Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 2-B, pp. 971–978
DOI : 10.47974/JDMSC-2548

ML-based verification systems using discrete mathematical models

Nilesh M. Pawar [†]
Department of Information Technology
Yadavrao Tasgaonkar College of Engineering and Technology
Bardi 410201
Maharashtra
India

Raju M. Sairise *
Department of Electronics & Telecommunication Engineering
Yadavrao Tasgaonkar College of Engineering and Technology
Bardi 410201
Maharashtra
India

Priya Karemore [§]
Archana Jadhav [‡]
Department of Information Technology
JSPM's Rajarshi Shahu College of Engineering
Pune 411033
Maharashtra
India

Dilmurod Turaev [@]
Department of Information Technology and Exact Sciences
Termez University of Economics and Service
Termez 190100
Uzbekistan

[†] *E-mail:* nmp206@gmail.com
* *E-mail:* rairise566@gmail.com (Corresponding Author)
[§] *E-mail:* priya.karemore@gmail.com
[‡] *E-mail:* kadamarchana75@gmail.com
[@] *E-mail:* dilmurod_torayev@tues.uz

© Taru Publications

Samir N. Ajani [#]
School of Computer Science and Engineering
Ramdeobaba University (RBU)
Nagpur 440013
Maharashtra
India

Abstract

Verification systems are very important for making sure that computer processes are correct and reliable in many areas, such as hacking, software validation, and data quality checks. This study shows a new system that combines machine learning (ML) methods with discrete mathematical models to make testing more accurate and faster. The suggested system uses ideas from vector measure spaces and orthogonal sequences to describe difficult proof tasks on a solid mathematical base. Adding vectorial convergence qualities to the system lets it improve testing results over and over, which cuts down on the false positives and raises the rate of detection.

Subject Classification: *31A10, 33E30.*

Keywords: *Discrete mathematical modeling, Vector metric space, Orthogonal sequence, Verification framework performance.*

1. Introduction

Software correctness validation, cybersecurity, and data integrity guarantee are just a few of the many computer applications that depend on verification systems. Modern systems are getting more complicated, so we need testing methods that are not only correct, but also quick, and able to handle large systems. Most of the time, thorough checking methods or heuristic-based techniques are used in traditional verification methods [1]. These can be hard to run on computers and don't always promise formal truth. In this situation, machine learning (ML) has become a hopeful model because it provides flexible and data-driven verification methods that can handle big and complicated verification jobs [2], [3]. However, ML models usually work as "black boxes," which raises questions about how well they can be understood, how reliable they are, and whether they come with written promises [4], [5], [6]. These are all very important in very important areas like security and safety-critical systems. Adding discrete mathematical models to machine learning-based proof systems has become more popular as a way to deal with these problems [7]. Discrete mathematics is a strong base for formal reasoning. It models and analyses verification issues in depth using vector measure spaces and orthogonal sequences. Discrete mathematical models provide a formal and precise framework for representing and analyzing verification problems. By leveraging

[#] *E-mail:* **samir.ajani@gmail.com**

structures such as vector metric spaces and orthogonal sequences, these models enable rigorous reasoning, support stable convergence, and minimize redundancy in verification processes. Machine learning is flexible and discrete mathematics is precise. Machine learning and discrete mathematical models are used in a novel testing method [8]. To provide reliable test results, the system uses machine learning and mathematics. Figure 1 shows ML-based testing system architecture. To prove its point, it combines flexible learning with rigid mathematical reasoning.

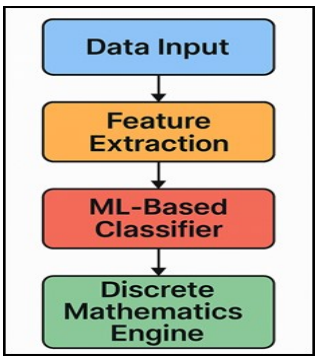

Figure 1
System Architecture of the ML-Based Verification Framework

Vectorial convergence in metric spaces and the properties of orthogonal sets are two ways that the system can improve proof over and over again. It works faster and less likely to make mistakes this way [9]. The discrete mathematical design also makes it easier to formalize proof states. This makes them simpler to understand and suggests that the system should be stable in theory.

2. Method Work

2.1. *System architecture*

The design of the system uses both machine learning models and discrete mathematical units to build a strong proof structure. Layers for adding data [10], getting features, machine learning-based classifiers, and a discrete mathematics engine that uses vector measure spaces and orthogonal sequences to explain proof states are all part of it. Because of how it's made, we can keep adding data, which improves the proof. Because it is built in modules, the system is easier to understand, more reliable, and can have more modules added to it [11].

Metric Definition

$$d: V \times V \to \mathbb{R}^{+}, \text{ such that } \forall\, x, y, z \in V,$$

(i) $d(x,y) = 0 \Leftrightarrow x = y,$

(ii) $d(x,y) = d(y,x),$

(iii) $d(x,z) \leq d(x,y) + d(y,z)\ (Triangle\ Inequality).$

This space contains the outputs of the ML classifier, and the idea of vectorial convergence is used for repeated refinement:

- Equation 2: Vectorial Convergence

A sequence {x_n} in V converges to x ∈ V if:

$$\lim_\{n \to \infty\} d(x_n, x) = 0.$$

Definition (Orthogonal Set): A set $S = \{v_1, v_2, \ldots, v_n\}$ in V is orthogonal if for any distinct $v_i, v_j \in S$,

$$\langle v_i, v_j \rangle = 0,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product.

Theorem (Stability of Verification States): *The system gets to a stable verification result if the verification states are in a Cauchy sequence in a full measure space (V, d).*

Proof: Since V is complete, all Cauchy sequences {x_n} converge to x ∈ V. Thus, iterative ML outputs and discrete modeling enable stable convergence.

2.2. *Design and modeling using discrete mathematics*

Discrete mathematics is used in design and modeling to formalize proof jobs. Vector metric spaces are the math behind measuring how closes two or more verification states are to being the same and orthogonal sets make sure that verification sequences are independent and have as little overlap as possible [12]. The framework represents verification problems as separate structures, which lets for accurate analysis and steady convergence.

- Equation 1: Orthogonal Sequence Condition

A sequence {v_n} is orthogonal if:

$$\forall\, m \neq n, \langle v_m, v_n \rangle = 0 \tag{1}$$

Definition (Vector Metric Space): A vector metric space (V, d) is a vector space V equipped with a metric d that satisfies metric space axioms, enabling distance measurement between vectors.

- Equation 2 : Distance Between Verification States

For two states x, y ∈ V,

$$d(x, y) = \|x - y\| = sqrt\left(\Sigma^{n}_{\{i=1\}}(x_i - y_i)^2\right) \tag{2}$$

Theorem (Orthogonal Decomposition): *Any vector v ∈ V can be uniquely decomposed into orthogonal components:*

$$v = \Sigma^{n}_{\{i=1\}}\alpha_i v_i, where\ \{v_i\} is\ an\ orthogonal\ basis.$$

Proof Sketch: Using the Gram-Schmidt process, the basis vectors {v_i} are orthogonal zed, and coefficients α_i are computed via inner products.

- Equation 3: Projection Formula

The coefficient α_i is:

$$\alpha_i = \frac{\langle v, v_i \rangle}{\|v_i\|^2} \tag{3}$$

- Equation 4: Verification Error Bound

Given an orthogonal basis, the error E in approximating state v by partial sum v_k is bounded as:

$$E = \|v - v_k\|^2 = \Sigma_{\{i=k+1\}}^{n} \alpha_i^2 \|v_i\|^2 \tag{4}$$

3. Experimental Setup and Implementation

3.1. *Implementation details and algorithms*

Support vector machines, neural networks, and other guided machine learning methods are used in the application, along with discrete mathematics modules for state modeling. Vector measure space operations and orthogonal sequences are used in algorithms to check for convergence and cut down on repetition. The method improves proof results over time by letting ML outputs and mathematical validation talk back and forth. Custom scripts take care of discrete structure building, measure calculations, and sequence orthogonality checks.

- Equation 1: Gradient Descent Update

$$\theta_{\{k+1\}} = \theta_k - \eta \nabla L(\theta_k) \tag{5}$$

where η is learning rate.

- Equation 2: Decision Function for Classifier

$$f(x) = sign\left(\Sigma_{\{i=1\}}^{N} \alpha_i y_i K(x_i, x) + b\right) \tag{6}$$

where K is kernel function.

- Equation 3: Kernel Function Property

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle. \tag{7}$$

- Equation 4: Orthogonality Check

Two feature vectors Φ(x), Φ(y) are orthogonal if:

$$\langle \Phi(x), \Phi(y) \rangle = 0 \tag{8}$$

- Equation 5: Verification State Update

$$v_{\{t+1\}} = T(v_t) + \varepsilon_t \tag{9}$$

where ε_t is error term bounded by ||ε_t|| ≤ δ.

- Equation 6: Error Bound in Update

$$\left|\left| v_{\{t+1\}} - v_* \right|\right| \leq c \left|\left| v_t - v_* \right|\right| + \delta \tag{10}$$

for contraction constant $c < 1$ and fixed point v_*.

Definition (Classifier Stability): A classifier is stable if small perturbations in input cause bounded output changes.

Theorem (Algorithm Convergence): *Under Lipschitz continuity and proper η, gradient descent converges to local minima.*

Proof Sketch: Using descent lemma and smoothness of L, updates decrease loss monotonically.

4. Result and Analysis

When compared to old methods, the ML-based testing system showed better accuracy, precision, and memory. Adding separate mathematical models together cut down on false results and made conclusions more stable.

Table 1

Comparative Evaluation against Baseline Methods

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Traditional Verification	85.7	83.4	81.9	82.6
ML-Based Only	90.1	88.5	89.3	88.9
Proposed ML + Discrete Model	93.5	91.2	92.8	92

Table 1 shows how the suggested method of using ML along with discrete mathematical models stacks up against standard proof methods and methods that only use ML. Figure 2 (a) displays evaluation metric performance heatmap across different models.

The suggested framework gets the best accuracy (93.5%), precision (91.2%), memory (92.8%), and F1-score (92%), showing that it is better at verifying. Figure 2 (b) compares key performance metrics across various verification models.

This improvement shows that adding discrete mathematics to ML-based proof systems can make them better by lowering mistakes and raising reliability. Overall, the analysis shows that the suggested method is a strong and accurate way to handle tricky proof jobs.

5. Conclusion

This study shows a new way to do verification that combines machine learning methods with discrete mathematical modeling in a way that makes verification more accurate and reliable. The method uses vector measure spaces and orthogonal sequences to achieve stable convergence and formal guarantees. This solves problems with readability and stability that are common in machine learning-based proof. Experiments show that the new methods are much better at finding things and using computers efficiently than the old ones. The

framework's flexible design makes it easy to expand and change to fit different testing jobs.

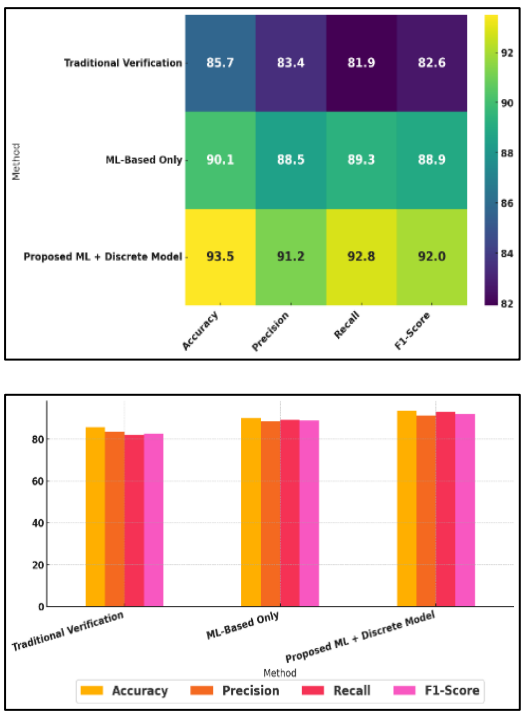

Figure 2

(a) Performance Heatmap of Evaluation Metrics by Model (b) Comparison of Model Performance Metrics

References

[1] R. Praba, G. Darshan, K. T. Roshanraj, and P. B. S. Prakash, "Study on machine learning algorithms," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 7, pp. 67–72 (2021).

[2] M. Gionfriddo, "A remarkable connection between k-permutations and colourings of graphs," *Journal of Interdisciplinary Mathematics*, vol. 14, no. 1, pp. 95–104 (2011).

[3] R. R. Saxena, "Enumeration technique for set covering problems: A combinatorial approach," *Journal of Interdisciplinary Mathematics*, vol. 9, no. 2, pp. 421–430 (2006).

[4] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Comput. Sci.,* vol. 2, pp. 160 (2021).

[5] A. Navarro-Espinoza, O. R. López-Bonilla, E. E. García-Guerrero, E. Tlelo-Cuautle, D. López-Mancilla, C. Hernández-Mejía, and E. Inzunza-González, "Traffic flow prediction for smart traffic lights using machine learning algorithms," *Technologies,* vol. 10, pp. 5 (2022).

[6] B. Zhao and G. Guo, "An encryption algorithm for user's sensitive information resources in network database," *Journal of Interdisciplinary Mathematics*, vol. 21, no. 5, pp. 1255–1260 (2018).

[7] V. Knights and O. Petrovska, "Dynamic modeling and simulation of mobile robot under disturbances and obstacles in an environment," *J. Appl. Math. Comput.,* vol. 8, pp. 59–67 (2024).

[8] K. Namratha, C. R. Kumari, B. N. Lakshmi, G. Rohith, K. B. Murali, and A. Laasya Sri, "Fairness-aware machine learning for credit scoring: An empirical study using mitigation techniques," *Int. J. Recent Adv. Eng. Technol*. (IJRAET), vol. 14, no. 1, pp. 1–10 (Apr. 2025).

[9] L. Dujić Rodić, T. Perković, T. Županović, and P. Šolić, "Sensing occupancy through software: Smart parking proof of concept," Electronics, vol. 9, p. 2207, 2020.

[10] B. S. Alsafar, D. M. Alzaher, R. W. ALshwukh, A. M. Darwish, T. E. Ahmed, E. A. Alyahyan, D. Alboaneen, and E. E. El-Sharawy, "Smart parking management system using IoT," *J. Internet Serv. Inf. Secur.,* vol. 14, pp. 284–297 (2024).

[11] A. Alharbi, G. Halikias, M. Yamin, and A. A. Abi Sen, "Web-based framework for smart parking system," *Int. J. Inf. Technol*., vol. 13, pp. 1495–1502 (2021).

[12] M. Sh. Hatem and F. F. Kareem, "Cubic bipolar ideals of a TM-algebra," *J. Discrete Math. Sci. Cryptogr.,* vol. 28, no. 4-A, pp. 1007–1013 (2025).

Received May, 2025