International Journal of Mathematics and Statistics (IJMS)

DOI: 10.5281/zenodo.21846552 | ISSN: 3141-6438 | Volume: 1 | Issue: 2 | 2026

Machine Learning-Assisted Selection of Algebraic Structures for Post-Quantum Cryptographic Systems: A Multi-Objective Computational Optimization Model

Tombotamunoa W. J. LAWSON[1]

[1]Department of Mathematics, Ignatius Ajuru University of Education, Rumuolumeni, Port Harcourt, Nigeria

Corresponding Author: tombotamunoa.lawson@iaue.edu.ng

Abstract

The transition to post-quantum cryptography creates a mathematical decision problem in which security strength, algebraic structure, key and ciphertext sizes, execution time, memory demand, implementation complexity, and side-channel resilience must be considered simultaneously. This paper develops a machine learning-assisted multi-objective computational framework for ranking candidate algebraic structures and parameter configurations for post-quantum cryptographic deployment. The framework combines a normalized utility model, feasibility constraints, Pareto dominance, and supervised learning. A reproducible simulation study with 2,400 synthetic candidate configurations representing lattice-, code-, hash-, multivariate-, and group-based families is used to demonstrate the methodology without presenting the simulated values as implementation benchmarks. Random Forest, Gradient Boosting, and Support Vector Machine classifiers are compared using a stratified 70:30 train-test split and five-fold cross-validation. On the held-out test set, Gradient Boosting achieved an accuracy of 0.901, F1-score of 0.869, and ROC-AUC of 0.968, while five-fold cross-validation produced a mean ROC-AUC of 0.960. Security strength was the dominant predictor in permutation analysis, but side-channel resilience, algebraic dimension, key size, memory demand, and decapsulation time also contributed to the selection boundary. A Pareto analysis identified 25 non-dominated feasible configurations in the simulated design space. The proposed model provides a transparent mathematical mechanism for combining cryptographic constraints with data-driven classification and can be adapted to

measured benchmark datasets as they become available. The principal contribution is therefore methodological: it gives a reproducible bridge between computational algebra, multi-criteria optimization, and machine learning for cryptographic parameter selection.

Keywords: post-quantum cryptography; computational mathematics; machine learning; algebraic structures; lattice cryptography; multi-objective optimization; Pareto frontier; gradient boosting.

1. Introduction

Public-key cryptography is built on mathematical problems whose computational difficulty supplies the security of the system. Classical schemes such as RSA rely on integer factorization, while elliptic-curve and related systems rely on discrete logarithm problems. The development of quantum algorithms, particularly Shor's algorithm, changed the long-term security assumptions surrounding these systems [8]. Post-quantum cryptography therefore seeks cryptographic constructions based on problems for which no efficient classical or quantum solution is presently known.

Lattice-based methods have become especially important because they combine strong mathematical foundations with comparatively efficient implementations. Regev's learning-with-errors formulation connected average-case cryptographic constructions to worst-case lattice problems [9]. Subsequent developments produced practical module-lattice systems and culminated in the standardization of ML-KEM and ML-DSA by the National Institute of Standards and Technology (NIST) [12, 13]. Earlier work by John and collaborators also emphasized the interaction between group theory, lattices, number theory, nilpotent groups, and algebraic constructions in cryptography [1–7].

The choice of a cryptographic construction, however, is not determined by security strength alone. A parameter set with high nominal security may impose unacceptable key sizes, latency, memory demand, bandwidth cost, or implementation difficulty. Conversely, a highly efficient candidate may fall below a required security threshold. The selection problem is therefore naturally multi-objective. It contains continuous variables, discrete family choices, hard feasibility constraints, and nonlinear trade-offs. These characteristics motivate a computational mathematics framework in which optimization and machine learning are used together.

This paper develops such a framework. The proposed model has four layers. First, a feature vector describes each candidate algebraic or cryptographic configuration. Second, hard constraints remove infeasible candidates. Third, a weighted normalized utility function and Pareto dominance model quantify trade-offs among feasible configurations. Fourth, supervised learning approximates a nonlinear suitability boundary from candidate features. Because publicly comparable benchmark data across substantially different post-quantum families are heterogeneous, platform-dependent, and often incomplete, the numerical section uses a clearly labelled synthetic dataset. The purpose is to test the mathematical framework and computational workflow, not to claim new measured performance for real cryptographic standards.

The contributions are summarized as follows. A constrained multi-objective model is

formulated for cryptographic structure selection; a machine learning layer is integrated with the optimization model; three classifiers are evaluated under identical data partitions; Pareto-efficient configurations are extracted; and a reproducible simulation protocol is reported with sufficient detail for replacement by empirical benchmark data. The framework is designed to complement, rather than replace, cryptanalytic evidence and standards-based security evaluation.

2. Related Literature

2.1. Algebraic structures and post-quantum cryptography

Lattices are discrete additive subgroups of Euclidean space generated by linearly independent basis vectors. Their computational problems, including variants of the shortest vector problem (SVP) and closest vector problem (CVP), provide the foundation for an extensive family of post-quantum constructions. Regev [9] established a major connection between learning with errors (LWE) and worst-case lattice problems. Peikert [10] surveyed the subsequent development of lattice cryptography and its mathematical foundations.

The module-lattice approach used in modern standardized schemes provides a balance between the compactness of ring-based constructions and the conservative structure of unstructured lattices. NIST FIPS 203 specifies ML-KEM and relates its security to Module-LWE [12]. NIST FIPS 204 specifies ML-DSA, a module-lattice-based digital signature standard [13]. The earlier CRYSTALS-Kyber specification gives detailed algebraic and implementation motivation for the construction that became the basis of ML-KEM [11].

The algebraic literature cited in the present study broadens this perspective. John, Udoaka, and Udoakpan [1] investigated group theory in lattice-based cryptography. John and collaborators studied lattices in quantum-era cryptography [5], number theory in RSA encryption systems [6], nilpotent groups in key exchange [4], solvable and monolithic character conditions [2], conjugacy classes in finitely generated groups [3], and fuzzy group actions in near-rings [7]. These works motivate the inclusion of algebraic family, dimension, structural regularity, and implementation-related features in a unified selection model.

2.2. Machine learning as a decision layer

Machine learning is useful when a decision surface depends on several interacting variables and cannot be represented adequately by a single linear score. Random Forests aggregate randomized decision trees and are robust to nonlinear relations and feature interactions [14]. Support Vector Machines construct separating surfaces in transformed feature spaces and remain effective in moderate-dimensional classification settings [15]. Gradient boosting builds an additive predictive model by sequentially fitting weak learners to residual structure [16].

In cryptographic engineering, machine learning must be used carefully. A predictive classifier cannot establish cryptographic security. It can, however, assist with engineering decisions when the labels encode a separately justified suitability criterion. The present work therefore places the machine learning classifier after explicit security and feasibility constraints. This separation is important: a candidate that violates a required security threshold is rejected

regardless of its predicted deployment score.

2.3. Multi-objective optimization

Many engineering design problems have several conflicting objectives. Pareto optimality offers a natural formalization: a solution is non-dominated if no alternative improves at least one objective without worsening another. Evolutionary multi-objective algorithms such as NSGA-II are widely used when the design space is complex [17]. In the present study, the simulated design space is sufficiently small to identify the Pareto set directly. The same formulation can later be embedded in NSGA-II, particle swarm optimization, Bayesian optimization, or mixed-integer search when the candidate space is larger.

3. Mathematical Formulation

3.1. Candidate representation

Let $C = \{c_1, c_2, \ldots, c_N\}$ denote a finite set of candidate cryptographic configurations. Each candidate c_i is represented by

$$\mathbf{x}_i = (f_i, n_i, s_i, k_i, z_i, t_i^{(e)}, t_i^{(d)}, m_i, r_i, h_i, q_i), \tag{1}$$

where f_i is the algebraic family, n_i is an algebraic dimension or size parameter, s_i is modeled security strength in bits, k_i is public-key size, z_i is ciphertext/signature payload size, $t_i^{(e)}$ and $t_i^{(d)}$ are forward and reverse operation times, m_i is memory demand, r_i is a structural regularity score, h_i is an implementation-complexity score, and q_i is a side-channel resilience proxy.

The interpretation of n_i depends on the family. For a lattice model it may represent a module or ambient dimension; for a code-based model it may represent a code length-related parameter; for a group-based construction it may represent an order-related or presentation-size surrogate. The model therefore treats n_i as a computational design variable rather than asserting that these dimensions are directly comparable across families.

3.2. Normalization

For a benefit criterion x whose larger value is preferred, min-max normalization is

$$B(x_i) = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}. \tag{2}$$

For a cost criterion whose smaller value is preferred,

$$C(x_i) = 1 - \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}. \tag{3}$$

These transformations map all criteria to $[0, 1]$ and ensure that larger normalized values correspond to more desirable values.

3.3. Utility function

Define total computational latency

$$\ell_i = t_i^{(e)} + t_i^{(d)}. \tag{4}$$

A baseline utility score is then defined by

$$U_i = w_s B(s_i) + w_k C(k_i) + w_z C(z_i) + w_\ell C(\ell_i) + w_m C(m_i) \tag{5}$$

$$+ w_q q_i + w_r r_i + w_h (1 - h_i), \tag{6}$$

where

$$\sum_j w_j = 1, \qquad w_j \geq 0. \tag{7}$$

For the simulation we use

$$(w_s, w_k, w_z, w_\ell, w_m, w_q, w_r, w_h) = (0.34, 0.14, 0.08, 0.14, 0.08, 0.12, 0.05, 0.05). \tag{8}$$

The weighting deliberately assigns the largest contribution to security while retaining non-negligible engineering costs.

3.4. Feasibility constraints

Let $s_{\min}$ denote the minimum acceptable security strength. A generic feasible set is

$$\mathcal{F} = \{c_i \in C : s_i \geq s_{\min},\ k_i \leq k_{\max},\ \ell_i \leq \ell_{\max},\ m_i \leq m_{\max}\}. \tag{9}$$

Not all upper bounds need to be active in every deployment. In the numerical experiment the only universal hard constraint used to construct the suitability label is $s_i \geq 128$ bits; resource variables remain in the utility function so that the classifier must learn trade-offs rather than merely reproduce several hard cutoffs.

Proposition 1. *If all normalized benefit terms lie in* $[0, 1]$*, all normalized cost terms are transformed to* $[0, 1]$*, and the weights are nonnegative and sum to one, then* $0 \leq U_i \leq 1$.

Proof. Each component of U_i belongs to $[0, 1]$. Hence U_i is a convex combination of quantities in $[0, 1]$, and the result follows immediately. □

3.5. Pareto dominance

For the two-dimensional illustration used in this paper, define security as a benefit and deployment cost as a cost. Candidate c_a dominates c_b when

$$s_a \geq s_b, \qquad d_a \leq d_b, \tag{10}$$

with at least one strict inequality. The deployment-cost index is

$$d_i = 0.35\widetilde{k}_i + 0.15\widetilde{z}_i + 0.30\widetilde{\ell}_i + 0.20\widetilde{m}_i, \tag{11}$$

where tildes denote min-max normalized cost variables. The Pareto frontier is the subset of feasible candidates that are not dominated.

4. Machine Learning-Assisted Selection

4.1. Suitability label

The synthetic experiment defines a latent suitability function

$$L_i = U_i + 0.04 \sin(n_i/150) + \varepsilon_i, \tag{12}$$

where $\varepsilon_i \sim \mathcal{N}(0, 0.035^2)$. The sinusoidal term introduces a mild nonlinear interaction with the dimension variable, while ε_i represents unobserved implementation factors. A candidate is labelled suitable if

$$y_i = \begin{cases} 1, & s_i \geq 128 \text{ and } L_i \geq \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{13}$$

where τ is the 45th percentile of L_i among candidates satisfying the security constraint. This construction produced a positive-class proportion of 35.625%.

This label is a modelling device, not a cryptographic certification. The aim is to determine whether machine learning can recover a nonlinear decision rule when suitability depends jointly on security and resource criteria.

4.2. Classifiers

Three classifiers are used.

Random Forest. For B trees $T_b(\mathbf{x})$, the class probability estimate is

$$\widehat{p}_{RF}(y = 1 \mid \mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} T_b(\mathbf{x}). \tag{14}$$

We use 500 trees with class balancing and a minimum leaf size of three.

Gradient Boosting. An additive model is constructed as

$$F_M(\mathbf{x}) = F_0(\mathbf{x}) + \sum_{m=1}^{M} \eta \, \gamma_m h_m(\mathbf{x}), \tag{15}$$

where h_m is a depth-limited decision tree, η is the learning rate, and γ_m is the stage-wise step size. The experiment uses 200 estimators, maximum depth three, and learning rate 0.05.

Support Vector Machine. The SVM seeks a separating surface in a kernel-induced feature space. In its soft-margin form the optimization is

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_i \xi_i \tag{16}$$

subject to

$$y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0. \tag{17}$$

The implementation uses a radial basis kernel and $C = 3$.

4.3. Decision score

For ranking the feasible candidates after model fitting, define

$$D_i = 0.55\widehat{p}_i + 0.30\widetilde{s}_i + 0.15(1 - d_i), \tag{18}$$

where $\widehat{p}_i$ is the predicted probability of suitability, $\widetilde{s}_i$ is normalized security strength, and d_i is normalized deployment cost. This final score gives the learning model the largest influence while preserving explicit security and resource information.

Algorithm 1 Machine learning-assisted cryptographic configuration selection

Require: Candidate set C, minimum security $s_{\min}$, feature matrix X
Ensure: Ranked feasible set and Pareto frontier

1: Normalize benefit and cost criteria
2: Compute utility U_i for every candidate
3: Reject candidates violating hard security constraints
4: Split data into stratified training and test sets
5: Train Random Forest, Gradient Boosting, and SVM models
6: Evaluate accuracy, precision, recall, F1-score, and ROC-AUC
7: Select the best-performing model by ROC-AUC and balanced classification performance
8: Predict $\widehat{p}_i$ for feasible candidates
9: Compute deployment cost d_i and decision score D_i
10: Identify the non-dominated Pareto set
11: Rank candidates by D_i subject to feasibility and cryptanalytic review

5. Simulation Design

5.1. Reproducible synthetic data

A total of 2,400 candidate configurations were generated with a fixed NumPy random seed of 20260807. The family mixture was 42% lattice-based, 18% code-based, 14% hash-based, 13% multivariate, and 13% group-based. Family-specific distributions were used to generate plausible relative patterns, such as comparatively larger public keys for code-based candidates and higher structural regularity for lattice or multivariate candidates. These patterns are illustrative and must not be interpreted as standardized benchmark values.

Table 1: Variables in the simulated design space.

Variable	Type	Interpretation
Family	categorical	Lattice, code, hash, multivariate, or group-based archetype
Dimension n	integer	Family-dependent algebraic size parameter
Security bits s	continuous	Modelled security-strength proxy
Key size k	continuous	Public-key storage cost in kB
Payload z	continuous	Ciphertext/signature-like payload in kB
Encapsulation/forward time	continuous	Forward operation latency in ms
Decapsulation/reverse time	continuous	Reverse operation latency in ms
Memory m	continuous	Working-memory demand in kB
Regularity r	$[0, 1]$	Algebraic structural regularity proxy
Implementation complexity h	$[0, 1]$	Relative engineering-complexity proxy
Side-channel score q	$[0, 1]$	Relative resilience proxy

The dataset was divided using a stratified 70:30 split. The categorical family variable was one-hot encoded. Continuous predictors were standardized for the SVM pipeline; the same preprocessing object was used consistently across all models to avoid data leakage. Five-fold stratified cross-validation was also performed on the full simulated dataset.

5.2. Evaluation metrics

For true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), the standard metrics are

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{19}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{20}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{21}$$

$$F_1 = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{22}$$

The area under the receiver operating characteristic curve (ROC-AUC) measures ranking quality over all probability thresholds.

6. Results

6.1. Held-out test performance

Table 2 reports the test-set results. Gradient Boosting produced the highest accuracy, F1-score, and ROC-AUC. Random Forest produced slightly higher recall than Gradient Boosting, while the SVM achieved strong recall but lower precision.

Table 2: Held-out test-set performance.

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Random Forest	0.894	0.800	0.938	0.863	0.958
Gradient Boosting	**0.901**	**0.825**	0.918	**0.869**	**0.968**
Support Vector Machine	0.858	0.741	0.926	0.823	0.944

The ROC curves in Figure 1 show that all three models substantially exceed random ranking in the synthetic experiment. Gradient Boosting gives the strongest overall discrimination.

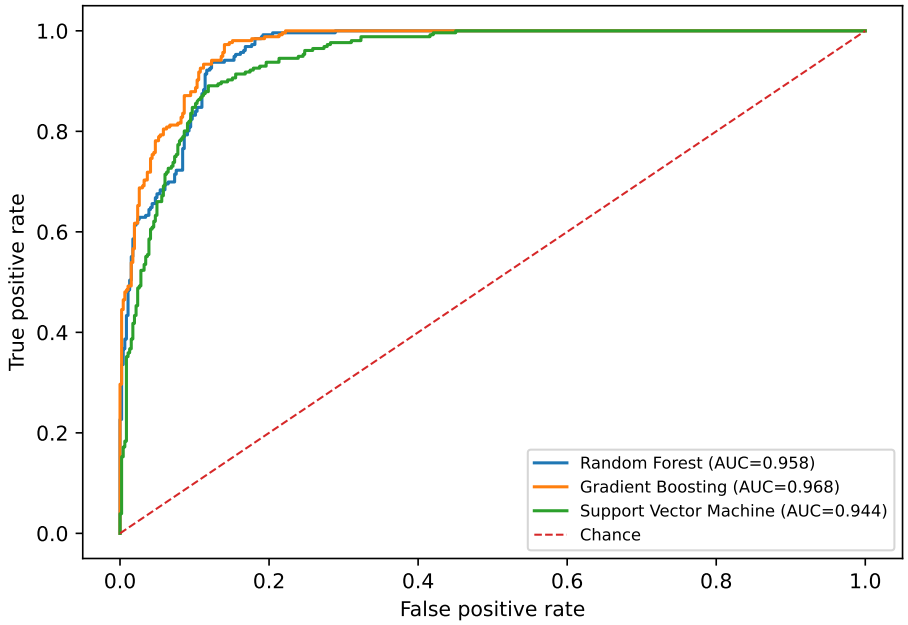

Figure 1: ROC curves for the three classifiers on the held-out synthetic test set.

6.2. Cross-validation

Table 3 reports five-fold cross-validation. The small standard deviations indicate that the observed ranking is not driven by a single train-test split. Gradient Boosting again gives the highest mean ROC-AUC at 0.960.

Table 3: Five-fold stratified cross-validation results (mean ± standard deviation).

Model	Accuracy	F1	ROC-AUC
Random Forest	0.880 ± 0.011	0.840 ± 0.016	0.956 ± 0.004
Gradient Boosting	**0.885 ± 0.011**	**0.842 ± 0.015**	**0.960 ± 0.005**
Support Vector Machine	0.878 ± 0.009	0.841 ± 0.011	0.947 ± 0.008

6.3. Permutation importance

Permutation importance was calculated for the Gradient Boosting pipeline using reduction in ROC-AUC as the importance measure. Security strength is dominant because the synthetic label contains an explicit 128-bit feasibility condition and the utility function places the largest weight on security. This is expected and desirable: the model should not rank resource efficiency above the minimum security requirement.

Table 4: Leading permutation importance values for the Gradient Boosting model.

Feature	Mean importance	Standard deviation
Security bits	0.4480	0.0244
Side-channel score	0.0218	0.0025
Dimension	0.0179	0.0026
Key size	0.0112	0.0014
Memory	0.0107	0.0026
Decapsulation time	0.0089	0.0021
Regularity	0.0065	0.0021
Ciphertext size	0.0049	0.0011

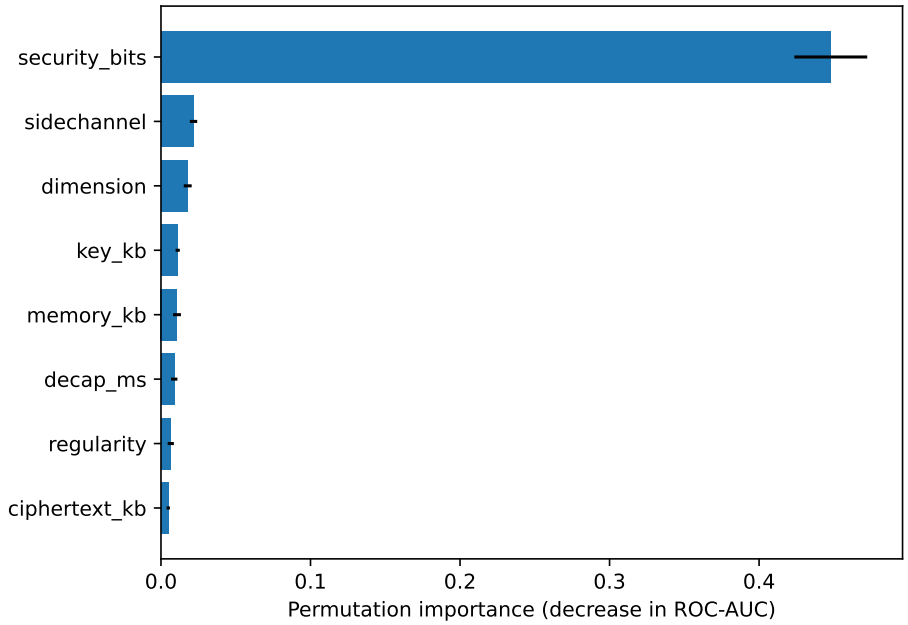

Figure 2: Permutation importance for the leading predictors.

The smaller but nonzero importance of implementation-oriented variables indicates that the learned boundary is not reducible to the security threshold alone. In a real benchmark dataset, the relative importance pattern would depend on the deployment platform, measured side-channel countermeasures, optimization level, and security category.

6.4. Pareto frontier

Among candidates satisfying the 128-bit security constraint, 25 configurations were non-dominated in the two-objective security-versus-deployment-cost plane. Figure 3 illustrates the frontier. Movement along the frontier represents the unavoidable trade-off between higher modeled security and higher resource cost.

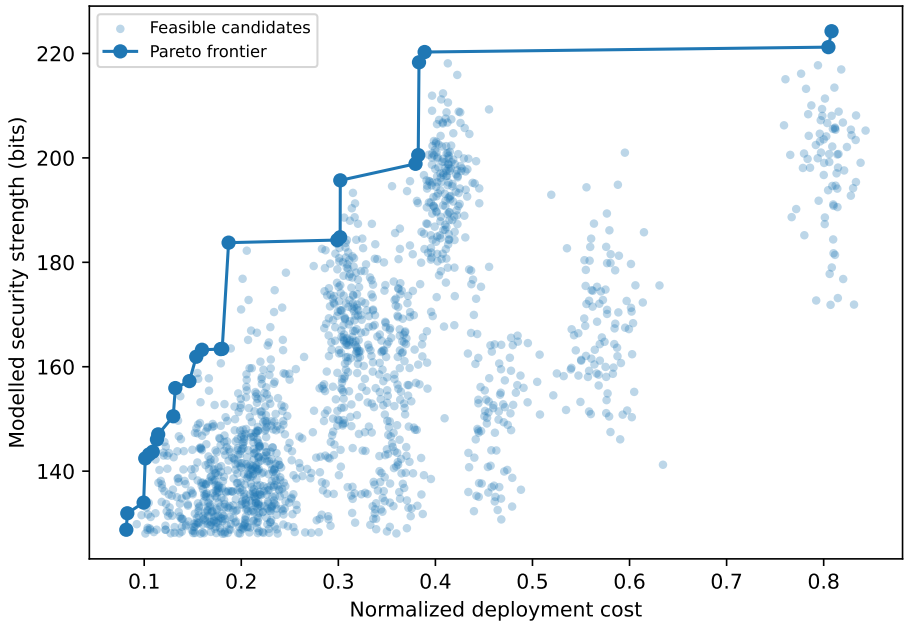

Figure 3: Pareto frontier for feasible synthetic candidates.

The Pareto set should be interpreted as a screening set. A non-dominated candidate is not automatically secure or standardized; it simply survives the model's resource/security comparison. Cryptanalytic review, standards compliance, protocol context, and implementation testing remain necessary.

6.5. Illustrative top-ranked candidates

Table 5 gives five top-ranked synthetic candidates under the decision score D_i. Because the numerical values are generated by the simulation model, they are identified only by family and parameter features rather than by names of real cryptographic algorithms.

Table 5: Top five synthetic candidates ranked by the combined decision score.

Family	Dim.	Security	Key (kB)	Payload (kB)	Latency (ms)	Memory (kB)	ML prob.	Score
Lattice	1024	218.31	3.15	2.50	2.28	91.23	0.968	0.907
Lattice	1024	218.11	3.12	2.52	2.30	103.12	0.966	0.900
Lattice	1024	211.91	3.15	2.50	2.25	96.40	0.973	0.887
Lattice	1024	212.36	3.17	2.50	2.26	105.30	0.970	0.885
Lattice	1024	215.89	3.14	2.46	2.26	113.35	0.954	0.885

The concentration of top-ranked candidates in the lattice archetype is a consequence of the family distributions used in this synthetic experiment. It must not be interpreted as evidence that every lattice construction dominates every non-lattice construction. With empirical benchmark data or different application weights, the ranking could change materially.

7. Sensitivity and Robustness Analysis

A decision framework is only useful if its conclusions are not entirely determined by arbitrary weights. Let w_s vary while the remaining weights are rescaled proportionally to maintain a unit sum. When w_s increases, candidates with higher security strength move upward in the ranking, while low-cost configurations become less influential. Conversely, reducing w_s gives greater importance to latency, key size, payload size, and memory. This behavior is mathematically expected from the convex utility formulation.

The model also separates hard constraints from soft preferences. This distinction improves robustness. If a deployment requires $s_i \geq 192$, the feasible set can be recomputed without retraining the classifier. If a constrained device imposes $m_i \leq 64$ kB, the memory requirement can be added as a hard constraint. In contrast, if memory is merely preferred to be small, it remains a weighted cost term. Thus the same model supports server, desktop, mobile, embedded, and bandwidth-constrained contexts by changing the feasible region and weight vector rather than changing the underlying mathematical structure.

The five-fold cross-validation results provide an additional robustness check on the predictive layer. Mean ROC-AUC values remained between 0.947 and 0.960 with low fold-to-fold variability. Nevertheless, these values describe recovery of a simulated decision rule. They do not quantify performance on measured cryptographic benchmarks.

8. Discussion

The numerical experiment supports three main observations. First, multi-objective cryptographic selection can be represented cleanly as a constrained optimization problem. Security acts as a mandatory feasibility property, while resource requirements appear as costs. This avoids the undesirable situation in which a highly efficient but insecure candidate is selected merely because a weighted average compensates for insufficient security.

Second, a machine learning classifier can approximate a nonlinear suitability surface that combines algebraic and engineering variables. Gradient Boosting performed best in the current experiment, but the difference from Random Forest was modest. This suggests that the practical value of the learning layer may come less from a particular algorithm and more from the ability to capture interactions that are difficult to encode manually.

Third, Pareto analysis adds interpretability. A single weighted score produces an ordering, but it may hide alternative trade-offs. The Pareto frontier exposes configurations for which no simultaneous improvement in security and deployment cost exists within the sampled design space. Decision makers can therefore inspect the frontier before applying application-specific weights.

The framework also relates naturally to the algebraic cryptography literature. Work on group theory and lattice-based cryptography [1], quantum-era lattices [5], nilpotent-group key exchange [4], and conjugacy-based structures [3] demonstrates that cryptographic constructions can arise from diverse algebraic settings. A selection framework should therefore be capable of incorporating both structural descriptors and implementation metrics rather than assuming a

single family in advance.

From a computational mathematics perspective, the model can be extended in several directions. Continuous parameter optimization can be handled by evolutionary methods; uncertainty in benchmark values can be propagated by Monte Carlo sampling; weight uncertainty can be modeled by Bayesian distributions; and cryptanalytic uncertainty can be represented by robust or distributionally robust optimization. Graph-based representations of algebraic constructions could also allow graph neural networks to learn structure-dependent performance descriptors, although such applications require carefully curated data.

9. Limitations

The main limitation is that the numerical data are synthetic. This was intentional: cross-family performance figures reported in different publications are produced on different processors, software versions, optimization settings, and security categories, making direct aggregation potentially misleading. The current study therefore demonstrates the framework under controlled conditions and does not claim that the generated key sizes, timing values, memory values, or security estimates are measurements of ML-KEM, ML-DSA, Classic McEliece, or any other standardized algorithm.

Second, the side-channel and implementation-complexity variables are proxies. In a real system, side-channel resilience should be assessed using implementation-specific evidence such as constant-time behavior, masking, leakage tests, fault resistance, and platform-specific countermeasures. Third, a machine learning probability is not a security proof. Security claims must continue to rely on cryptanalysis, reductions, standards evaluation, parameter analysis, and implementation review.

Finally, the utility weights are application-dependent. The chosen weights illustrate one security-first setting and should not be treated as universal. A practical implementation should obtain weights from system requirements, analytic hierarchy methods, stakeholder elicitation, or optimization against explicit service-level constraints.

10. Conclusion

This paper proposed a machine learning-assisted multi-objective computational model for selecting algebraic structures and parameter configurations in post-quantum cryptographic systems. The framework combines hard feasibility constraints, normalized utility modelling, Pareto dominance, supervised classification, and a final deployment score. A reproducible synthetic experiment with 2,400 candidate configurations showed that Gradient Boosting recovered the simulated suitability rule with test ROC-AUC 0.968 and five-fold mean ROC-AUC 0.960. The Pareto procedure identified 25 non-dominated feasible candidates, demonstrating how security and deployment cost can be examined jointly.

The results should be understood as validation of a mathematical workflow rather than empirical benchmarking of real algorithms. The immediate next step is to populate the same feature schema with standardized measurements obtained on a common hardware/software

platform for ML-KEM, ML-DSA, and selected alternative families. With such data, the model can support transparent, application-specific selection while maintaining the essential principle that machine learning complements but does not replace cryptographic security analysis.

Data and Reproducibility Statement

The numerical results in this paper were generated from a synthetic dataset using a fixed random seed (20260807). The simulation code generates the 2,400 observations, fits all three classifiers, computes cross-validation results, permutation importance, Pareto dominance, and the final rankings. Because the dataset is synthetic, it contains no confidential or personal information and can be regenerated exactly from the accompanying script.

Funding

The author received no specific funding for this work.

Conflict of Interest

The author declares no conflict of interest.

References

[1] John, M. N., Udoaka, O. G., & Udoakpan, I. U. (2023). Group theory in lattice-based cryptography. *International Journal of Mathematics and Its Applications, 11*(4), 111–125.

[2] John, M. N., Udoaka, O. G., & Musa, A. (2023). Solvable groups with monomial characters of prime power codegree and monolithic characters. *Bulletin of Mathematics and Statistics Research, 11*(7), 1–4. https://doi.org/10.33329/bomsr.11.4.98

[3] Michael, N. J., Musa, A., & Udoaka, O. G. (2023). Conjugacy classes in finitely generated groups with small cancellation properties. *European Journal of Statistics and Probability, 12*(1), 1–9. https://doi.org/10.37745/ejsp.2013/vol12n119

[4] John, M. N., Udoaka, O. G., & Musa, A. (2023). Nilpotent groups in cryptographic key exchange protocol for $N \geq 1$. *Journal of Mathematical Problems, Equations and Statistics, 4*(2), 32–34. https://doi.org/10.22271/math.2023.v4.i2a.103

[5] John, M. N., Ozioma, O., Ngozi, O. P., Egbogho, H. E., & Udoaka, O. G. (2023). Lattices in quantum-era cryptography. *International Journal of Research Publication and Reviews, 4*(11), 2175–2179. https://doi.org/10.5281/zenodo.10207210

[6] John, M. N., Ozioma, O., Obukohwo, V., & Egbogho, H. E. (2023). Number theory in RSA encryption systems. *GPH–International Journal of Mathematics, 6*(11), 7–16. https://doi.org/10.5281/zenodo.10207361

[7] John, M. N., & Udoakpan, I. U. (2023). Fuzzy group action on an R-subgroup in a near-ring. *International Journal of Mathematics and Statistics Studies, 11*(4), 27–31. https://doi.org/10.37745/ijmss.13/vol11n42731

[8] Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (pp. 124–134). IEEE. https://doi.org/10.1109/SFCS.1994.365700

[9] Regev, O. (2005). On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing* (pp. 84–93). ACM. https://doi.org/10.1145/1060590.1060603

[10] Peikert, C. (2016). A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science, 10*(4), 283–424. https://doi.org/10.1561/0400000074

[11] Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., Seiler, G., & Stehlé, D. (2021). *CRYSTALS-Kyber: Algorithm specifications and supporting documentation (Version 3.01)*.

[12] National Institute of Standards and Technology. (2024). *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard*. U.S. Department of Commerce. https://doi.org/10.6028/NIST.FIPS.203

[13] National Institute of Standards and Technology. (2024). *FIPS 204: Module-Lattice-Based Digital Signature Standard*. U.S. Department of Commerce. https://doi.org/10.6028/NIST.FIPS.204

[14] Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32. https://doi.org/10.1023/A:1010933404324

[15] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297. https://doi.org/10.1007/BF00994018

[16] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

[17] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197. https://doi.org/10.1109/4235.996017

[18] Micciancio, D., & Regev, O. (2009). Lattice-based cryptography. In D. J. Bernstein, J. Buchmann, & E. Dahmen (Eds.), *Post-Quantum Cryptography* (pp. 147–191). Springer. https://doi.org/10.1007/978-3-540-88702-7_5

[19] Albrecht, M. R., Player, R., & Scott, S. (2015). On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology, 9*(3), 169–203. https://doi.org/10.1515/jmc-2015-0016

[20] Chen, Y., & Nguyen, P. Q. (2011). BKZ 2.0: Better lattice security estimates. In *ASIACRYPT 2011* (pp. 1–20). Springer. https://doi.org/10.1007/978-3-642-25385-0_1

[21] Bos, J. W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., Seiler, G., & Stehlé, D. (2018). CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy* (pp. 353–367). IEEE. https://doi.org/10.1109/EuroSP.2018.00032