Reducing the Size of the Lattice using K-means

Shaurya Pratap Singh[1], Brijesh Kumar Chaurasia[2], Ayush Pal[3], Siddharth Gupta[4], Tanmay Tripathi[5]

[1-5] Department of Computer Science and Engineering,
[1-5] Pranveer Singh Institute of Technology, Kanpur, INDIA
shauryapratap2114@gmail.com

Abstract. The most powerful and adaptable subfield of post-quantum encryption has come to be known as lattice-based cryptography. Traditional cryptographic systems, including RSA and DSA, are predicated on ideas like the supposed intractability of the discrete and prime number difficulties with logarithms. However, these presumptions are threatened by the introduction of quantum computing. To overcome this obstacle, lattice-based as a reliable solution, cryptography has grown in popularity. In this cryptographic paradigm, one of the major obstacles is the problem of the shortest vector (SVP). This work focuses on addressing approaches to lattice issues, especially in two-dimensional up to ten-dimensional spaces. In order to do this, we use the K-means machine learning (ML) technique. In this paper, findings and analyses show that the strategy we've suggested can reach an accuracy of 60% on datasets that we have prepared ourselves. This study offers a contribution to enhancing lattice-based cryptography's security against emerging threats, especially in the quantum computing context.

Keywords: Shortest Vector Problem (SVP), Rivest-Shamir-Adleman (RSA), Digital Signature Algorithm (DSA), K-Means Algorithm, Post-Quantum Cryptography (PQC).

1. Introduction

Lattices, a branch of mathematics used in computer science, are used to develop novel encryption techniques by resolving problems like the shortest vector problem (SVP) and closest vector problem (CVP). In the framework of Euclidean space, a lattice is a regular grouping of points [1]. They can perform a wide range of mathematical and computer processes, such as solving integer programming problems, the Diophantine approximation and estimation, cryptanalysis, creating integer-based programming problems, creating error-correcting canons for multiple antenna systems, and many more. Recently, buildings have drawn significant attention as well [2]. The concept of concept decomposition (CD) allows for the application of cluster analysis as a data reduction technique. In order to reduce the size of the term-document matrices in IR applications, fuzzy K-means (FKM) clustering has been shown to be an effective substitute for SVD due to its decreased computational complexity [3]. Motivated by the study in [4], [1], and in this paper, we propose using K-Means to reduce the size of the lattice.

The organization of this work is as follows: Section 2 defines the lattice and concept lattices. Section 3 explains SVP with its hardness. In Section 4, we present K-means' working process. A proposed methodology is discussed in Section 5. Result analysis is discussed in Section 6. In Section 7, we present a conclusion with future scope.

2. Lattice

A Lattice is a discrete additive subgroup of R^n, which means it is a subset of $A \subseteq R$ that satisfies all the criteria stated below.

Example 1: Because Q_n is not discrete, it is a subset of R^n rather than a lattice. A set of all n-dimensional vectors having integer components is the most fundamental form of lattice (Fig. 1).

Example 2: The integer vectors in the set Z^n may be added and subtracted, and it is obvious that there must be an absolute minimum distance of one between any two integer vectors. Z^n can be changed using a (nonsingular) linear conversion to produce additional lattices [5].

For instance, $B(Z^n) = \{Bx: x \in Z^n\}$ is also a lattice if $\{B \in R^{k \times n}\}$as complete column ranking (*i.e.*, the columns of B are linearly independent). It is obvious that addition and subtraction conclude this set [6].

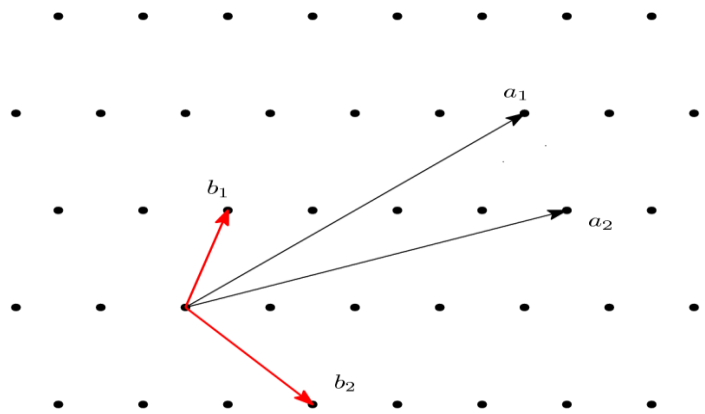

Fig.1: Two Dimensional Lattices

3. Shortest Vector Problem (SVP)

The SVP is focused on locating the lattice $\mathscr{L}$s shortest non-zero vector [7], as shown in Fig. 2. In order to preserve generality, we restrict the exact form of SVP to integer lattices (and thus integral bases) [8]. The exact form of SVP has three frequent variants:

- Determine if the instances $\lambda_1(\mathscr{L}(B)) \leq d$ and $\lambda_1(\mathscr{L}(B)) > d$ exist given a lattice basis B and a real $d > 0$.
- Calculation: Locate $\lambda_1(\mathscr{L}(B))$ given a lattice basis B.
- Look over a (nonzero) $v \in \mathcal{L}(B)$ such that $||v|| = \lambda_1(\mathscr{L}(B))$ given a lattice basis B.

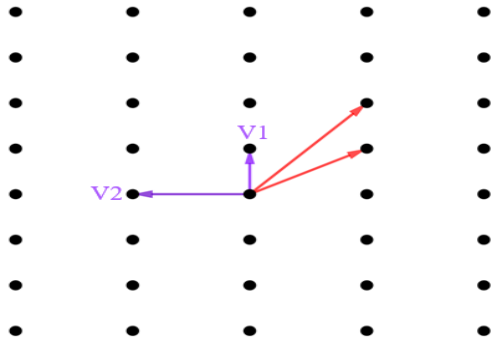

Fig. 2: Shortest Vector v_1 in $\mathscr{L}$

***Hardness of* SVP**

GapSVP's objective is to ascertain whether the length of such a vector exceeds or falls short of a predetermined limit. On a lattice, the SVP is the most significant statistical challenge. The aim of this puzzle is to identify the shortest non-zero vector in the input lattice. We clearly state the graphical representation of a lattice and swiftly define these

problems. Fig. 3(a) shows complexity of γ - GapSVP on $\mathscr{L}$ [9]. For $O(1)$, the task is NP-hard (under randomized reductions), while for = 2^n, it can be solved in polynomial time. Lattice-based cryptography is demonstrably safe if it is difficult for = n^c for sufficiently big constants $c > 0$. Fig. 3(b) shows that in green box is labeled for algorithm's complexity *i.e.* $2^{nlog\log n/\log n}$. However, it is solvable in polynomial time.

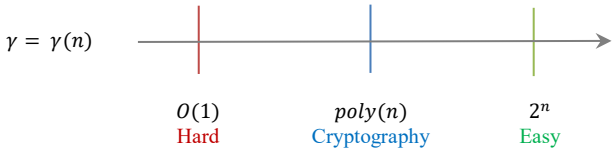

Fig. 3 (a): Complexity of γ - GapSVP on $\mathscr{L}$

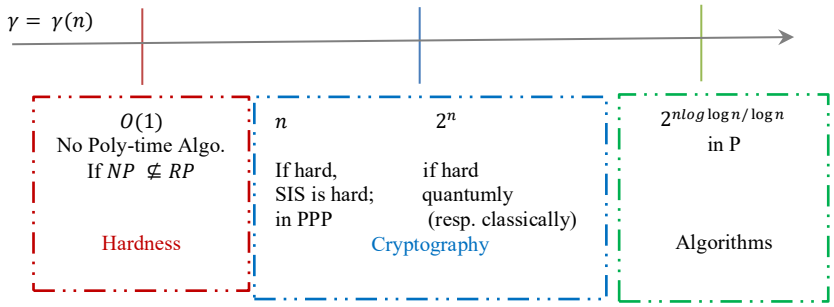

Fig. 3 (b): Complexity of γ - GapSVP on $\mathscr{L}$ dimension *n*

4. K-Means Algorithm

Unsupervised ML technique K-Means Clustering splits the unlabeled dataset into various clusters. In order to ensure that the data points within each group are more comparable to one another than the data points within the other groups, clustering aims to divide the population into a number of groups. In a nutshell, it's a collection of items with both similarities and differences [10]. One of the most popular and scale-constructive algorithms is K-means, which has a $O(n)$ complexity. Since K-means uses a centroid-based clustering method, each data point is assigned to a cluster based on how far it is from a centroid. The crucial step is determining the K count of clusters in the collection of data [11]. Data points are organized progressively according to their aspect. Each data point is iteratively assigned to the correct cluster. The objective is to reduce the total of the distances between the data points to the cluster centroid in order to identify the cluster that each data point must belong to [12] then, we divide a data set into K clusters and assign a mean value to each one.

The clusters that have data points that are closest to their average are selected. A number of distance characteristics are offered in order to calculate the distance. Fig. 4 shows the clusterization before applying K-Means and after applying K-Means algorithm.

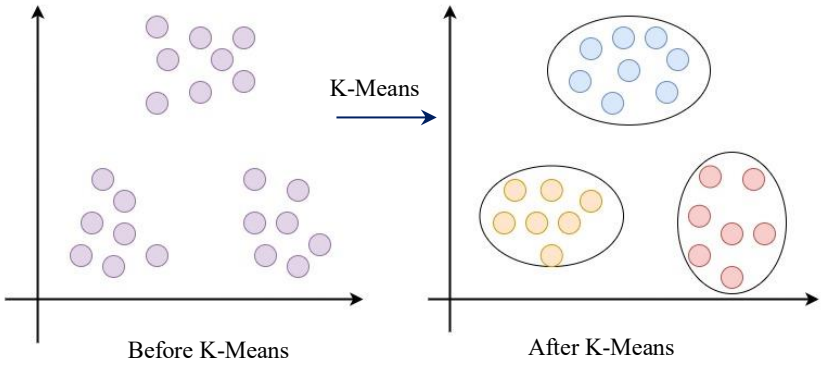

Fig. 4. Clustering of Data

4.1 Flow-chart of K-means Algorithm

The flowchart as shown in Fig. 5 is for k-means clustering, an unsupervised machine learning algorithm for grouping similar data points together. It works by iteratively assigning data points to the cluster with the nearest centroid (mean). The centroids are updated after each iteration to reflect the new cluster assignments. The algorithm terminates when the cluster assignments no longer change.

K-means clustering is a simple but effective clustering algorithm. It is easy to implement and can be used to cluster data of any dimensionality. However, it is important to note that k-means clustering is sensitive to the initialization of the centroids. If the centroids are not initialized properly, the algorithm may converge to a local optimum, which is not the global optimum. The detailed explanation of the flowchart is as:

1. **Start:** The algorithm starts by initializing the centroids of the clusters. This can be done randomly or using a heuristic method.
2. **Selection of Centroids:** The algorithm then assigns each data point to the cluster with the nearest centroid.
3. **Calculation of distance:** The algorithm then calculates the distance between each data point and the centroid of its assigned cluster.
4. **Minimization of Distance:** The algorithm then checks if the distance between the data point and the centroid of its assigned cluster is the minimum distance between the data point and all centroids. If so, the algorithm moves to the next data point. Otherwise, the algorithm moves the data point to the cluster with the nearest centroid.

5. **Move to the same cluster:** If the algorithm has moved a data point to a new cluster, it updates the centroid of the new cluster to reflect the new cluster assignment.
6. **Repeat the process from step 2:** The algorithm then repeats steps 2-5 until the cluster assignments no longer change.
7. **Stop:** The algorithm terminates when the cluster assignments no longer change.

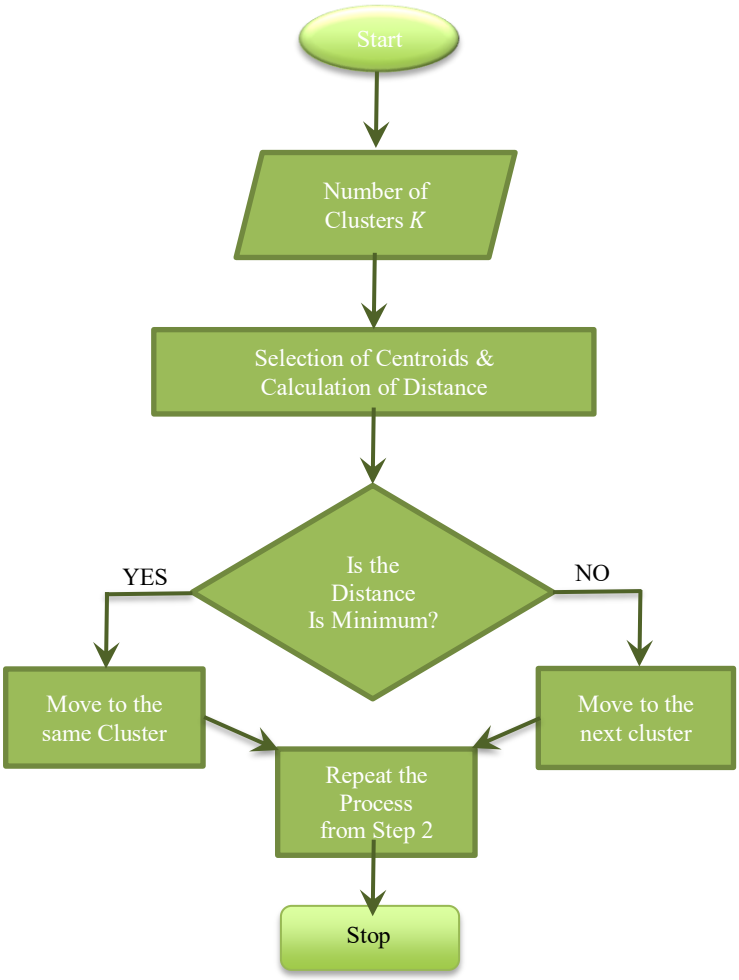

Fig. 5. Flowchart of K-Means Algorithm

4.2 Algorithm and Working of K-means

This sub-section describes the functional process of the suggested method for using machine learning as well as K-means to target SVMs. We present a clustering approach for datasets. The dataset is self-generated with 350 points to evaluate the efficacy of the proposed algorithm.

4.2.1 *Deciding how many clusters to use*

Choosing K, the total number of clusters in which the data is going to be arranged, is the initial step. K = 3 will be used in this case.

4.2.2 *Setting up centroids*

The centroid stands for a cluster's focal point. However, since the exact centre of the data points is initially unknown, we choose certain data points at random to serve as the very first centroids for each of the clusters. Fig. 6 (a) initialize three centroids for our dataset.

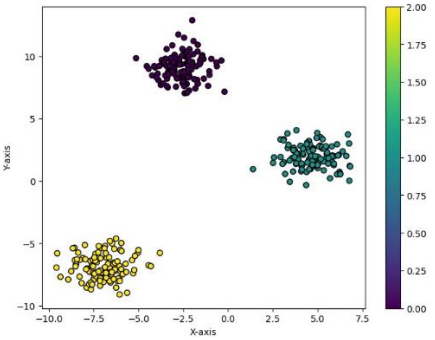

Fig. 6(a). Clustering dataset with K=3

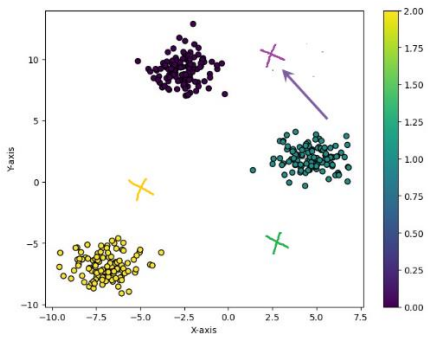

Fig. 6(b). Closest Clustering dataset with K=3

4.2.3 *Align the closest cluster with the data points*

After initializing the centroids, the next step is to connect data points X_n with the nearest cluster centroid C_k as illustrated by Fig. 6(b). The first objective in this phase is to use the Euclidean distance metrics to calculate the distance between data point X and the centroid.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \tag{1}$$

Choose the data point cluster in which the distance between the data point and the centroid is the shortest.

4.2.4 *Re- Initialize centroids*
By averaging all the data points in that cluster, the centroids will then be reset.

$$C_i = \frac{1}{|N_i|}\sum X_i \tag{2}$$

4.2.5. *Repeat steps three and four*

Phases 3 and 4 will continue to be performed until we obtain the best centroids and reliable data point assignments to the appropriate clusters as represented in Fig. 7.

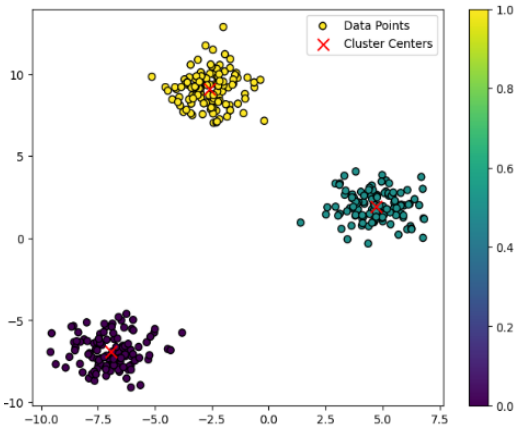

Fig. 7 Clustering dataset with K=3

5. Proposed Methodology

This section presents a K-Mean-based attack on SVP. The SVP is described as a large-size, multi-dimensional lattice with the ability to extend in n dimensions. Finding the clusters in a big lattice is extremely significant. There is currently no algorithm that can solve this SVP problem in polynomial time. This is the first attempt in this approach, to the best of my knowledge. On a self-generated dataset, the assault was located using lattice attributes.

- **Vector Lattice Property:** Addressing the vector lattice challenge involves a focus on the lattice's inherent vector nature. Lattice vectors can be expressed through coordinate representations, providing a comprehensive understanding of their structural properties. Multidimensional lattice is depicted in Fig. 8. [14] provides a thorough description of the 2-$\mathcal{D}$ honeycomb lattice constructed from the vertices of the regular triangles spanning the plane.

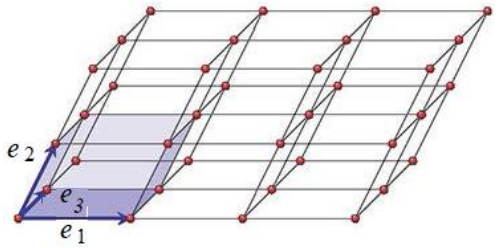

Fig. 8: Multi-dimensional Lattice where E1, E2, E3 are vectors [13]

- **Cryptographic Lattice Size**: As detailed in the 'Basis' section, the finite memory capacity of computers underscores the impracticality of employing an infinitely large lattice. Therefore, the dimensions of the lattice are tailored to the specific capacity constraints of the system in use.

- **K-Mean Algorithm Approach for SVP:** The K-Mean algorithm operates on a $2\mathcal{D}$ or $4\mathcal{D}$ dataset. By inputting the dataset values, the algorithm calculates the silhouette score, enabling the determination of the optimal number of clusters. Subsequently, the algorithm generates clusters based on this determined count, facilitating the creation of a heat-map that visually represents the dataset values.

The following is a full summary of the suggested methodology's steps: As seen in Fig. 9, the procedure begins by providing the entrance points (X and Y) and the number of groups (K). The K-group centers will be chosen in the first step. After that, the procedure moves into an iterative phase that entails processing (such as matching each data point to the closest centroid) and updating (recalculating cluster centroids). A check is done to see how much the cluster's mean has changed after each iteration. The program then goes back to the allocation phase in this situation. If not, the algorithm moves on to the last output stage. Each data point's cluster function and the coordinates of the cluster center are provided in the output's final form.

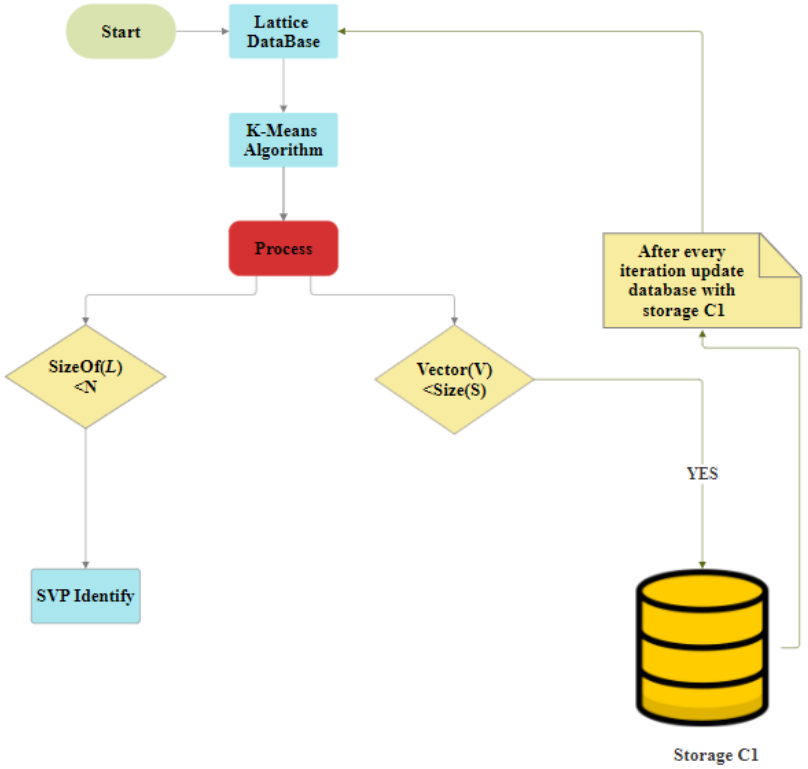

Fig. 9. Working Process of Proposed Methodology

6. Results and Analysis

The way the K-Means algorithm works Specifies the number of clusters K. Initialize centroids by first mixing the dataset and then randomly selecting K data points for the centroids without any change or replacement. Keep iterating until there is no change to the centroids. The experimental evaluation of attacks and comparison among 2 $\mathcal{D}$ and 4 $\mathcal{D}$, dataset respectively, is as follows:

6.1 Resize 2 $\mathcal{D}$ Lattice using KMean

The following is an outline of the analysis of the suggested approach across a 2 $\mathcal{D}$ lattice dataset:

6.1.1 Dataset of 2 dimensional Lattice

We have generated more than 350 pairs of coordinates to evaluate the efficacy of our proposed scheme. In this we implement K-Mean clustering as the no of cluster s are given by us and it implements the clusters graph which is given below and ac cording to the dataset values it calculate the silhouette score and the number of clu sters we draw it generates its heat-map as per the dataset value.

6.1.2 Graph Representation of Clusters Drawn in 2D Dataset

In Fig. 10, there are four graph show that clusters drawn in 2 $\mathcal{D}$ lattice dataset first one is for X_1 highest values is 695.4875 in X and lowest values vary between 250.48-279.97, same in Y shows that highest value is 695.48 and lowest value is vary between 250.48-279.97. The graph is for outcome it shows the value distribution after the execution of proposed methodology. The result shows that cluster sizes are 2 $\mathcal{D}$, 4 $\mathcal{D}$, 6 $\mathcal{D}$, and 8 $\mathcal{D}$ respectively. The accuracy of 8 $\mathcal{D}$ is more, however, at 2 $\mathcal{D}$ accuracy is less.

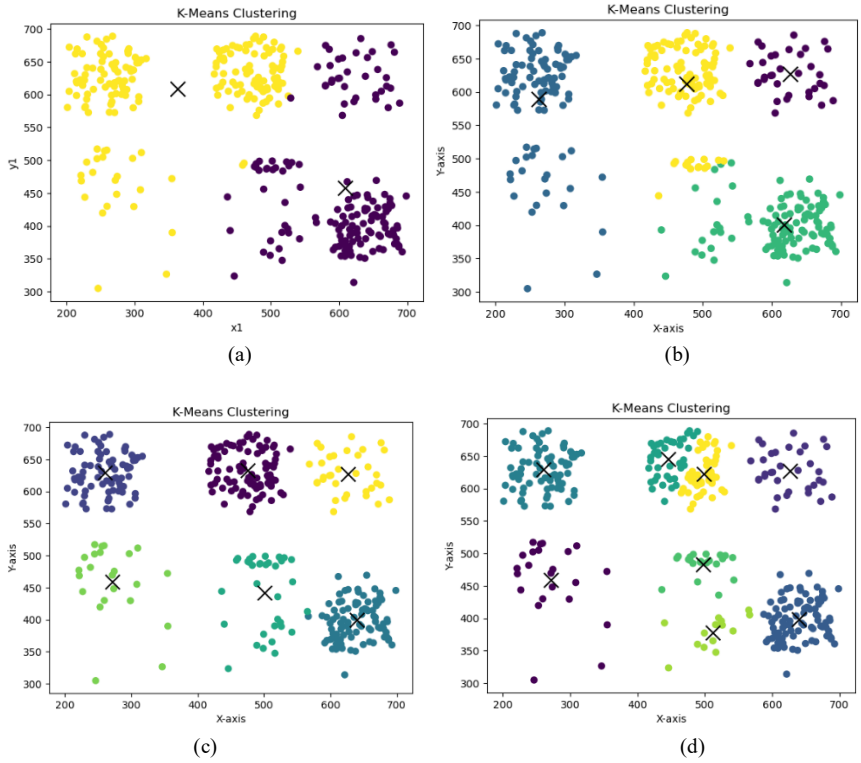

Fig. 10. Number of clusters drawn in 2D dataset

6.1.3 Heat Map representation of 2 $\mathcal{D}$ Lattice

It shown in this subsection in terms of data classification, a heat map, and performance using ML parameters. The inefficient correlations or interactions between variables in a dataset are depicted in Fig. 11. In this illustration, the color classes for 0 and 1 are orange and blue, respectively. The K-Mean algorithm, which

determines the clusters among the pertinent characteristics, is used to distribute the distinctive feature vector in various classes in the result. Using K-Mean and the best possible separation bounds, correlated feature points are shown as different orange and blue dots.

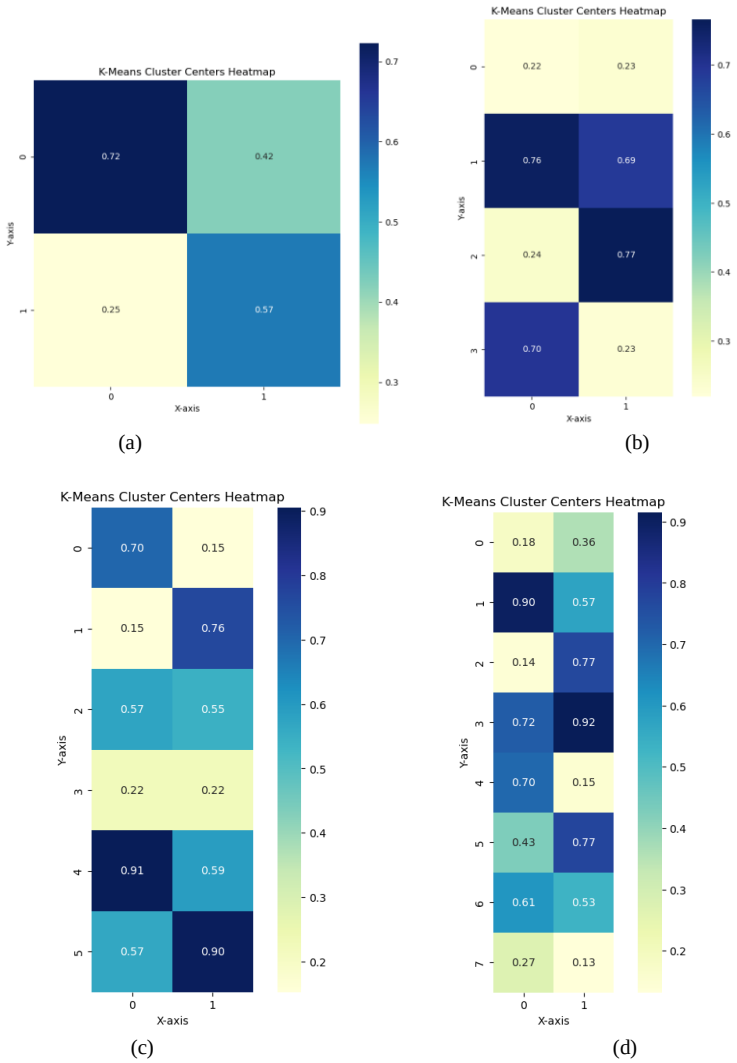

Fig.11. Number of clusters drawn in 2D dataset

The heat map shows the number of clusters that are drawn, and the different graph represents that different numbers of clusters are drawn, and each cluster has its own accuracy. According to this, the heat map changes its parameters, and its figure changes

at different numbers of clusters. The heat map cannot be the same, and the result can't be the same as compared to the result before we get it.

Fig. 12 shows that for different values of k, we get different values according to the dataset, and this is the result for 2 $\mathcal{D}$ dataset values. It is observed that the accuracy of clustering or attack over SVP at K = 6 is high and at K = 2 is low. It is observed that the possibility of attack due to reducing lattice size and increasing the accuracy of clustering over 6 $\mathcal{D}$ is high up to 60.7%.

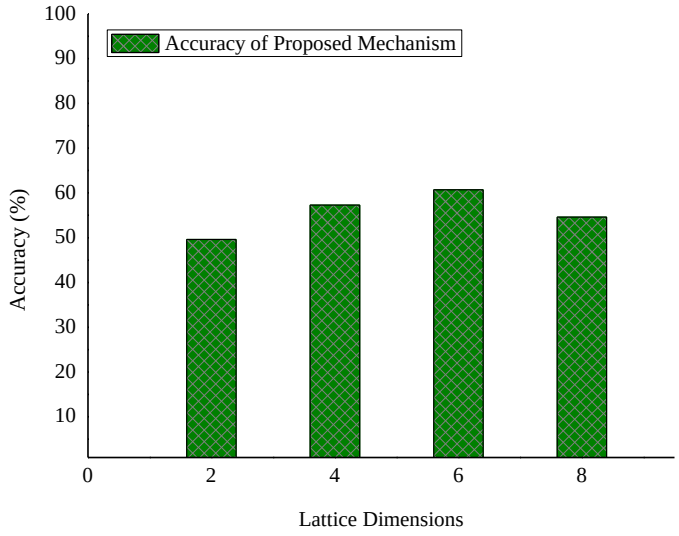

Fig. 12: Performance of proposed K-Mean scheme over 2 $\mathcal{D}$ Lattice Dataset

6.2 Resize 4- D Lattice using KNN

The following is an outline of the analysis of the suggested approach across a 4 $\mathcal{D}$ lattice:

6.2.1 Dataset of 4 dimensional Lattice

We have generated more than 350 pairs of coordinates to evaluate the efficacy of our proposed scheme. In this, we implement K-Mean clustering as the number of clusters is given by us, and it implements the cluster graph, which is given below, and according to the dataset values, it calculates the silhouette score, and the number of clusters we draw generates its heat map as per the dataset value.

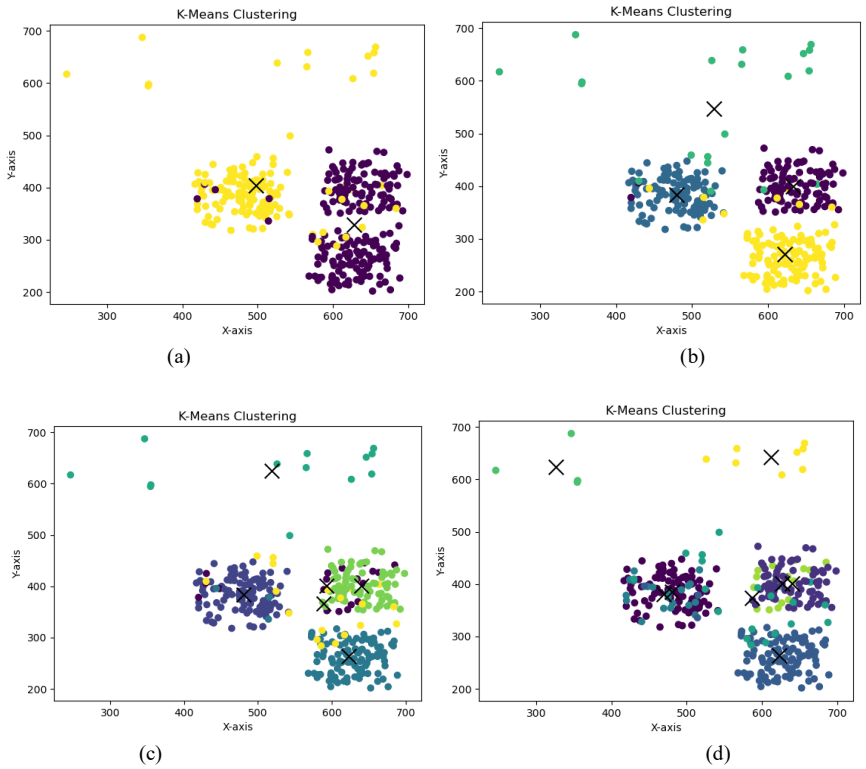

Fig. 13. Presentation of number of clusters drawn in 2 $\mathcal{D}$ dataset

In Fig. 13, there are nine graphs that show the variation of values in a 4 lattice dataset. The first one is for the highest values, which vary between 695.48 and the lowest values, which vary between 220.49 and 290.79. The same graph shows that the highest value is between 675.48 and the lowest value is between 280.97 and 600.76. The highest value is more than 695.4875, and the lowest value varies between 250.67 and 275.94. The highest value is more than 695.48, and the lowest value varies between 280.87 and 340.24.

6.2.2 Results of 4- $\mathcal{D}$ Lattice

In this section, results are represented in terms of classification of data, heat map, and performance based on ML parameters. It shows ineffective relationships or correlations between variables in a dataset. In Fig. 14, orange and blue color classes are considered for 0 and 1, respectively. The result shows the distribution of the distinctive feature vector in distinctive classes using the K-Mean algorithm that finds the distance metric among the relevant features. Correlated feature points are plotted in various orange and blue dots using K-means with optimal separation boundaries.

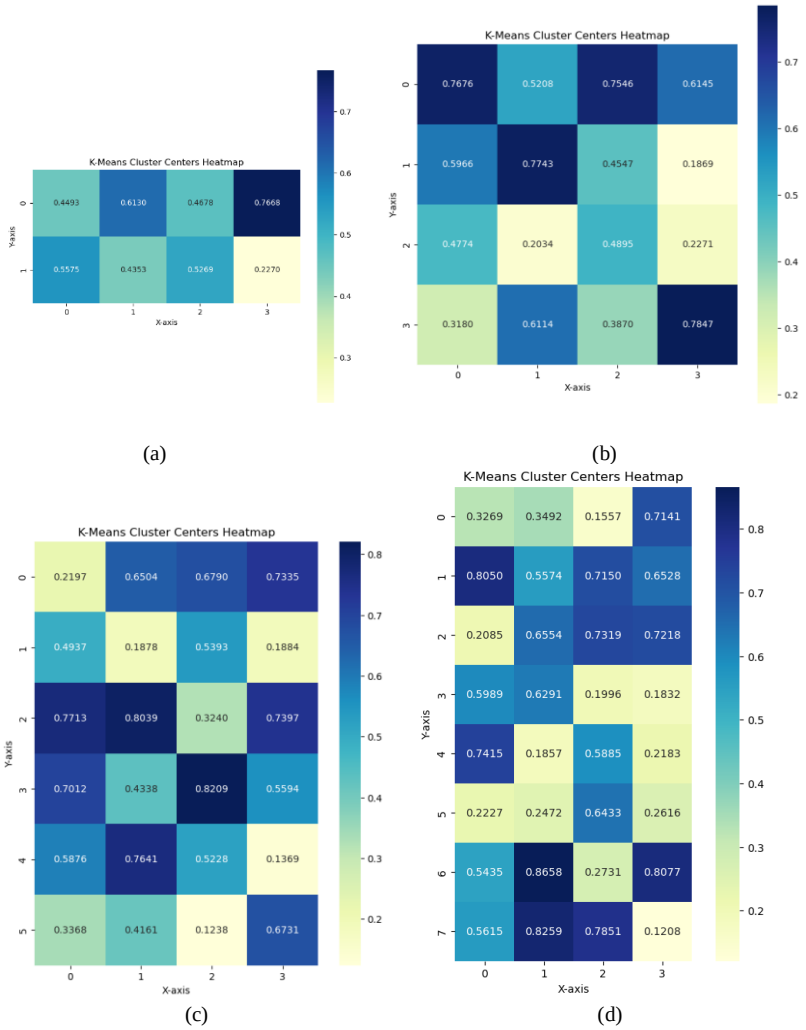

Fig. 14. Heat Map representation of 4 $\mathcal{D}$ Lattice Dataset

Fig. 15 shows that over 4 $\mathcal{D}$, reducing the size of lattice accuracy is high; however, at 2 $\mathcal{D}$, it is high when cluster size is 6. It is also observed that reducing the size of the lattice at cluster size 4 is the same over a 2- and 4-dimensional lattice. In addition, attack over four dimensions is hard and exponentially decreases accuracy at values of K up to 5 to 10. Moreover, over two-dimensional accuracy is ups and downs at values of K from 2 to 10.

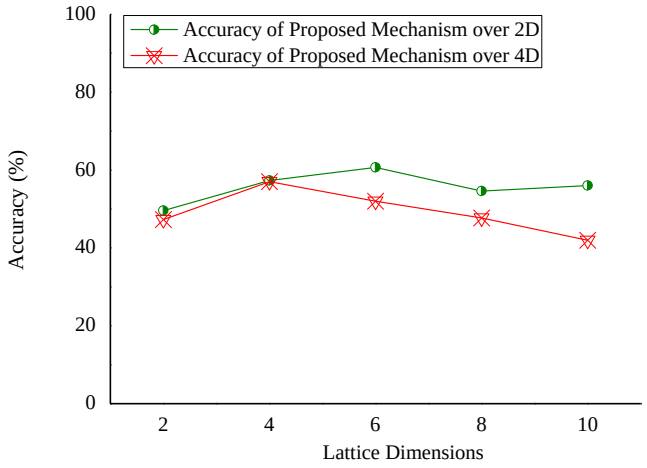

Fig.15. Comparative Analysis of proposed K-Mean scheme over 2 $\mathcal{D}$ and 4 $\mathcal{D}$ Lattice Dataset

7. Conclusion and Future Work

In this work, reducing the size of the lattice using K-means clustering is presented. This method has demonstrated its effectiveness in simplifying complicated structures, enabling more effective calculations, and improving the understanding of big datasets. However, over a large dataset, reducing the size of the lattice is hard. The proposed mechanism achieves an accuracy of up to 60%. In the future, we will explore the attack instead of clustering, and KNN will also be applied to explore the possibilities of hybrid approaches.

References

1. Kumar Ch. A. and Srinivas S. Concept lattice reduction using fuzzy K-Means clustering. In Expert Systems with Applications 37, 2696–2704 (2010).
2. Damien S. and Ron S. Making NTRU as secure as worst-case problems over ideal lattices. In EUROCRYPT: Advances in Cryptology – EUROCRYPT 2011, 27–47 (2011).
3. Dobsa J. and Basic, B.D. Comparison of information retrieval techniques: Latent semantic indexing and concept indexing. In Journal of Information and Organizational Sciences 28, 1–17 (2004).
4. Cheung, K.S.K. and Vogel, D. Complexity reduction in lattice based information retrieval. Information Retrieval 8, 285–299 (2005).
5. Forst, M. and Fukshansky, L. Counting Basis Extensions in a Lattice. In Mathematics Subject Classification, 1-15 (2010).
6. Fukshansky, L. and Shi, Y. Positive semigroups and generalized Frobenius numbers over totally real number fields. In Moscow Journal of Combinatorics and Number Theory 9(1), 29-41 (2020).
7. Ajtai, M. The shortest vector problem in l2 is np-hard for randomized reductions. In Proc. 13th Annu. ACM Symp. Theory Comput., 10–19 (1998).
8. Chuang, Yu-L., Fan, Chun-I., Tseng, Yi-F. An Efficient Algorithm for the Shortest Vector Problem. In IEEE Access 6, 61478 – 61487 (2018).
9. Bennett, H. The Complexity of the Shortest Vector Problem. In Electronic Colloquium on Computational Complexity 170, 1-25 (2022).
10. Boukhdhir, A., Lachiheb, O., and Gouider, M. S. An improved mapReduce design of kmeans for clustering very large datasets. 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), 1-6 (2015).
11. Fan, Y., Gongshen, L., Kui, M., Zhaoying, S. Neural Feedback Text Clustering With BiLSTM-CNN-Kmeans. In IEEE Access 6, 57460- 57469 (2018)
12. Park, H.S., Jun, C.H. A simple and fast algorithm for K-medoids clustering. In Expert Syst. Appl. 36, 3336–3341 (2009).
13. Daniele Micciancio Lattice Algorithms and Applications CSE 206A Winter 2010 UCSD CSE.
14. Mundhe, P., Yadav, VK., Verma, S. Venkatesan, S. Efficient Lattice-Based Ring Signature for Message Authentication in VANETs. In IEEE Systems journal, **14**(4), 5463 - 5474 (2020).
15. Helfrich, B. Algorithms to construct Minkowski reduced and hermite reduced lattice bases. In Theor. Comput. Sci., 41, 125–139(1985).
16. Hendrik, W. and Lenstra, Jr.: Lattices. Algorithmic Number Theory. MSRI Publications, 44, 2008
17. Chi, D.P., Choi, J. W., Kim, J.S., Kim, T. Lattice Based Cryptography for Beginners. Online available at: https://eprint.iacr.org/2015/938.pdf, Last accessed 23 March 2023.
18. Zhang, J. and Zhang, Z. Lattice-Based Cryptosystems-A Design Perspective.Springer Singapore, XIII, 174, 2020.

19. M. Ajtai. The Shortest Vector Problem is NP-Hard for Randomized Reductions, In Proc. 30th Annual ACM Symposium on Theory of Computing (STOC '98), 10–19 (1998).
20. Ajtai, M. Generating Hard Instances of the Short Basis Problem. In J. Wiedermann et al. (Eds.): ICALP'99, LNCS 1644, Springer-Verlag Berlin Heidelberg, 1−9 (1999).
21. Bennett, H., Peikert, C. Hardness of the (Approximate) Shortest Vector Problem: A Simple Proof via Reed-Solomon Codes. APPROX/RANDOM, 37:1-37 (2023).
22. Ajtai, M., Kumar, R., Sivakumar, D. A Sieve Algorithm for the Shortest Lattice Vector Problem. STOC'01, 610-610 (2001).
23. Micciancio, D. The Shortest Vector in a Lattice is Hard to Approximate to Within Some Constant. In SIAM Journal on Computing, 30(6), 1-26 (2001).
24. Aggarwal, D., Dadush, D. Regev, O. Solving the Shortest Vector Problem in 2^n Time via Discrete Gaussian Sampling. STOC'15, 733-742 (2015).
25. Sun, J., Liu, J., Zhao, L. Clustering algorithms Research. In *Journal of Software*, 19(1), 48-61 (2008).
26. Abdul Nazeer, K.A., Sebastian, M.P. Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. In *Proceeding of the World Congress on Engineering*, 1, 1-5 (2009).
27. Python online available at: https://www.python.org/, Last accessed on 23 March 2023.
28. Dataset, Online available at: https://techpubs.jurassic.nl/manuals/0530/ developer/Expl_MWG/sgi_html/ch03.html, Last accessed on 23 March 2023.