analysis of gradient boosting algorithms," Artificial Intelligence Review, vol. 54, no. 3, pp. 1937–1967, 2021. https://doi.org/10.1007/s10462-020-09896-5

[40] R. Andonie, "Hyperparameter optimization in learning systems," Journal of Membrane Computing, vol. 1, no. 4, pp. 279–291, 2019. https://doi.org/10.1007/s41965-019-00023-0

[41] T. Joachims, "Making large-scale svm learning practical," Technical report, Tech. Rep., 1998.

Received on 05-11-2025 Accepted on 06-12-2025 Published on 23-12-2025

https://doi.org/10.65879/3070-5789.2025.01.10