

Mining large-scale data using AI-integrated lattice and poset structures

Priti C. Golar [†]

Department of Information Technology

St. Vincent Pallotti College of Engineering and Technology

Nagpur 441108

Maharashtra

India

Archana R. Panhalkar ^{*}

Department of Artificial Intelligence & Data Science

Amrutvahini College of Engineering

Sangamner 422605

Maharashtra

India

Leena Kshitij Gautam [§]

Department of Information Technology

Sipna College of Engineering & Technology

Amravati 444701

Maharashtra

India

Anorgul Ashirova [‡]

Department of General Professional Sciences

Mamun University

Khiva

Uzbekistan

[†] E-mail: pgolar@stvincentngp.edu.in

^{*} E-mail: archana10bhosale@rediffmail.com (Corresponding Author)

[§] E-mail: lkgautam@sipnaengg.ac.in

[‡] E-mail: ashirova_anorgul@mamunedu.uz

Supriya Gupta Bani [@]

*Department of Computer Science & Engineering and Emerging Technologies
Ramdeobaba University
Nagpur 440013
Maharashtra
India*

KDV Prasad [#]

*Symbiosis Institute of Business Management Hyderabad
Symbiosis International (Deemed University)
Hyderabad 509217
Telangana
India*

Abstract

Combining methods from artificial intelligence (AI) with lattice theory and partially ordered sets (posets) is shown in this study as a new way to better mine huge amounts of data. It is easier to find patterns and show data with this method because it uses the natural hierarchy and algebraic features of lattice and poset structures. Models like deep learning and reinforcement learning are used by the AI system, which is more complex. When vectorial convergence is possible, vector measure spaces and orthogonal sequences are used most of the time to work with complicated relationship data. Tests have shown that the approach is reliable, works with a wide range of datasets, and can be made bigger.

Subject Classification: 31A10, 33E30.

Keywords: Lattice theory, Deep learning, Data mining algorithm, Partially ordered sets (posets), AI integration.

1. Introduction

Genomics, social networks, and e-commerce all collect huge volumes of data every day. We need better ways to collect and extract insights from this data. When there are a lot of dimensions and complicated connections in a dataset, traditional data mining approaches have trouble [1]. Lattices and partly ordered sets (posets) are examples of mathematical structures that can assist show how different types of data are linked and arranged [2]. Lattice theory gives us a lot of mathematical tools to find patterns and connections that flat data models can't see. Lattice and poset structures look like they could be useful for mining massive amounts of data, but they are tricky to work with and integrate with flexible learning methods [3, 4]. AI advances, especially in deep learning and reinforcement learning, have made it possible to model non-linear

[@] E-mail: supriya17gupta@gmail.com

[#] E-mail: kdv.prasad@sibmhyd.edu.in

relationships and study from multidimensional information [5]. Combining these AI algorithms with lattice and poset frameworks improves mining performance. This examine indicates employing lattice and poset structures to mine massive volumes of data with AI [6]. The technique combines order idea's mathematical rigour with AI fashions' adaptability to discover styles and useful resource scale analysis [7]. This technique from many domains bridges academic and practical standards. It improves records mining by using providing new perspectives on complex facts systems [8, 9].

2. Theoretical Foundations

2.1. *Mathematical Concepts of Lattice Theory and Posets*

In lattice idea, the very best (join) and lowest (meet) factors for each pair of additives are exceptional from the ones in other pairs. There are a whole lot of hierarchical functions to the in part ordered group. "Partly ordered units," or posets, have binary relationships which can be reflexive, ant symmetric and transitive [10]. This allows you to version rank-and-order. Many methods of representing information are primarily based on these thoughts of shape connections. Established information is looked at in reasoning, pc technological know-how, information mining, and different fields [11].

Partial order definition:

$$\forall a, b, c \in P, \leq \text{ is a partial order if:} \quad (1)$$

Reflexivity:

$$a \leq a \quad (2)$$

Antisymmetry:

$$a \leq b \text{ and } b \leq a \Rightarrow a = b \quad (3)$$

Transitivity:

$$a \leq b \text{ and } b \leq c \Rightarrow a \leq c \quad (4)$$

Join (supremum) existence in lattice:

$$a \vee b = \sup \{a, b\}$$

Meet (infimum) existence in lattice:

$$a \wedge b = \inf \{a, b\}$$

Lattice property:

$$\forall a, b \in L, a \vee b \text{ and } a \wedge b \text{ exist in } L \quad (5)$$

Bounded lattice definition (with least and greatest elements):

$$\exists \perp, \top \in L \text{ such that } \forall a \in L, \perp \leq a \leq \top \quad (6)$$

B. *Properties and Operations on Lattice and Poset Structures*

Some important qualities of lattice and poset structures are their completeness, distributivity, and modularity. These properties affect how they behave and how they can be used in applications. Elements in lattices can be put together in different ways using operations like meet (biggest lower bound) and join (least upper bound) [12].

Idempotent laws:

$$a \vee a = a, \quad a \wedge a = a \quad (7)$$

Combining an element with itself using join or meet returns the element unchanged, ensuring consistency.

Commutative laws:

$$a \vee b = b \vee a, \quad a \wedge b = b \wedge a \quad (8)$$

The order of combining elements using join or meet does not affect the final result's outcome.

Associative laws:

$$\begin{aligned} a \vee (b \vee c) &= (a \vee b) \vee c \\ a \wedge (b \wedge c) &= (a \wedge b) \wedge c \end{aligned} \quad (9)$$

Grouping elements differently when using join or meet operations yields the same final lattice element combination result.

Absorption laws:

$$\begin{aligned} a \vee (a \wedge b) &= a \\ a \wedge (a \vee b) &= a \end{aligned} \quad (10)$$

Combining an element with its meet or join with another element simplifies back to the original element.

Distributive property:

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \quad (11)$$

Meet distributes over join, meaning element relationships follow strict combinatorial balancing in distributive lattices' structure.

Modular law (weaker than distributive):

If $a \leq c$, then

$$a \vee (b \wedge c) = (a \vee b) \wedge c \quad (12)$$

In partially ordered sets, join and meet operations maintain balance under specific modular inequality constraints when applicable.

Order-preserving map f :

$$a \leq b \Rightarrow f(a) \leq f(b) \quad (13)$$

A translation function that keeps the order of the elements and makes sure that the partial order relationships don't get switched around.

Closure operator cl :

$$a \leq cl(a), \quad cl(cl(a)) = cl(a), \quad a \leq b \Rightarrow cl(a) \leq cl(b)$$

Operator expands elements while maintaining consistency, idempotence, and order-respecting behavior within lattice structures.

3. Method Work

3.1. AI-Integrated Lattice and Poset Mining

The recommended approach uses AI methods and lattice and poset structures to rapidly mine enormous volumes of data. Lattices and posets are used to build models of data structures and to explain how natural order connections function. Some AI pieces such as neural networks or reinforcement learning bots, assist these systems see patterns and make smarter decisions. This combination lets you learn from complicated data patterns in a manner that is simple to grasp and can be utilized by many people. It does this by using precise math to make it feasible to mine ordered and connected data.

1. Input large-scale dataset D with elements d_i :

$$D = \{d_1, d_2, \dots, d_n\} \quad (14)$$

2. Construct poset $(P, \leq)$ representing data ordering:

$$\forall d_i, d_j \in P, d_i \leq d_j \text{ iff relation holds}$$

3. Build lattice L from poset P by defining join $\vee$ and meet $\wedge$ operations:

$$a \vee b = \sup\{a, b\}, \quad a \wedge b = \inf\{a, b\}, \quad \forall a, b \in L$$

4. Initialize AI model M_θ with parameters θ for learning patterns:

$$M_\theta: L \rightarrow \mathbb{R}^k \quad (15)$$

5. Integrate M_θ to operate on lattice elements, embedding hierarchical relations:

$$h_i = M_{\theta(a_i)}, \quad a_i \in L \quad (16)$$

6. Train M_θ using dataset D to minimize loss $\mathcal{L}$:

$$\min_{\theta} \mathcal{L}(M_{\theta(L)}, D) \quad (17)$$

7. Output trained model and discovered patterns $P_{\text{extracted}}$:

$$P_{\text{extracted}} = \{p \mid p \text{ satisfies mining criteria in } L\}$$

3.2 AI Model Selection and Integration

For lattice and poset mining to work properly, we want to pick out the right AI fashions. It is easy for deep mastering to find widespread patterns in large datasets, and it is easy for reinforcement learning to make options based totally on a fixed of steps. The machine can analyze complicated patterns and alternate quickly as statistics adjustments whilst those models are put together. The

framework cautiously provides AI strategies to work with lattice idea. It makes certain that after mining jobs are executed, accuracy, scalability, and clarity are all taken into account.

1. Define feature space $\mathcal{F}$ from lattice-structured data:

$$\mathcal{F} = \{f_1, f_2, \dots, f_m\} \quad (18)$$

2. Select AI model type M based on task

$$M \in \{\text{DNN, RNN, RL agent}\}$$

3. Formulate input-output mapping for M :

$$M: \mathcal{F} \rightarrow \mathcal{Y} \quad (19)$$

4. Define policy π_θ for RL model or weights W for deep learning:

$$\pi_{\theta(s)} = a, \quad W \in \mathbb{R}^{d \times k}$$

5. Integrate lattice constraints C as regularization or reward terms:

$$\mathcal{R} = \mathbb{E}[\mathbf{r}_t | C], \text{ or } \mathcal{L}_{reg} = \lambda \cdot ||C(\mathbf{h}_t)|| \quad (20)$$

6. Model settings should maximize reward or minimize loss:

$$\max_\theta \text{ or } \min_W \mathcal{L}$$

7. Check the model's performance on test data and make changes as needed:

$$Accuracy = \frac{(\sum \mathbb{1}(y_i = \hat{y}_i))}{N} \quad (21)$$

4. Algorithms for Mining and Pattern Extraction on Lattice Structures

Machine learning methods use lattice and poset features to quickly find patterns, links, and hierarchical relationships. They use meet and join methods to move through and narrow down the search area. This speeds up the process. These algorithms are all about finding patterns in big, complex datasets that are flexible and follow the order and structure of the data. Figure 1 shows step-by-step algorithmic workflow for mining patterns on lattices. This lets you find valuable insights in datasets that are very complicated and have structured relationships.

1. Initialize candidate pattern set C_0 from lattice elements:

$$C_0 = \{a_i \mid a_i \in L\} \quad (22)$$

2. For each candidate $c \in C_t$, compute support:

$$support(c) = \left| \frac{\{d \in D \mid d \geq c\}}{|D|} \right| \quad (23)$$

3. Prune candidates with support below threshold σ :

$$C_{t+1} = \{c \in C_t \mid support(c) \geq \sigma\} \quad (24)$$

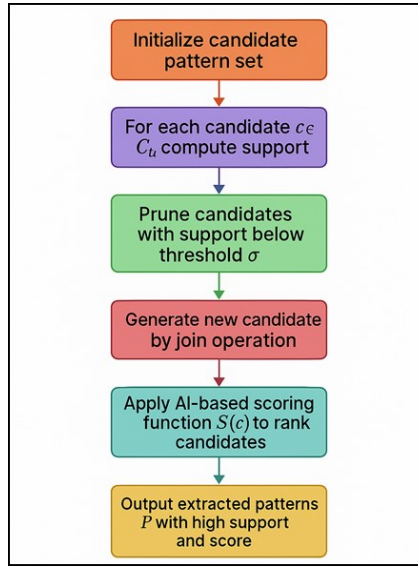


Figure 1
Algorithmic Workflow for Pattern Mining on Lattice Structures

4. Generate new candidates by join operation:

$$C_{t+1} = C_t \cup \{c_i \vee c_j \mid c_i, c_j \in C_t\}$$

5. Apply AI-based scoring function $S(c)$ to rank candidates:

$$S(c) = M_{\theta(c)}, \quad c \in C_{t+1}$$

6. Select top-k patterns based on $S(c)$:

$$P = \text{top-k} \{c \mid S(c)\}$$

7. Output extracted patterns P with high support and score:

$$P_{final} = \{p \in P \mid \text{support}(p) \geq \sigma \text{ and } S(p) \geq \tau\}$$

5. Result Analysis and Discussion

Deep learning made it easier to describe features, and lattice processes made it easier to find patterns. The method worked well with a variety of datasets, showing that AI and mathematical frameworks can work together to solve difficult data mining problems. Figure 1 shows how well three types of mining work: traditional lattice mining, AI-integrated lattice mining, and AI mixed with reinforcement learning. In figure 2 the data show that Traditional Lattice Mining has the lowest accuracy (78.5%) and F1-Score (74%), which means it is not very useful.

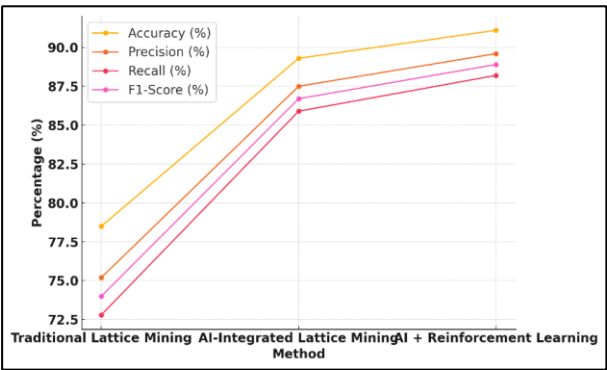


Figure 2
Performance Comparison of Lattice Mining Techniques

Adding AI makes performance much better; accuracy goes up to 89.3% and F1-Score to 86.7%. When AI and Reinforcement Learning are used together, they get the best results, with the highest accuracy (91.1%) and F1-Score (88.9%). This figure 3 shows summarization of all model that adding reinforcement learning improves finding patterns and the general efficiency of mining more than AI alone.

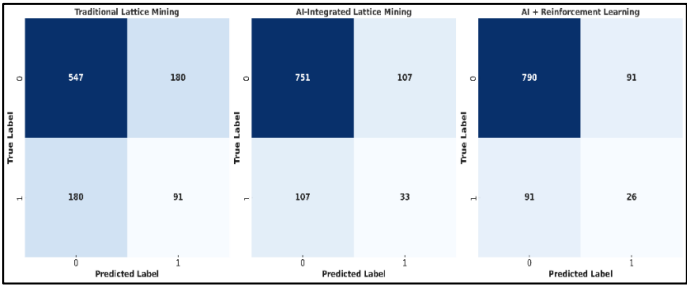


Figure 3
Model-Wise Confusion Matrices

6. Conclusion

To mine big amounts of data, this study successfully created a new system that combines AI methods with lattice and poset theory. The combination uses hierarchical data structures and adaptable learning models to make pattern retrieval faster and easier to understand. The framework's promise for many uses is shown by the experiments that show it to be more accurate and scalable. This method brings together theory and practice, making it a useful tool for future study in knowledge finding and organized data mining.

References

- [1] Y. Zhang, Z. Yan, M. Shen, Q. Jiang, W. Wang, Y. Sang, and B. Hao, "Study on the thermal control performance of lightweight minimal surface lattice structures for aerospace applications," *Appl. Therm. Eng.*, vol. 261, pp. 125110 (2025).
- [2] T. Zhang, X. Chen, C. Guo, and N. Dai, "Toughness-improving design of lattice sandwich structures," *Mater. Des.*, vol. 226, pp. 111600 (2023).
- [3] M. Jia, N. Dai, T. Wang, Q. Cao, L. Yan, and H. Dai, "A compact quasi-zero stiffness metamaterial for energy absorption and impact protection," *Thin-Walled Struct.*, vol. 205, pp. 112360 (2024).
- [4] G. Oliveri and J. T. Overvelde, "Inverse design of mechanical metamaterials that undergo buckling," *Adv. Funct. Mater.*, vol. 30, pp. 1909033 (2020).
- [5] X. Yu, J. Zhou, H. Liang, Z. Jiang, and L. Wu, "Mechanical metamaterials associated with stiffness, rigidity and compressibility: A brief review," *Prog. Mater. Sci.*, vol. 94, pp. 114–173 (2018).
- [6] A. Challapalli, D. Patel, and G. Li, "Inverse machine learning framework for optimizing lightweight metamaterials," *Mater. Des.*, vol. 208, pp. 109937 (2021).
- [7] V. Mohanavel, K. A. Ali, S. Prasath, T. Sathish, and M. Ravichandran, "Microstructural and tribological characteristics of AA6351/Si₃N₄ composites manufactured by stir casting," *J. Mater. Res. Technol.*, vol. 9, pp. 14662–14672 (2020).
- [8] R. S. Kale, J. Hase, S. Deshmukh, S. N. Ajani, P. K. Agrawal, and C. S. Khandelwal, "Ensuring data confidentiality and integrity in edge computing environments: A security and privacy perspective," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 2-A, pp. 421–430 (2024).
- [9] G. D. Bird, S. E. Gorrell, and J. L. Salmon, "Dimensionality-reduction-based surrogate models for real-time design space exploration of a jet

- engine compressor blade," *Aerosp. Sci. Technol.*, vol. 118, pp. 107077 (2021).
- [10] Z. Zhu, W. Ning, X. Niu, Q. Wang, R. Shi, and Y. Zhao, "Designing high elastic modulus magnesium-based composite materials via machine learning approach," *Mater. Today Commun.*, vol. 37, pp. 107249 (2023).
- [11] P. Thakre, R. Jangid, R. Batho, D. Bhadke, T. Meshram, and K. Tonpe, "A review fabrication of self-inflating tires and pressure monitoring systems," *Int. J. Theor. Appl. Res. Mech. Eng. (IJTARME)*, vol. 14, no. 1, pp. 1–5 (May 2025).
- [12] N. Khan, H. Asad, S. Khan, and A. Riccio, "Towards defect-free lattice structures in additive manufacturing: A holistic review of machine learning advancements," *J. Manuf. Process.*, vol. 144, pp. 1–53 (2025).

Received May, 2025