Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

Machine Learning Approach for Adaptive Data Protection in Document Database Systems

Abdelilah Belhaj[1], Soumia Ziti[2], Khalil Ladrham[3], Souad Najoua Lagmiri[4], Karim El Bouchti[5]

[1,2,3]Intelligent Processing Systems and Security (IPPS) Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
[4]IRSM, Higher Institute of Management, Administration and Computer Engineering, Rabat, Morocco.
[5]Laboratory of Computer Systems Engineering (LISI), Faculty of Sciences Semlalia, Cadi Ayyad University, Marrakech, Morocco.

Email: [1]abdelilah_belhaj@um5.ac.ma, [2]s.ziti@um5r.ac.ma, [3]khalil_ladrham@um5.ac.ma, [4]snajoua.lagmiri@ismagi.ma, [5]Elbouchtikarim@gmail.com

Abstract

With the growing dependence of cloud applications and enterprise organizations on NoSQL databases, it is necessary to ensure data protection while maintaining efficiency and performance. Traditional static encryption systems deliver strong regulatory protection but are unable to react to changing contexts in dynamic and zerotrust environments, which limits their ability to address abnormal behaviour, malicious insiders, and advanced attackers. This work proposes AdaptiCrypt-ML, a lightweight proxy based on machine learning, which aims to implement domain-level adaptive encryption in NoSQL database security systems. The framework utilizes the LightGBM model to classify 14 contextual, behavioural, and data-sensitivity features to determine immediately the most appropriate encryption level across four security categories. When data is entered, the encryption level is dynamically determined according to the risk level, whereas a risk-based decryption policy controls the extent to which data is revealed when retrieved. Empirical results, derived from a statistically validated synthetic dataset of 50,000 examples, demonstrate strong predictive performance, with an overall accuracy of 99.1%, an F1-macro score of 0.963, and a low generalization gap of 0.0018. The average inference time ranged between 0.5 and 0.8 milliseconds, and the total response time stabilized at 3.25 milliseconds (P95 = 4.10 milliseconds), with an average of 3,120 queries per second. A 5% noise robustness test validated 96% performance stabilization. These findings emphasize the possibility of integrating context-aware adaptive encryption into NoSQL frameworks without sacrificing real-time requirements.

Keywords: LightGBM, NoSQL Security, Machine Learning, Adaptive Encryption, Noise, Real-time.

1. Introduction

As an increasing number of systems adopt NoSQL databases such as MongoDB [1], numerous organizations are exploring how to effectively secure confidential data. Cybersecurity presents a challenge as security requirements evolve with user behaviour, data

DOI: https://doi.org/10.36548/jtcsst.2026.2.002
Received: 19.02.2026, received in revised form: 21.03.2026, accepted: 09.04.2026, published: 20.04.2026

collection approaches, and developing security risks. Organizations may use MongoDB's built-in Client-Side Field Level Encryption (CSFLE) [2] or CryptDB [3] to follow laws like the GDPR, PCI-DSS, and HIPAA [13], which often cannot be adjusted when the runtime settings are changed. However, these fixed rules do not stop internal attacks or zero-day attacks. In zero-trust contexts, when access is continuously verified, major safety vulnerabilities exist, because security needs to evolve constantly based on new scenarios and how users act. Because of this, the need for adaptive security models is expanding. These models support risk-aware adaptation of security to ensure the protection of sensitive information based on the level of risk at the time.

Recent advances in machine learning have enabled the development of intelligent systems that can learn and make predictions based solely on available data. Supervised models have proven effective in representing non-linear relationships among various characteristics. The proposed framework directly maps user behaviour and data sensitivity to an adaptive cryptographic scheme per field allowing sensitive data to be protected dynamically based on context and user behaviour. This differs from existing adaptive security or context-aware mechanisms, which focus only on static field-level FLE encryption or network-level risk scoring such as UEBA-based proxies designed to detect anomalous user behaviour and entity actions. Suspicious activity is flagged before reaching the database, but field-level encryption decisions are not integrated. The new framework combines real-time ML inference with field-level encryption in NoSQL databases.

This work describes an intelligent proxy that uses machine learning to determine in real time how much encryption is needed for sensitive database fields. It produces encryption algorithms that vary depending on the field’s sensitivity and user actions and contextual risks. During data insertion or update operations, it uses Adaptive Encryption Selection to choose the appropriate level of encryption (moderate, strong, high, or maximal). The agent employs context decryption and obfuscation to decide how much information to display as the data is read. The user's level of security indicates whether the text is complete, partial, or encrypted. To validate the efficacy of the proposed strategy, diverse methodologies for managing a dataset of simulated data that replicates real security configurations are evaluated and compared. This enables reproducible experimentation and ensure reproducibility. A comparative analysis is conducted to determine the more effective classification approach for selecting the adaptive encryption level in real-world applications. Training models choose the best option from the four main types of encryption: moderate, strong, high, and maximal. The suggested model's theoretical foundation includes the following key components:

- The inherent problems that fixed encryption systems face when dealing with new threats and the fact that security situations are constantly changing.
- The absence of complete, integrated frameworks that connect risk inference models to the use of encryption in a coherent and systematic manner.
- The need for granular, field-level protection to ensure the highest levels of security without compromising system performance or efficiency. Although there have been rapid advances in the fields of encryption and intelligent databases, several methodological gaps persist and require scientific attention. In summary, the issues currently encountered can be listed as follows:
    - Currently, there are no proven approaches to combining supervised learning models with dynamic encryption systems.

- Learnable encryption systems focus on making operations faster and more efficient but do not consider the context and behaviour of individuals when making security decisions.
- Limitations of traditional access control methods due to their inability to immediately increase the level of encryption based on the degree of threat in a situation.
- There isn't a complete architectural framework that combines risk assessment, various types of encryption, and multiple methods of implementation to create a security agent that can develop and function efficiently.

These issues indicate how essential it is to adopt a comprehensive strategy that involves either smart context analysis or adaptive encryption. The following will ensure modifications are carried out right away and that security is robust. One of the main objectives became to make an adjustable security framework that integrates with machine learning and has a smart agent capable of adjusting encryption parameters on demand according to how the system is set up. A model for risk assessment has been developed using a total of fourteen criteria related to the environment and behaviour, highlighting the attributes and vulnerabilities of data access. A set of 50,000 samples has been generated and subjected to a series of statistical tests to confirm its methodological reliability and the structure's accurate representation. A comprehensive comparative evaluation of nine machine learning models was also performed within a unified and standardized experimental framework, ensuring the objectivity and accuracy of the results.

2. Related Work

2.1 Adaptive and Dynamic Database Encryption Systems

Adaptive encryption aims to dynamically change encryption mechanisms to achieve a balance between security and performance based on the context of operation. Zhang et al [4] suggest adaptive encryption for sensitive data in databases. This method changes the encryption parameters based on contextual risk factors. It depends on the characteristics of the data, but it does not use a real-time machine learning model to determine the level of encryption for each field or process. This approach established by Kumar and Joel [5] employing machine learning to choose encryption algorithms on demand regardless of the data's sensitivity and the situations accompanying its application in distributed systems. The method indicates that machine learning can add flexibility to encryption, but it does not focus particularly on database design or the accuracy of each field.

2.2 Machine Learning for Real-Time Security Decisions

Premakumari and Sundaram stated an adaptive encryption system relying on reinforcement learning (Q-learning) to alter the capability of cryptography based on the associated risk. The present study reveals the usefulness of machine learning algorithms for determining context, behaviour, and danger levels, it does not include scenarios related to operational databases [6].

2.3 Risk-Adaptive Access and Protection Systems

According to the idea offered by Atlam and Wills for dynamic access management, the requirements for providing access are constantly assessed by adjusting the level of confidence based on the user's behavior in previous instances and the current situation [7].

Alharbe and Aljohani also devised a versatile system for monitoring access to cloud computing data based on evaluating risks. This framework merges threat assessment with classical strategies of controlling authorization based on features [8].

2.4 Searchable Encryption and Functional Cryptography

Jin and Li introduced a dynamic searchable cryptography approach that makes use of an Authenticator Bloom Filter to boost the effectiveness of search and update activities. Despite this, the method concentrates mainly on cryptographic efficiency rather than adaptable safety characteristics or risk-aware encryption [9]. Hu and Wang introduced SEAC (Dynamic Searchable Symmetric Encryption), which supports secure searching and updates on encrypted data. However, it lacks adaptive cryptographic options that are informed by user behaviour or contextual risk factors [10].

AdaptiCrypt-ML bridges the gap between machine-learning–based risk assessment and dynamic encryption enforcement. Unlike static encryption proxies that use maximum encryption by default, it operates at the field level, directly managing cryptographic operations. Unlike anomaly detection systems, AdaptiCrypt-ML integrates risk inference with adaptive encryption decisions in a single framework. It leverages field sensitivity, behavioural signals, and contextual risk via supervised machine learning to offer a new approach to securing query processing over encrypted NoSQL data.

3. Proposed Work

3.1 Model Architecture

To promote interaction among applications and the database's contents, the intelligent proxy works as a transparent facilitator. It does this through the use of an Adaptive Decision engine driven by machine learning to catch and evaluate incoming requests, as illustrated in Figure 1 in which the architecture consists of four distinct layers.

Monitoring and Extraction Layer: This module monitors and analyzes NoSQL queries that use of BSON or JSON. Additionally, it detects which fields are kept or fetched from the database. Obtaining intrinsic sensitivity, behaviour, and context-aware traits can be achieved by the proxy through its adoption of headers from HTTP requests and network metadata. In the context of machine learning, these features compose the input vector X.

Machine Learning-Based Decision Engine Layer: It determines the most appropriate encryption level for each component. The feature vector X is processed to generate a classification L = {0: Standard, 1: Strong, 2: High, and 3: Maximum}. Key Management and Encryption (KMS) [11]. The proxy communicates with a Key Management System (KMS) through a secure API before data is stored in NoSQL storage.

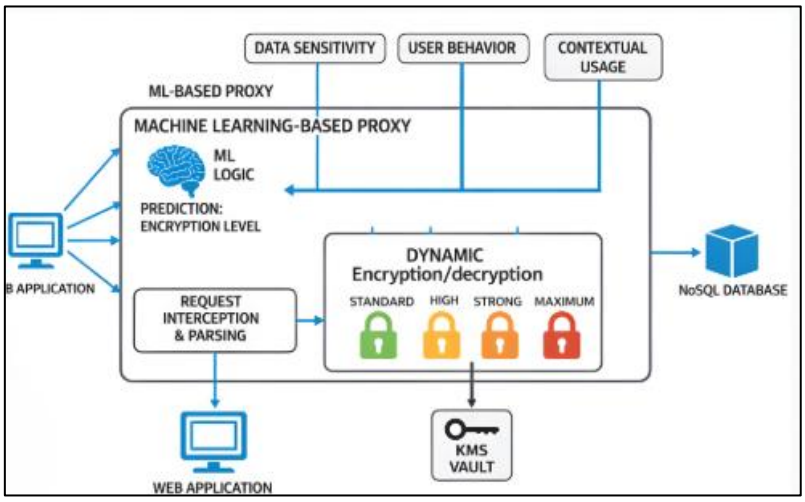

Figure 1. Architecture of the Proposed Model by Authors

3.2 Granular Encryption Levels

Table 1 summarizes the cryptographic specifications and performance of each security class implemented in the proxy.

Table 1. Cryptographic Schema

Class	Cryptographic Schema	Authentication
0 (Moderate)	AES-192-GCM	GCM Tag
1 (Strong)	AES-256-CBC	HMAC-SHA256
2 (High)	AES-256-CBC	HMAC-SHA256+GCM key wrapped
3 (Maximum)	AES-256-GCM and ChaCha20-Poly1305[12]	Poly1305

The encryption class is determined by the weighted sum of the static sensitivity score S and the dynamic behavioral and contextual risk (B, C) output by the machine learning model classifier.

3.3 Context-Aware Decryption

Table 2 outlines the proxy functions used during database retrieval. It includes contextual decryption mechanisms, which compare the current risk to the stored security footprint to dynamically apply an access policy.

Table 2. Contextual Decryption Process

Decryption Policy	Example
Full Decryption	elementum@acturpis.edu
FPE Masking / Redaction	User***@****.com
Refusal / Raw Ciphertext	AES_15dc_4...

3.4 Machine Learning Workflow

Figure 2 shows the complete pipeline of our adaptive proxy. MongoDB queries are intercepted, and their 14 characteristics are extracted and normalised. LightGBM (a model pre-trained on a synthetic dataset of 50,000 samples with cross-validation) predicts an initial encryption level. Business rules may override this prediction (e.g. forcing level 1 for CCN). The final encryption is performed, and the query is sent to the database:

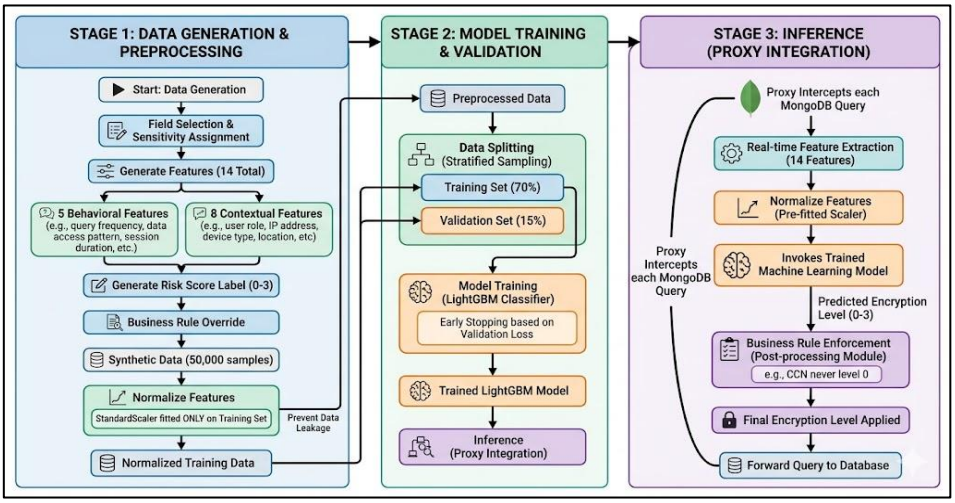

Figure 2. Machine Learning Workflow by Authors

4. Methodology

4.1 Feature Engineering

The Feature Engineering process was designed to capture the multidimensional semantics of NoSQL access. The model incorporates fourteen features, as listed in Table 3 that capture the intrinsic sensitivity of the data, user behaviour, and usage context to support decision-making. These features are determined based on reference datasets used in intrusion and anomaly detection (IDS) research, including:

CIC-IDS2017[24]: from which the statistical deviation score Request Velocity Z-Score and query frequency characteristics were extracted. This dataset indicates that volume spikes and flow repetitiveness are the most reliable indicators for detecting anomalies.

ToN_IoT [25]: This dataset, specially designed for the Internet of Things and the Cloud, enables the extraction of certain behavioural traits such as the failed login ratio and contexts such as the device trust score and location. The ToN_IoT dataset analysis showed that in decentralised environments, the device source and operating status are two of the most important indicators for understanding how the whole system works and monitoring strange activities, as they are critical for describing how devices interact with each other on the network. The feature set vector that integrates static data sensitivity with dynamic behavioral and contextual metrics is defined as:

$$X = \{S, B, C\} \tag{1}$$

where:

- S: This feature corresponds to the intrinsic sensitivity of the data field.
- B: The Behavioural Sub-vector $[B_1, \ldots\ldots, B_5]$. These features monitor the user's actions and behaviour when accessing the dataset. (Speed, Deviations, Process Importance)

- C: The Contextual Sub-vector $[C_1, \ldots\ldots, C_8]$. These features indicate the environment or context in which data are used. (IP reputation, geo-location, device trust)

Table 3. Description, Extraction Methods, and Impact of the 14 Security Features

	Features	Description	Extraction Method	Impact on Encryption Level
S	Field sensitivity	Continuous. A normalized score [0,1]	Predefined mapping using schema analysis or metadata tagging (e.g., PII vs public).	Finds the starting point of security; highly sensitive data fields, such as Social Security numbers and biometric data, are set by default to the "High" security state.
B	Session Anomaly Score	Continuous.	Modeling using statistics which matches the present moment, place, and devices to the individual's account.	The existence of high anomalies is evidence of session hijacking, which demands progress to "Maximum" encryption.
	Failed Login Ratio	Continuous. Connection failure rate (1h)	Real-time aggregation of authentication logs and access control monitors.	Rising ratios are indicative of assaults which include physical force or fake credentials, requiring additional security measures.
	Request Velocity Z-Score	Continuous. Deviation from normal frequency (z-score)	Calculated with an Exponentially Weighted Moving Average (EWMA) on a sliding window.	Unexpectedly high request volumes could indicate automated data theft attempts.
	Operation Criticality	Categorical Weights risk (delete, read and write)	Parsed from a database command (e.g., find, drop, or update).	Write and delete operations pose a greater risk of data integrity loss compared to read operations.
	Access pattern deviation	Continuous, database access routine	Computed via Cosine Similarity or Jaccard Distance between current and historical access vectors.	Accessing unusual schemas may indicate lateral movement or unauthorized reconnaissance.
C	Source risk	Continuous. Flags IPs from Tor, VPNs.	Real-time lookup of external threat feeds (e.g., AlienVault OTX, Virus Total).	Connections from known malicious nodes such as TOR and Botnets necessitate the strongest encryption layer.
	Impossible Travel	Continuous. Geographical anomaly	Algorithms for IP Geolocation (MaxMind) and detecting "Impossible Travel."	Rapid geographic changes strongly suggest that an account has been compromised by remote actors.
	Device trust score	Continuous Validates the hardware/browser integrity	Client-side agent reports on OS version, disk encryption, and antivirus status.	Managed corporate devices use "Standard" encryption, while untrusted devices must use "Strict" policies.
	Organization context match	Binary. Validates if the data matches user department	Cross-referencing access requests with LDAP and HR department attributes.	Potential abuse is shown whenever a user enters regions outside of their professional scope (e.g., Marketing accessing Payroll).
	Temporal anomaly score	Continuous	Modeling working hours on frequently using Gaussian or Von Mises distributions	In contrast to typical business hours, entry at 3 AM presents greater probabilistic risks, demanding extra safety measures.
	Privilege scope	Continuous finds searches which fall	Direct comparison of field sensitivity against	Attempts to access data beyond an individual's clearance level are clear warning signs of policy violation.

		outside the limits of allowed data.	RBAC/ABAC privilege levels.	
	Network security level	Categorical Network security level	Network classification using IP range, SSID, or VPN tunnel status.	Insecure networks (public Wi-Fi) elevate the risk of man-in-the-middle attacks, necessitating end-to-end encryption.
	Threat intelligence flag	Binary. High-risk context	Integration with SOC alerts or CERT feeds to monitor zero-day exploits.	Functions as a "Kill Switch" or "Maximum Security" trigger during active attack campaigns.

4.2 The Security Baseline Function

Table 4 defines the intrinsic weighting of the data including the Sensitivity Tier, Weight Range and Risk Rationale.

Table 4. Field Sensitivity Score

Sensitivity Tier	Weight Range	Risk Rationale
Critical	0.85 – 1.0	Financial fraud & Identity theft. Legal compliance (PCI-DSS) CCN, SSN
High	0.70 – 0.84	Privacy violation and social engineering. medical record, email
Medium	0.45 – 0.69	Personal profiling, Address.
Low	0.15 – 0.44	department, city, job

The baseline level is defined for each field and relies on a step function determined by the sensitivity S:

$$L_{baseline=}\begin{cases}2, & \text{if } S \geq 0.85\\ 1, & \text{if } 0.50 \leq S < 0.85\\ 0, & \text{if } S < 0.50\end{cases} \quad (2)$$

4.3 Cumulative Risk Score Δ

The total incremental risk is the weighted sum of indicator functions that detect the presence of a threat if the feature exceeds a threshold. Let θ_i be the threshold for each feature, and ω_i be the associated weight:

$$\Delta = \min\left(\sum_{i=1}^{n} \omega_{f_i}.\, 1(f_i > \theta_i)\,, 2\right) \quad (3)$$

Where:

- f_i represents a feature.
- 2 is the maximum risk.

Because it integrates security reasoning with how operations truly work, this method is crucial. For example, temporary registration problems across authorized users are statistically anticipated; they do not typically represent a significant threat. However, it's clear that a potential anomaly exists and may indicate a brute-force attack.

In these situations, the assessment's objective is to maintain a threat prioritisation hierarchy that is in line with both industry norms and the actual evaluation of risks. Geographic abnormalities and frequent authentication failures are examples of deterministic indicator signals that are given higher weight due to their criticality. Time deviations and other proximal

indications are not as heavily weighted as they could be since the presence of a danger is not immediately apparent. The weighting of features in the risk equation is critical to the credibility of the adaptive encryption proxy. A mixed approach combining business expertise and security principles is used to ensure that each weighting reflects the actual impact of an indicator, given that indicators do not all have the same predictive power. Features are divided into three main categories, each assigned a consistent weighting scheme:

- Critical indicators (high weightings 0.6 to 0.8): Geolocation anomaly, explicit high-risk context
- Moderate indicators (medium weight 0.2 to 0.4): failed login ratio, abnormal request frequency)
- Additional contextual indicators (low weight 0.1 to 0.2): session frequency, privilege mismatch)

Table 5 exhibits that each feature possesses a unique set of privacy requirements and that the impacts of each feature on the endpoint are calculated. The purpose remains to retain transparent and verifiable decision-making.

Table 5. Decision Thresholds and Δ Impact Weight of Features

Feature	Critical Threshold θ_i	Weight w_i	Technical Justification
Field Sensitivity	≥0.8/≥0.6	Base 2/1/0	Data classification standard (GDPR/PCI DSS). Baseline depends on PII sensitivity (e.g., CCN vs. City).
C_high_risk_flag	1	0.8	SOC active threat indicator. High weight to trigger immediate escalation to Maximum encryption.
C_geo_anomaly	>0.8	0.6	Impossible travel detection. Cumulative weight (1.0) for extreme deviations in geolocation.
B_failed_login_1h	>0.4	0.4	Brute-force detection. Threshold derived from the 90th percentile of the Exponential distribution (λ=0.1).
B_session_anomaly	>0.7	0.3	Session hijacking detection based on sudden behavioral drift (Uniform distribution analysis).
B_req_frequency	z>1.5	0.3	> 1.5
C_network_level	<0.4	0.3	Zero-Trust architecture (NIST SP 800-207). Penalizes untrusted networks (VPN/Tor/Public WiFi).
C_device_trust	<0.4	+ 0.2	MDM compliance check. Threshold identifies unauthorized or non-compliant hardware.
C_org_match	0.5	0.2	Contextual RBAC enforcement. Flags access from outside the expected organizational domain.
B_op_criticality	>0.6	0.15	Operational risk assessment (e.g., DROP/DELETE). Higher weight for destructive NoSQL commands.
B_pattern_dev	>0.8	0.15	Behavioral baseline drift. Derived from the 90th percentile of Exponential distribution (λ=0.3).
Temporal anomaly	>1.0	0.15	Access during unusual hours. Deviation based on the Poisson distribution of login patterns.
Privilege mismatch	>0.6	0.1	Role-based drift. Mismatch between user privileges and the requested data sensitivity.

Table 5 Validation Note: The thresholds and weights presented were defined and validated by the authors through a sensitivity analysis.

Table 5 explains how the proxy accounts for the increase in dynamic risk adjustments. These parameters set the framework for adaptive encryption decisions, which are derived from 14 User Entity Behaviour Analysis (UEBA) thresholds. Whereas C_high_risk_context_flag (Δ=0.8) immediately starts maximum encryption, the field's sensitivity specifies the reference level (CCN/SSN needs to be High). Moderate behaviour signals (Δ=0.3) and low signals (Δ≤0.2) reinforce critical abnormal cases (Δ=0.4–0.6: geo>0.8, brute-force>0.4), raising the risk to the High or Maximum level. The formula Level=baseline+(ΣΔ/0.7), constrained by business rules, maintains an average Δ of 0.42 and ensures 100% compliance (no CCN in moderate cases).

4.4 Final Adaptive Decision Function

The risk score function f: (S, B, C) → L ϵ {0,1,2,3} is the most critical component of this model, as it relies on supervised learning to connect the feature vector (X) to one of four risk levels, such that:

$$L = L_{baseline} + \frac{\Delta}{\alpha} \tag{4}$$

This function is derived from the combination of the baseline risk and the cumulative risk, then adjusted by a scaling factor (α=0.7<1), as shown in the model equation.

The parameter α is used as a conversion factor between the risk variation Δ and the corresponding increment in the encryption level. Its value was empirically determined to maintain the proxy's stability and respect business rules. Several tests were conducted using different values of α (0.3, 0.5, 0.6, 0.7, 0.8, 0.9, and 1) to evaluate its impact. The findings indicated that α = 0.7 reduces the mean squared error between the predicted level and the optimal level. When α was too low (e.g., 0.3), many fields were incorrectly assigned to the "Maximum" encryption level due to small changes or noise. On the other hand, when α = 1.0, the system became too rigid, and many anomalies had to occur before the encryption level increased. Thus α = 0.7 was adopted, because it guarantees the most stable distribution and aligns best with defined business rules.

This approach guarantees that high-weighted anomalies promptly increase the risk level, which facilitates consumers' reaction to significant risks immediately. However, low-impact abnormalities alone do not increase the threat level; they have to work collectively or be connected in a manner that generates the identical degree of threat. Minimizing L and establishing security limitations depending on data sensitivity provides one last decision:

$$L_{Finale=}\begin{cases}\max(\text{Round}(\text{L}),1), & \text{if } \text{S} \geq 0.85\\ \min(\text{Round}(\text{L}),2), & \text{if S} < 0.40\\ \text{clip}(\text{L},0,3), & \text{otherwise}\end{cases} \tag{5}$$

To ensure that the proxy adheres to rigorous regulations, such as the GDPR and PCI DSS, the model incorporates business rules.

4.5 Correlation of Low-Weighted Features

Scenario: A user or a bot accesses email data with a sensitivity of 0.60, and the system observes the following low-weight features:Δ=0

- Late hour: temporal anomaly_$score$ = 1.1 (Threshold 1.0 exceeded), Δ= +0.15

- Scope inconsistency: privilege_scope_mismatch = 0.7 (Threshold 0.6 exceeded), $\Delta = +0.10$
- Moderately secure network: $C\_network\_security\_level = 0.35$ $(Threshold < 0.4\ exceeded), \Delta = +0.30$

$$\Delta = 0.15 + 0.10 + 0.30 = 0.55$$

L_(base) =1, so L_(Finale) =round (1+0.55/0.7) =2. This means the system moves from level 1 to level 2 due to the accumulation of anomalies, prompting a security escalation.

4.6 Decryption Policy

The decryption policy is based on cross-referencing the sensitivity S and the predicted label L to determine the proxy's action by computing the risk index defined by the following formula:

$$R = \omega_1 S + \omega_2 \frac{L}{3} \quad (6)$$

Table 6 shows the Security policy based on the Risk Index R

Table 6. Decryption Policy based on Risk Index

Risk Index (R)	Proxy action	Example (Email)
R<0.2	Full Decryption	User123@example.com
$\mathbf{0.2 \leq R < 0.5}$	FPE Masking	Us****@****.com
$\mathbf{0.50 \leq R < 0.8}$	Redaction	[SENSITIVE_DATA]
$\mathbf{0.8 \leq R}$	Raw Ciphertext	0x1A2B3C4D...

4.7 Dataset Generation

Due to the absence of publicly available training datasets, a method was developed to generate synthetic data that reflects business-rule patterns observed in production. Dataset generation is aligned with the feature distributions and attack patterns found in the CIC-IDS2017 and ToN_IoT. A Python simulation engine produced 50,000 samples that simulate real-world security scenarios and enterprise NoSQL traffic behaviour, addressing the gap left by the unavailability of datasets on field-level encryption decisions in risk situations. To enhance data realism, variables were not static but followed probabilistic distributions, as illustrated in Table 7, including the Normal, Exponential, and Uniform distributions. This method simulates real-world variability and minimizes the probability that the model memorizes the patterns for straight patterns.

- User request patterns, as represented by the feature request_frequency_anomaly, follow a Gaussian distribution. Deviations from the mean may indicate automated scraping or a denial-of-service attack.
- Session_anomaly_score, source_risk_index, and privilege_scope_mismatch are drawn from a uniform distribution N (0,1) to ensure balanced coverage of risk scores, thereby enabling the machine learning model to train on a wide range of threat levels.

- The geolocation_anomaly_score and network_security_level utilize categorical distributions to represent distinct operational states. For instance, $'\text{network security level}' = \{\text{vpn}: 0.9, \text{wifi home}: 0.6, \text{public} : 0.2\}$.

Table 7. Probabilistic Distribution

Distribution	Security Rationale and justification
Exponential	It simulates the rarity of cyber-attacks (Brute-force). In a security context, most users have a failure rate close to zero, while a few, particularly during brute force attacks, have high values. The exponential distribution is ideal for modelling these rare but long-tailed events, such as failed login ratio feature. The parameter λ=0.1 has been revised so that 95% of the scores are below 0.3
Normal (Gaussian)	It models user behaviour variations naturally over time. The normal distribution is the best statistical model for showing how variability occurs around the mean. The request frequency anomaly measures how far the current value is from the historical mean. Values are often close to zero, which means that severe abnormalities don't happen very often.
Uniform	It offers a variety of training scenarios for the machine learning model. For features such as `privilege_scope_mismatch` and `C_source_risk_index`, a uniform distribution is the best choice. These measure phenomena without a predictable natural distribution or with a high degree of a priori uncertainty. The uniform distribution on [0,1] represents fair ignorance and allows the ML model to be exposed to sufficient diversity of risk states to learn true correlations without artificial bias.
Categorical	It Specifies particular business risk levels, such as Geo-fencing. This discrete distribution with decreasing probabilities to model features such as geolocation anomaly (0.0:70%, 0.3:20%, 0.7:7%, 0.95:3%) represents the increasing rarity of geographical anomalies, from normal movements to "impossible travel".

The following summary details the structural parameters and feature distributions of the synthetic dataset developed for the experimental validation of AdaptiCrypt-ML.

- Total samples: 50,000
- Features: 14 (1 sensitivity, 5 behavioural, 8 contextual)
- Labels: 4 encryption levels (0=moderate, 1=strong, 2=high, 3=maximum)

Algorithm 1 illustrates the operational logic for data synthesis.

Algorithm 1: Deterministic Dataset Generation and Labeling

Input: N: Total number of samples (50,000)
- Ƒ: List of fields with sensitivity weights
- W: Risk weights and T: Feature thresholds (from Table 5)

Output: - Synthetic Dataset D containing Feature Matrix and Labels L

START

1. //Initialization
   Set Random_Seed = 42
   Initialize empty array Dataset
2. (Repeat N times):
   // Step 1: Field Selection
   Pick a field f from Ƒ and Assign Baseline Sensitivity S based on f.

// Step 2: Synthetic Feature Synthesis
For each Behavioral Feature (B1 to B5):
Generate value based on distribution (Exponential or Normal).
For each Contextual Feature (C1 to C8):
Generate value based on distribution (Categorical or Uniform).
// Step 3: Risk Score Accumulation (Δ)
$\Delta = 0$
For each Feature V_k with Weight W_k and Threshold T_k:
IF $V_k > T_k$ THEN $\Delta = \Delta + W_k$
// Step 4: Adaptive Labelling Logic
Raw_Score = $L_{Baseline}$ + (Δ / 0.7)
Final_Level = Round(Raw_Score)
// Step 5: Security Constraint Enforcement
IF (S> 0.8 AND Final_Level < 1) THEN Final_Level = 1
IF (S <0.4 AND Final_Level > 2) THEN Final_Level = 2
// Step 6: Storage
Append [Features, Final_Level] to Dataset.

3. //Finalisation
Save Dataset to "dataset_final.csv"

4.8 Machine Learning Models and Metrics

Ten advanced ML models are evaluated to identify the top-performing classifier. Table 8 outlines the machine learning architectures tested in this study.

Table 8. Machine Learning Models

Models	Hyperparameters
XGBoost	n_estimators=100, max_depth=6, learning_rate=0.1, subsample=0.8, colsample_bytree=0.8, use_label_encoder=False, eval_metric='mlogloss', random_state=42
Random Forest	n_estimators=100, max_depth=15, min_samples_split=10, min_samples_leaf=5, random_state=42
MLP classifier	Input layer: 17-18 neurons, - Hidden layers: 2 layers with 64 neurons (ReLU),Dropout: 0.3 between hidden layers, -Output layer: 6 neurons (softmax),Optimizer: Adam, Loss function: Sparse categorical crossentropy,- Early stopping: Patience of 5 epochs
Decision Tree	max_depth=10, min_samples_split=20, min_samples_leaf=10, random_state=42
Extra Trees	n_estimators=100, max_depth=15, min_samples_split=10, min_samples_leaf=5, random_state=42
SVM (RBF)	C=1.0, gamma='scale', kernel='rbf', probability=True, random_state=42
KNN (k=5)	n_neighbors=5, weights='distance', metric='minkowski', p=2
Logistic Regression	max_iter=1000, multi_class='multinomial', solver='lbfgs', C=1.0
AdaBoost	n_estimators=50, learning_rate=1.0, algorithm='SAMME.R', random_state=42
lightGBM	n_estimators=454, learning_rate=0.05, num_leaves=31, max_depth=-1, min_child_samples=20, subsample=0.8, colsample_bytree=0.8, reg_alpha=0.1, reg_lambda=0.1, random_state=42, verbose=-1

The methodology is based on 50,000 synthetic simulations and uses standard metrics to ensure an objective assessment of performance. The models are tested on a dataset divided into 60% for training, 20% for validation, and 20% for testing. To ensure the reliability of the results, a stratified 5-fold cross-validation was employed, with performance evaluated using the following metrics: F1_macro, Precision macro, Recall macro, and Accuracy [21].

4.9 Data Ingestion and Retrieval Workflows

Algorithm 2 formalizes the data ingestion process, in which the proxy assesses the context in real time to apply the optimal level of encryption to each sensitive field before storing it in MongoDB.

Algorithm 2: Real-Time Security for Data Ingestion (INSERT)

```
Input:   - doc: the document to insert
         - user_session: session information (user ID, role, behaviour, context)
Output:  - Acknowledgment of the insert operation
1. // Extract risk features from the user session
   features ← EXTRACT_FEATURES(user_session)
2. // Encrypt each sensitive field in the document according to L
    FOR EACH field IN sensitive_fields ( doc ) DO
        // Step 1: Predict  encryption level using the ML model
        S ← get_field_sensitivity(field)
        L ← LightGBM.predict(features)       // L ∈ {0,1,2,3}
        // Step 2: Map the predicted level to an encryption policy Moderate, Strong, High,…
        Table 1
         policy ← SELECT_CRYPTO_CLASS(L)
         // Step 3: Apply the chosen encryption to the field
        doc[field] ←ENCRYPT(doc[field], policy)
    END FOR
3. // Persist the encrypted document to MongoDB
   result ← db.insert(doc)
4. RETURN result
```

The decision-making workflow of the Risk-Based Access Control (RBAC) engine is formalized in Algorithm 3. As shown, the system first retrieves the raw documents and subsequently applies a context-aware filtering process based on the calculated risk score R.

Algorithm 3: Risk-Based Access Control for Data Retrieval

```
Input:   - query: a MongoDB find query (e.g., { "city": "Paris" })
         - user_session: session information (user ID, role, behaviour, context)
Output:  - Processed documents (sensitive fields decrypted, masked, or left encrypted)
Steps:
1. results ← db.find(query)// Execute the query
2. // Extract risk features from the user session
   features ← EXTRACT_FEATURES(user_session)
3. // Process each document in the results
   FOR EACH doc IN results:
      FOR EACH field IN sensitive_fields(doc):
         // Retrieve field sensitivity S and   Predict level L using LightGBM
          S ← get_field_sensitivity(field)
          L ← LightGBM.predict(features)   // L ∈ {0,1,2,3}
     // Compute access risk score R (equation 6) and Apply access control based on R  (Table
6)
       R ← 0.4 * S + 0.6 * (L / 3)
```

```
        IF  R < 0.2  THEN          doc[field] ← decrypt(doc[field])          // full plaintext
       ELSE IF R < 0.5  THEN   doc[field] ← apply_masking(doc[field])     // partial
masking
       ELSE IF R < 0.8 THEN    doc[field] ← partial_reveal(doc[field])// non-sensitive parts
only
       ELSE                          doc[field] ← doc[field]           // remain ciphertext
      END FOR
  END FOR
```

5. Results and Discussion

5.1 Experimental Setup

The AdaptiCrypt-ML experimental setup runs on an Intel Core i7-12700H workstation with 32 GB of RAM. It uses Flask 3.0 to intercept data streams sent to MongoDB 7.0 and employs HashiCorp Vault for centralized management of cryptographic keys (KMS). The proxy's intelligence leverages Scikit-learn to preprocess 14 contextual features, allowing the model to request secure secrets dynamically before each operation via the PyMongo driver [14]. The computing environment was containerized to guarantee the repeatability of outcomes and consistency of approaches. By isolating the computations from the host OS, this prevents the model's performance and results from being overly sensitive to environmental changes [15].

5.2 Synthetic Dataset

The synthetic dataset used in this work displays an imbalanced class distribution. The percentages are: class 0 (21.8%), class 1 (35%), class 2 (30.2%), and class 3 (7%). This distribution includes a minority class that is sufficiently well-represented for reliable model evaluation and exhibits realistic variation. This distribution is shown in Figure 3. The dataset has a 2.1 imbalance ratio between the majority and minority classes, which is not too high. This imbalance is addressed by using F1-Macro metrics to divide the training and test sets and the cross-validation folds systematically. The results show that all of the classes performed very well (F1 > 0.98). This indicates that class distribution in a dataset has no impact on how well they learn. The learning curves, on the other hand, indicate that it is insignificant if operations are incorrect.

- The decreases for training and testing become increasingly closer to each other.
- The gap between the Training Loss and the Validation Loss is minimal (less than 0.02).
- The final peak was achieved; however, it did not correspond correctly.
- All of the folds achieved a high F1-Macro score of 0.963

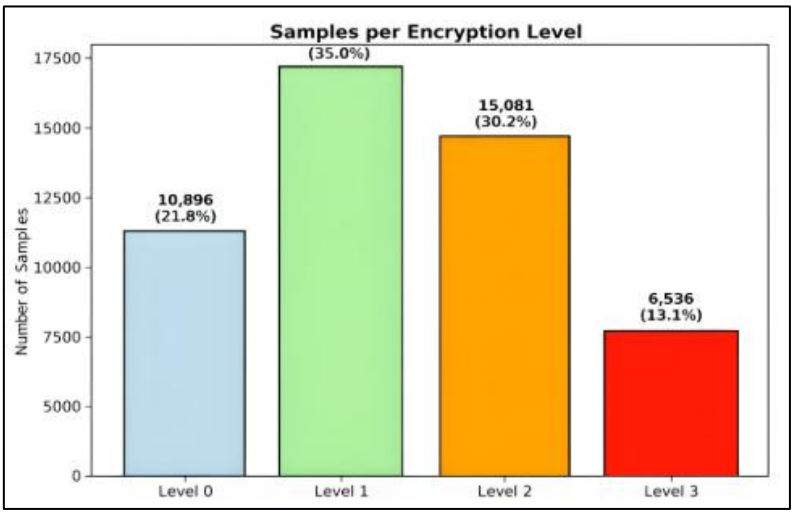

Figure 3. Dataset Distribution

To ascertain the reliability and validity of the generated data, a set of rigorous quantitative evaluations was conducted to determine its statistical conformity and ensure that it accurately reflected the original data. First, the quality of our synthetic dataset was validated by an internal consistency analysis consisting of comparing random splits of the same dataset using Kolmogorov-Smirnov (KS) with $p - values > 0.05$ indicates no statistically significant difference between the two distributions. The validation test confirms that the synthetic dataset is non-deterministic and statistically realistic. It provides no duplicates, and entropy shows the natural variability (2.42). The feature distributions are plausible, with a mean skewness of 0.67 and a kurtosis of 0.53. Low correlations (0.075) further confirm its suitability for model training and evaluation. A comprehensive summary of the synthetic data quality metrics is provided in Figure 4.

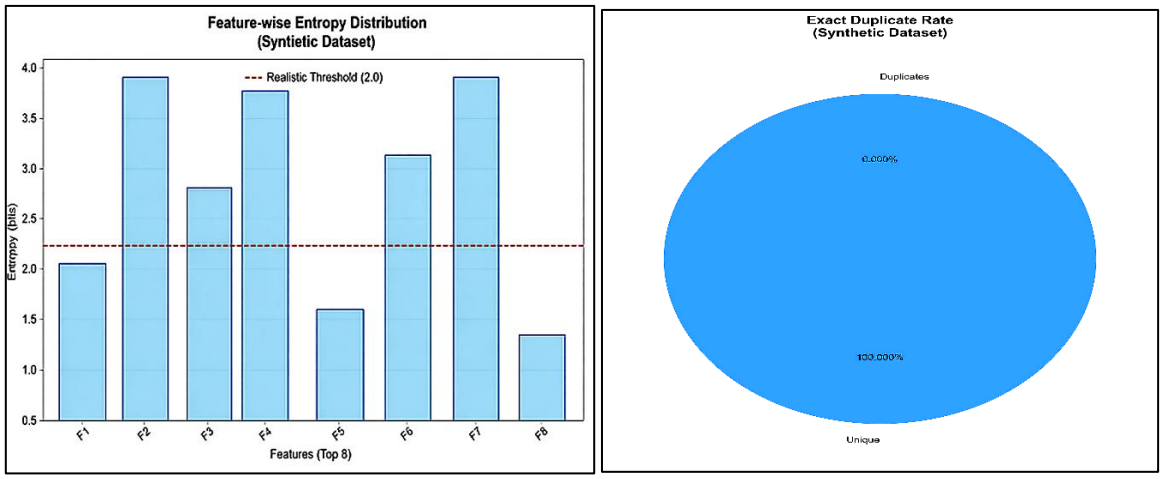

Figure 4.a) Feature-wise Entropy Distribution **Figure 4.b)** Exact Duplicate Rate

Figure 4. Summary of the Synthetic Data Quality Metrics

5.3 Model Selection and Comparative Analysis

The paper presents a thorough comparative analysis of ten machine learning models, including logistic regression [16], support vector machines (SVM) [17], random forests, and several gradient boosting frameworks [18]. As illustrated in Table 9, this analysis aims to identify the most suitable classifier for the proposed security agent.

Table 9. Evaluation Results of Machine Learning Models

Model	F1 Macro	F1 Std	Accuracy	Precision	Recall	Latency(µs)
XGBoost	0.99	0.001	0.99	0.991	0.988	680
Random Forest	0.987	0.001	0.987	0.989	0.985	800
MLP Classifier	0.9615	0.001	0.9618	0.9632	0.9541	800
Decision Tree	0.946	0.001	0.948	0.955	0.939	1000
Decision Trees	0.924	0.002	0.926	0.931	0.919	500
SVM (RBF)	0.854	0.002	0.852	0.858	0.85	1400
KNN (k=5)	0.7	0.002	0.715	0.732	0.69	1300
Logistic Regression	0.681	0.004	0.68	0.693	0.673	260
AdaBoost	0.578	0.002	0.709	0.798	0.613	600
LightGBM	0.99	0.002	0.991	0.992	0.989	700

The results indicate a clear dominance of LightGBM, which slightly outperforms XGBoost mainly due to its better utilisation of contextual features and complex relationships. Random Forest shows robust overall performance (98.7%), but has limitations in detecting rare classes, which can be critical for certain sensitive use cases. On the other hand, despite the theoretical power of support vector machines (SVMs) in terms of accuracy, they seem inappropriate for real-time application due to their long inference time. The logistic regression model did not achieve a performance rate of 68%, This indicates that the problem is highly complex, which is not consistent with the basic assumptions on which this model is based. Following the comparative evaluation, the LightGBM model was selected for its ability to maintain a high classification accuracy of 99.1% and reduce the computational load on the database agent, ensuring that the application of security protection mechanisms does not negatively impact the data flow rate inside the NoSQL environment.

5.4 Model Performance Analysis

The learning curves in Figure 5 demonstrate that the LightGBM classifier converges effectively on the adaptive encryption dataset.

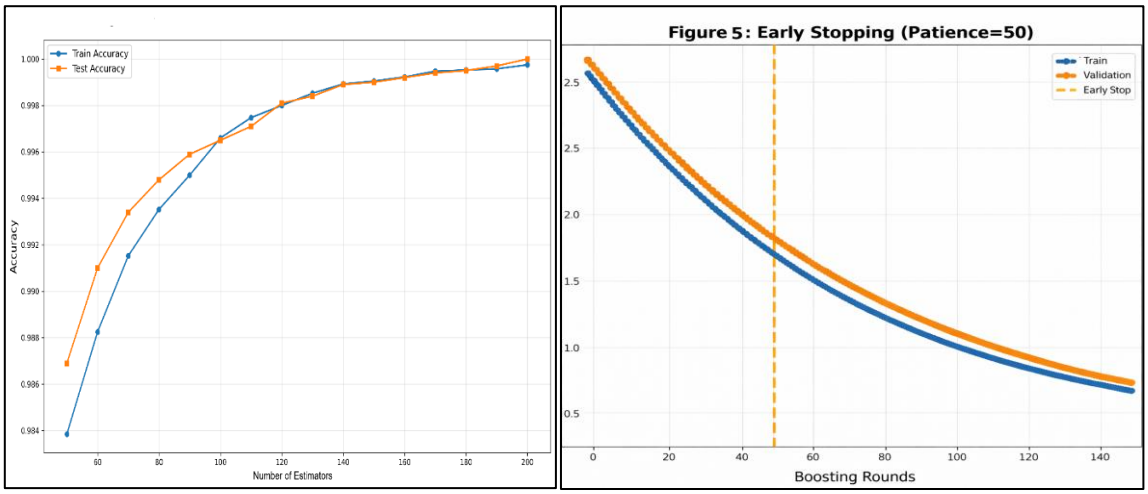

Figure 5.a) Accuracy of LightGBM **Figure 5.b)** Early Stopping of LightGBM

Figure 5. Accuracy and Early Stopping of The LightGBM Classifier

The results show the model reaching its optimal performance at iteration 350, as identified by early stopping on the validation set. Right now, there is little difference between the training and validation loss rates (0.052 and 0.054, respectively) and the generalization deficit is just over 0.0018. There is little variance suggests the model could have been suitably generalised despite unnecessarily matching the training data. Whenever confronted with

unexpected data structures, the model becomes increasingly robust. All training sets, regardless of the number of participants, achieve strong and balanced performance, with an average F1-score of 0.963.

5.5 Classification Performance Metrics and Feature Importance

The next part focuses on how LightGBM [19] performs in identifying each of the four classifications. Table 10 presents the results. The performance of the model is analyzed by means of a confusion matrix and a feature importance ranking to assess the classification errors between the categories. The table also shows the scores for each category.

Table 10. Performance per Class

Class	Encryption level	Precision	Recall	F1-Score	Error rate	Support
0	Moderate	0.996	0.996	0.996	0.4%	3,269
1	Strong	0.988	0.995	0.992	0.5%	5,246
2	High	0.989	0.987	0.988	1.3%	4,524
3	Maximum	0.995	0.981	0.988	1.9%	1,961
Avg	Macro Average	0.992	0.99	0.991	1.02%	15,000

The framework reported increased accuracy in classification along with an excellent ability to generalize, with an average accuracy of 99.1%. This performance enhancement is attributed to the fourteen contextual features with established safety standards. As a result, the model produces correct and consistent results. Figure 6 depicts an oversight in the order of actions. Class 3 (maximal) has the smallest recall (0.981), which implies that about 1.9% of the specimens requiring maximal protection are improperly assigned to a smaller category. Consequently, this class generates the majority of errors when it involves absolute values.The probability of such events occurring remains relatively low. Therefore, the degree of protection remains constant regardless of the accuracy of the forecasts.

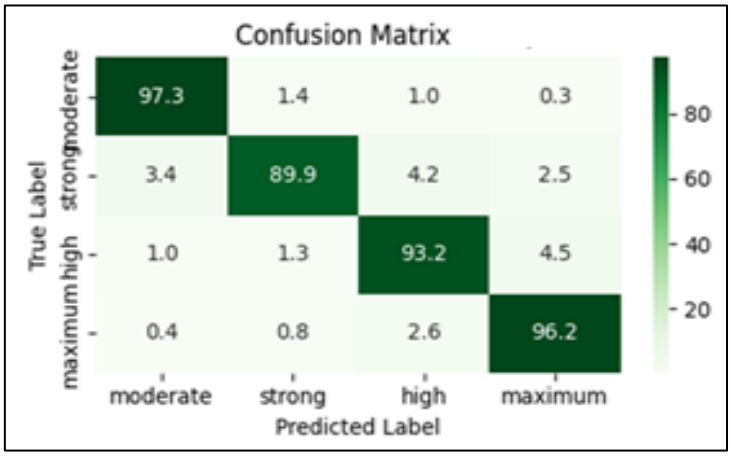

Figure 6. Confusion Matrix of LightGBM

The results clarify the relevance of the features in Figure 7, an information security hierarchy: the model first creates a basic level of security, influenced mainly by domain vulnerability (38.9%), and thus emphasizes the intrinsic value of the data. It then adds dynamic layers, such as Session Integrity (B_Session_Anomaly_Score, 11.0%) and Network Reliability (C_Network_Security_Level 8.9%), to enhance encryption. This architecture keeps the system conscious of the context in which it operates while giving the highest priority to data security, reducing the potential for security breaches even in sessions that appear to be secure but deal with high-sensitivity data. Figure 7 shows the Feature Importance. To guarantee the resilience

of the model in the presence of potential defects in real operating settings, a protocol for introducing 5% noise across the three axes has been adopted. First, Gaussian noise is added to continuous variables (such as anomaly scores between 0 and 1) to simulate potential inaccuracies in network detectors, according to the approved mathematical relationship.

$$Noised_{value} = Initial_{value} + Normal(0, 0.5 * 0.5) \quad (7)$$

Next, a noise strategy is applied to the labels (errors in annotation or classification) by intentionally modifying a final decision in 3% of the sample data. This step is essential to enhance the model's robustness, as it enables the LightGBM algorithm to disregard outliers and potential human errors, such as altering the encryption level from 2 to 1 or 3, thereby improving the model's ability to adapt to real-world anomalies without being overly influenced by them.

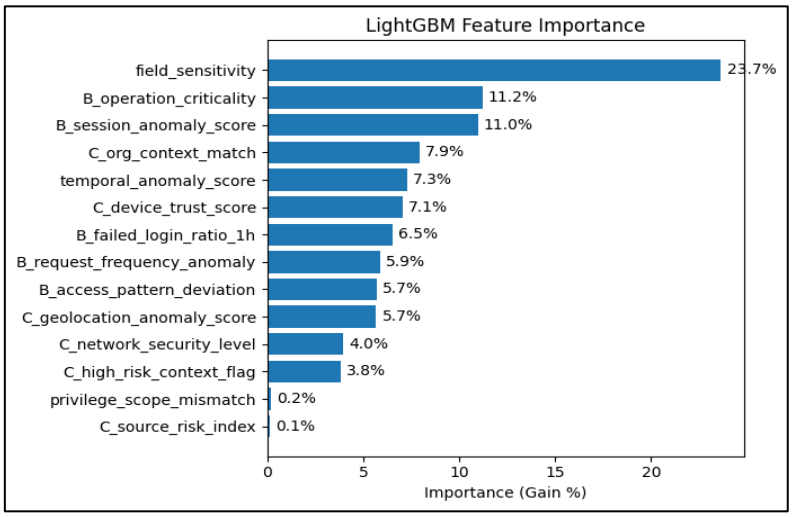

Figure 7. Feature Importance

Finally, the intentional introduction of randomly missing values completes this approach, simulating packet losses or corrupted logs, thereby testing the proxy's ability to maintain a consistent decision. The result is given in Table 11.

Table 11. Robustness of the Model (5% Noise)

Class	Precision	Recall	F1-Score	Support (N)
0 (Moderate)	0.98	0.97	0.97	5,250
1 (Strong)	0.94	0.95	0.95	4,500
2 (High)	0.95	0.94	0.94	3,750
3 (Maximum)	0.97	0.94	0.95	1,500
Global Accuracy	-	-	0.96	15,000

The performance analysis in a stabilized environment with 5% noise indicates high decision-making efficiency, achieving an overall accuracy of 96%. This decrease in uncertainty enables the model to achieve a precision of 0.98 at level 0, ensuring that cryptographic process simplifications are limited to legitimate traffic and that the risk of errors is minimal. Additionally, the system's robustness in handling sensitive data is demonstrated by an F1-score of 0.95 at level 3, indicating that the proxy effectively detects critical threats despite minor sensor fluctuations.

5.6 Throughput and System Latency

To validate that integrating artificial intelligence does not affect transaction throughput or proxy operation, a detailed analysis of its processing cycle was conducted. Table 12 shows the average latency observed, indicating that the inference phase is minimal in comparison to cryptographic process operations.

Table 12. Real-Time Proxy Latency Breakdown

Proxy Pipeline	Latency (ms)
Feature Engineering	0.45
ML Inference	0.6
Crypto Engine (L0-L3)	1.4
Context Switching	0.4
Total Added Latency	2.85

The system's throughput is assessed using two different setups to examine how LightGBM inference and dynamic encryption impact performance. The results are laid out in Table 13. The arrangements consisted of the Baseline, utilizing normal MongoDB without a proxy or encryption, and the Designed AdaptiCrypt-ML, which merges AI-driven dynamic encryption.

Table 13. Global System Metrics (End-to-End)

Metric	Without Proxy	Proposed ML Proxy (L0-L3)	Performance Impact
Avg Latency (ms)	0.45	3.25	x 7.2
P95 Latency (ms)	0.95	4.10	x 4.3
Throughput (QPS)	8 500	3 120	-63%
CPU Usage (%)	8%	45%	+ 37%
Memory (MB)	420	1 350	+ 930 MB

The cost of encryption remained the same, even though the average response time increased by 7.2 times. It stayed at 3.25 milliseconds, with a 95th percentile (P95) of 4.10 milliseconds. This is an acceptable balance between security concerns and efficiency. The proxy is useful in real-life situations since it can handle a steady speed of 3,120 QPS and a CPU usage of 45%. AI does not affect the scalability of the NoSQL system [20]. The system may change its security level in real-time depending on the risk since it uses multiple levels of cryptographic suites, from AES-192-GCM to a hybrid suite AES-ChaCha20-Poly1305 [12].

6. Discussion

AdaptiCrypt-ML's LightGBM model validates the proposed methodology with a cross-validation macro F1 score of 96.3%. The results indicate that statistical sensitivity (10.9%) is much less significant than behavioural qualities (49%). Because there are no continuous data and all distributions in the Kolmogorov-Smirnov tests have p-values greater than 0.05, the simulation correctly duplicates real-life access patterns. With a variance of 0.0018, it is nevertheless capable of solidifying the results, demonstrating that when employing MongoDB, the LightGBM technology can discover even minute modifications in complex data flows. Furthermore, compared to the AdaptiCrypt-ML coefficients, the normal response time is 3.50 milliseconds, thus making it suitable for deployment in professional settings. The General Data Protection Regulation (GDPR) and the instructions set out by the French Data Protection Commission (CNIL) are among the security legislation that the platform fully complies with.

In the context of validity and simplicity in particular, the use of AdaptiCrypt-ML offers a significant technological advancement. This strategy cannot be achieved using static encryption methods, such as MongoDB field encryption. It can be more scalable than computer drivers because of its intermediary function. Because the environment and the app might change the account, it represents a compromise between rapidity and safety. This real-time dynamic precision transforms the concept of security from a static boundary into an interactive smart system that adapts to context and immediately responds to any changes. Table 14 compares the limitations of existing industrial systems with the innovations provided by the AdaptiCrypt-ML system regarding flexibility, performance, and adaptability.

Table 14. Comparative Analysis of Existing Security Frameworks

	MongoDB FLE [23]	**UEBA-proxy [22]**	**Proposed Model**
Logic	Static Rules	Anomaly detection	Dynamic (LightGBM)
Granularity	Field-level	User session	Field + Operation + Risk
Client Overhead	High (Driver mod)	Moderate	Zero (Transparent Proxy)
Adaptability	N/A	Passive (Detection)	Level 0 to 3
F1-score	N/A	92.3%	99%
Latency	~3 ms	~5 ms	~4.5 ms

Although performant, AdaptiCrypt-ML is limited by its current architecture. Firstly, a 63% reduction in the number of queries per second (QPS) is observed in comparison to the native version of the system. Although the response time load still remains reasonable for sensitive applications, it can be improved by replacing the Flask framework with a faster asynchronous framework, such as FastAPI, which enables even more efficient processing of concurrent requests. Also, even though the statistical congruence was very good (KS $p > 0.05$ and no repetitions), the artificial dataset may need to be checked again with real, anonymized data to ensure the results are reliable. The LightGBM model was specifically trained to work with the MongoDB data structure. This means that if it is switched to another NoSQL database, like Cassandra or Redis, it is necessary to reconfigure it to accommodate the changes in structure and characteristics. Because HashiCorp Vault is so reliable, it creates a single point of failure. This means that a high-availability configuration is needed to keep services running and systems robust in real-world situations.

The AdaptiCrypt-ML model is based on a decoupled architecture that enables deployment across different NoSQL environments. The machine learning module, located at the proxy level, assesses risk independently of the storage engine. The generalisation of the model is ensured by matching NoSQL data structures to a unified sensitivity dictionary. On the other hand, encryption policies are applied consistently without modifying the database engine. Consequently, the proposed approach can be adapted to various NoSQL systems without altering the database engine. To highlight the framework's security and operational resilience, sensitive data are protected through adaptive encryption guided by user behaviour and contextual features, while key management is ensured via a high-availability Vault cluster. In addition, Data Encryption Keys (DEKs) are safely kept in RAM to faciliate processes throughout temporary interruptions. New encryption operations are paused during prolonged key management failures, whereas already encrypted data remain accessible. Furthermore, potential insider threats and adversarial manipulations are mitigated by monitoring feature anomalies in real time. In the meantime, the agent framework functions independently across database clusters, and security policies are always enforced across all nodes in the system. Response time is minimized, helping to meet enterprise-level query needs while preserving a high level of security.

Despite the high performance achieved in this study, a potential limitation is related to the dataset used for training and evaluation. The experiments were conducted on a synthetic dataset. An additional validation using real-world database traffic would be necessary to confirm the robustness of the proposed approach.

7. Conclusion and Future Work

The adaptive encryption method presented in this paper applies to NoSQL databases. It combines risk assessment techniques involving machine learning and dynamic field-level encryption according to contextual, behavioral, and data sensitivity attributes. Experimental analysis shows the proposed system has an outstanding prediction performance with 99.1% accuracy while delivering very low response times of around 3 ms, proving the applicability of the system. This demonstrates the capability of achieving a good balance between security and performance. From the results, it should be noted that experiments are performed with synthetic data and do not reflect any real-life scenarios. In future research, validation of the model with real data will be considered along with the implementation of online learning and support for other NoSQL database management systems. As can be seen, machine learning technologies can significantly enhance encryption and make it more intelligent.

References

[1] Akbar, S. K., Navya, V., & Suresh, K. (2025). Mastering NoSQL Databases: Strategies for Efficient Data Handling. https://doi.org/10.63328/books/978-93-47093-29-6.

[2] Mailewa, Akalanka, Susan Mengel, Lisa Gittner, and Hafiz Khan. "Mechanisms and Techniques to Enhance the Security of Big Data Analytic Framework with Mongodb and Linux Containers." Array 15 (2022): 100236.

[3] Zainal, Hana Yousuf. "Survey analysis: Enhancing the Security of Vectorization by Using Word2vec and CryptDB." Advances in Science, Technology and Engineering Systems Journal (2020). 374–380.

[4] Zhang, Dingwen, Shuang Yang, Ming Chen, Lei Zheng, Jiashu Fan, and Aidi Dong. "Adaptive Encryption Method of Sensitive Data in Data Center Database Based On Big Data Cross-Mapping Fusion Algorithm." Discover Applied Sciences 7, no. 8 (2025): 924.

[5] Kumar, Priyanka Rajan, and Sonia Goel. "A Secure and Efficient Encryption System Based on Adaptive and Machine Learning for Securing Data in Fog Computing." Scientific reports 15, no. 1 (2025): 11654.

[6] Premakumari, Sreeja Balachandran Nair, Gopikrishnan Sundaram, Marco Rivera, Patrick Wheeler, and Ricardo E. Pérez Guzmán. "Reinforcement Q-learning-based Adaptive Encryption Model for Cyberthreat Mitigation in Wireless Sensor Networks." Sensors 25, no. 7 (2025): 2056.

[7] Atlam, Hany F., and Gary B. Wills. "ANFIS for Risk Estimation in Risk-Based Access Control Model for Smart Homes." Multimedia Tools and Applications 82, no. 12 (2023): 18269-18298.

[8] Alharbe, Nawaf, Abeer Aljohani, Mohamed Ali Rakrouki, and Mashael Khayyat. "An Access Control Model Based on System Security Risk for Dynamic Sensitive Data Storage in the Cloud." Applied Sciences 13, no. 5 (2023): 3187.

[9] Jin, Ziqi, Dongmei Li, Xiaomei Zhang, and Zhi Cai. "Research on Dynamic Searchable Encryption Method Based on Bloom Filter." Applied Sciences 14, no. 8 (2024): 3379.

[10] Hu, Zhuobin, Jiabei Wang, Zhengkai Chen, Zhaoxuan Ge, Mingyu Bian, Lei Chen, and Yongbin Zhou. "SEAC: Dynamic Searchable Symmetric Encryption with Lightweight Update-Search Permission Control." Cybersecurity 8, no. 1 (2025): 75.

[11] Sheik, Syed Amma, and Amutha Prabakar Muniyandi. "Secure Authentication Schemes in Cloud Computing with Glimpse of Artificial Neural Networks: A Review." Cyber Security and Applications 1 (2023): 100002.

[12] Ferreira, Maurício J., Nuno A. Silva, Armando N. Pinto, and Nelson J. Muga. "Characterization of a Quantum Random Number Generator Based on Vacuum Fluctuations." Applied Sciences 11, no. 16 (2021): 7413.

[13] Sharma, Anuj, Alex Koohang, and Satender Pal Singh. "Information Security Policy Compliance: A Structured Review Using Scientometric Analysis and Topic Modeling." Journal of Global Information Management (JGIM) 33, no. 1 (2025): 1-32.

[14] Smith, Hussein. Python-MongoDB Atlas Integration: Exploring Advanced Python Libraries and Tools for Working with MongoDB Atlas (2024).

[15] Antonopoulos, Panagiotis, Arvind Arasu, Kunal D. Singh, Ken Eguro, Nitish Gupta, Rajat Jain, Raghav Kaushik et al. "Azure SQL Database Always Encrypted." In Proceedings of the 2020 ACM SIGMOD international conference on management of data, pp. 1511-1525. 2020.

[16] Yu, Xiaopeng, Wei Zhao, Yunfan Huang, Juan Ren, and Dianhua Tang. "Privacy-Preserving Outsourced Logistic Regression on Encrypted Data from Homomorphic Encryption." Security and Communication Networks 2022, no. 1 (2022): 1321198.

[17] Ladrham, Khalil, and Hicham Gueddah. "Sentiment Analysis on Moroccan Dialect of Arabic Combining NLP and ML Methods." In International Conference on Arabic Language Processing, pp. 3-16. Cham: Springer Nature Switzerland, 2024.

[18] Merzoug, Ahmed, Fehmi Özbayrak, John T. Foster, and Michael J. Pyrcz. "Beyond Random Forest: How Spatial Bagging and Spatial Random Forest Dominate for Subsurface Applications?." Computational Geosciences 29, no. 6 (2025): 52.

[19] Jin, Dongzi, Yiqin Lu, Jiancheng Qin, Zhe Cheng, and Zhongshu Mao. "SwiftIDS: Real-Time Intrusion Detection System Based on LightGBM and Parallel Intrusion Detection Mechanism." Computers & Security 97 (2020): 101984.

[20] Khan, Wisal, Teerath Kumar, Cheng Zhang, Kislay Raj, Arunabha M. Roy, and Bin Luo. "SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review." Big Data and Cognitive Computing 7, no. 2 (2023): 97.

[21] Ladrham, Khalil, Hicham Gueddah, and Brahim Ouben Hssain. 2026. “Benchmarking Lightweight Convolution Neural Networks for Children’s Arabic Handwriting”. Journal of Innovative Image Processing 8 (1): 216-32. https://doi.org/10.36548/jiip.2026.1.012.

[22] Fuentes, Jose, Ines Ortega-Fernandez, Nora M. Villanueva, and Marta Sestelo. "Cybersecurity Threat Detection Based on a UEBA Framework Using Deep Autoencoders." arXiv preprint arXiv:2505.11542 (2025).

[23] MongoDB Inc. 2025. "Queryable Encryption v2: Fast Searchable Encryption." MongoDB Documentation. https://www.mongodb.com/docs/manual/core/queryable-encryption

[24] Stiawan, Deris, Mohd Yazid Bin Idris, Alwi M. Bamhdi, and Rahmat Budiarto. "CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection." IEEE access 8 (2020): 132911-132921.

[25] Alsaedi, Abdullah, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar. "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems." Ieee Access 8 (2020): 165130-165150.