 ISSN 2393 - 9850
Pages :83-92

Rare Pattern Mining in Unsupervised Video Environments: A Study of K-means Variants

Mintu Movi
School of Computer Sciences, Mahatma Gandhi University, Kottayam 686560, Kerala, INDIA

Abdul Jabbar P
School of Computer Sciences, Mahatma Gandhi University, Kottayam 686560, Kerala, INDIA

Abstract

The use of CCTV systems for home surveillance generates a large amount of video data, which can make it difficult to identify rare but important patterns that could indicate potential threats. This research article focuses on detecting unusual patterns in CCTV footage, particularly those that are rare. It compares different advanced K-means clustering approaches, such as Fuzzy C-means (FCM), K-medoids (PAM), Mini-batch K-means, Bisecting K-means, Kernel K-means, and Weighted K-means. The study uses a diverse dataset of real-world CCTV footage from various home environments. By assessing how well these clustering techniques can detect and analyze rare patterns, the research aims to improve threat detection and provide insights into the most effective clustering methods for identifying rare patterns.

1 Introduction

Clustering is a fundamental technique in unsupervised learning and data mining(Amin et al., 2023), where the objective is to group similar data points into clusters based on their features(Amin et al., 2023). This process is pivotal for identifying inherent structures within data and is widely used in various applications, including image processing(Amin et al., 2023), segmentation(Ullah, Hussain, Ullah, Lee, & Baik, 2023), and anomaly detection (De Donato et al., 2023). In anomaly detection, particularly within CCTV footage from home security systems, clustering plays a crucial role. Anomalies(Guo, Lu, Jia, Zhang, & Li, 2024), or rare patterns, are data points that deviate significantly from the norm and can indicate potential threats or unusual activities(Yasin, Tahir, & Frnda, 2023). Identifying these anomalies is essential for enhancing security and ensuring timely responses to potential threats.

Traditional K-means clustering, a popular and straightforward method, partitions data into a predefined number of clusters based on proximity to cluster centroids(Jain & Pamula, 2018). While effective for many applications, standard K-means may struggle to detect rare patterns due to their reliance on mean values and sensitivity to the number of clusters specified(Liu, 2024). As a result, advanced variations of K-means have been developed to address these limitations and improve performance in complex scenarios(Zulfauzi, Dahlan, Sintuya, & Setthapun, 2023). Fuzzy C-means (FCM) introduces the concept of partial membership, where data points can belong to multiple clusters with varying degrees of membership(Izakian & Pedrycz, 2013). This flexibility is beneficial for capturing subtle anomalies that may not fit neatly into a single cluster. K-medoids (PAM), on the other hand, select actual data points as cluster centers, which enhances robustness against noise and outliers, making it suitable for detecting rare patterns that might be overshadowed by more frequent data(Sureja, Chawda, & Vasant, 2022). Mini-batch K-means offers efficiency for large-scale datasets by processing small, random subsets of data at a time, which can accelerate clustering without compromising significant pattern detection(Xiao, Wang, Liu, & Liu, 2018). Bisecting K-means provides hierarchical clustering by recursively splitting clusters, allowing for a more granular analysis that can be particularly useful in identifying rare patterns across different levels of detail(Gao, 2022). Kernel K-means extends the traditional approach by applying kernel methods to capture non-linear relationships within the data, offering enhanced flexibility in modeling complex patterns(Su, Guo, Wu, Jin, & Zeng, 2024). Weighted K-means assign different weights to data points, which can improve sensitivity to rare but critical patterns(X. Li, Guan, Deng, & Li, 2022).

Given the increasing complexity and volume of CCTV footage data, the need for advanced K-means variations becomes evident. These methods offer enhanced capabilities for rare pattern mining, enabling more accurate and effective detection of anomalies in home security contexts. This study explores these advanced clustering techniques, assessing their efficacy in identifying and analyzing rare patterns within a

Received : 17 November 2024
Revised : 29 November 2024
Accepted : 13 December 2024

comprehensive dataset of real-world CCTV footage(Kaur, Rani, & Kaur, 2024). The goal is to advance the state of anomaly detection in home security systems, ultimately contributing to more effective surveillance and threat management..

2 Literature Review

Weighted K-means Weighted K-means assign different weights to data points, allowing them to handle varying importance or frequencies of points. The process begins by assigning weights to each data point. Then, in each iteration of the algorithm, the weighted data points are used to update the centroids(Premkumar et al., 2024). The algorithm continues to adjust the centroids until convergence is reached, meaning the centroids no longer change significantly from one iteration to the next.

Fuzzy C-means (FCM) Unlike K-means, which assigns each point to exactly one cluster, Fuzzy C-means allows each data point to belong to multiple clusters with varying degrees of membership(J. Li, Izakian, Pedrycz, & Jamal, 2021). This is particularly useful when clusters overlap. The algorithm starts by initializing a membership matrix, where each entry indicates the degree to which a data point belongs to a specific cluster, with values randomly assigned between 0 and 1. During each iteration, the centroids of the clusters are recalculated using a weighted average, where the weights come from the membership values. After updating the centroids, the membership matrix is revised by adjusting the degree of membership for each data point based on its distance to the centroids data points closer to a centroid have a stronger membership in that cluster(Luo et al., 2024). This process continues iteratively until the algorithm converges, meaning the centroids and membership values no longer change significantly.

K-medoids (PAM) K-medoids, or Partitioning Around Medoids, is similar to K-means but uses actual data points as cluster centers (medoids), making it more robust to outliers(Best, Foo, & Tian, 2022). The algorithm begins by randomly selecting k medoids from the data set(Best et al., 2022). Then, each data point is assigned to the nearest medoid, forming clusters(Best et al., 2022). After the initial assignment, for each medoid, the algorithm checks if swapping the medoid with any other data point in the cluster improves the clustering quality(Best et al., 2022). Specifically, it tests if the swap reduces the total distance between the data points and their assigned medoid(Best et al., 2022). This process of swapping and reassignment is repeated until no further improvements can be made, meaning the total distance of points to their medoids can no longer be reduced. The result is a stable clustering where each data point is assigned to its nearest medoid(Best et al., 2022).

Mini-batch K-means Mini-batch K-means(Wang, Zhou, & Li, 2020) reduces the computational load by using small, random subsets (mini-batches) of the data to update cluster centers, making it suitable for large datasets. It begins by randomly initializing k cluster centers. Instead of using the entire dataset in each iteration, the algorithm selects a random subset of the data, known as a "mini-batch" (Wang et al., 2020). Each point in this mini-batch is assigned to the nearest cluster center based on a predefined distance metric (usually Euclidean distance) (Wang et al., 2020). After assignment, the cluster centers are updated based on the points in the mini-batch, rather than the entire dataset. This process is repeated iteratively until the algorithm converges, meaning that the cluster centers stop changing significantly(Wang et al., 2020).

Bisecting K-means Bisecting K-means combines hierarchical and partitional clustering approaches (Chen et al., 2021). It iteratively splits clusters to form a binary tree of clusters. It starts by treating all data points as a single cluster (Chen et al., 2021). In each iteration, the algorithm selects a cluster to split and applies the K-means algorithm with k = 2 to divide it into two sub-clusters (Chen et al., 2021). This splitting process continues until the desired number of clusters is reached (Chen et al., 2021). By focusing on splitting one cluster at a time, this algorithm provides a more balanced and efficient way of clustering large datasets compared to the standard K-means, especially for cases where the number of clusters is predefined(Chen et al., 2021).

Kernel K-means Kernel K-means extends the algorithm (Ikotun, Ezugwu, Abualigah, Abuhaija, & Heming, 2023) to operate in a higher-dimensional space using kernel functions (Ikotun et al., 2023), allowing it to capture non-linear relationships. The data is first mapped into a higher-dimensional space using a kernel function (Ikotun et al., 2023), which allows linear separability in the transformed space. Once the data is mapped, the standard K-means algorithm is applied in this new space (Ikotun et al., 2023). The cluster assignments are computed, and centroids are updated in this kernel space, enabling the algorithm to capture complex, non-linear relationships in the data(Ikotun et al., 2023).

K-means++ K-means++ improves the initialization step of K-means by carefully choosing initial cluster centers to accelerate convergence and improve the clustering result. It selects initial centroids distant from each other, reducing the likelihood of poor clustering (Zulfauzi et al., 2023). The first centroid is chosen randomly from the data points (Zulfauzi et al., 2023). For each subsequent centroid, the algorithm calculates the distance between each point and the nearest existing centroid, then selects the next centroid from the data points with a probability proportional to the square of these distances (Zulfauzi et al., 2023).

Extended versions of K-means clustering provide enhanced flexibility, robustness, and accuracy, making them suitable for a wider range of applications(Lu et al., 2024). Each variant addresses specific limitations of the traditional K-means algorithm, offering solutions for improved initialization, handling overlapping clusters, robustness to outliers, scalability, hierarchical structure, complex data structures, non-linear relationships, and weighted importance of data points(Thiyagarajan & Murugan, 2023).

3 Design

Preprocessing phase: A one-hour .dav video file is processed to extract frames at one-second intervals, creating a dataset of unique frames (Fan, Xia, Liu, & Li, 2021).
Feature extraction phase: The Mask R-CNN model, powered by the ResNet50 backbone, is employed to segment and identify human subjects in each extracted frame (Fan et al., 2021).
This method integrates the recognition of family members and enables the detection of potential intruders within the recorded surveillance footage.

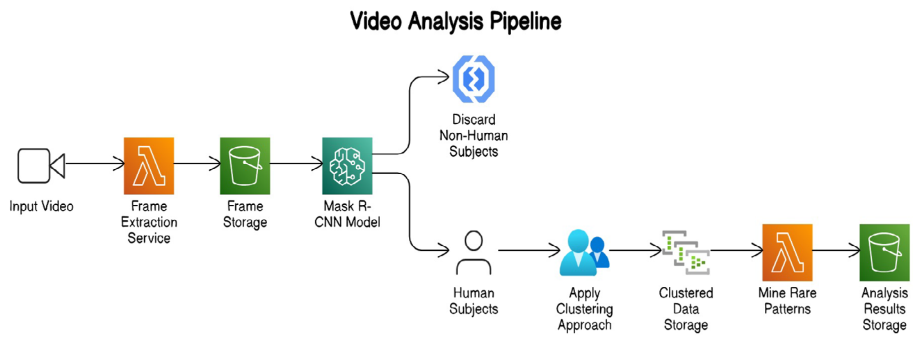

Figure 1: Schematic Diagram

Figure 1 shows the video analysis pipeline takes an input Video, which is the starting point of the pipeline. The input video is the raw data that will be processed. Frame Extraction Service extracts individual frames from the input video. These frames serve as the basic units for subsequent analysis. Frame Storage is where the extracted frames are stored for future reference and processing. Mask R-CNN Model: This is a deep learning model (Fan et al., 2021) that is used to detect and segment objects within each frame. It can identify human subjects and other relevant objects. The model's output is filtered to discard frames or regions that do not contain human subjects. The frames containing human subjects are selected for further analysis. Different possible clustering algorithms are applied to group similar human subjects or behaviors together. This can help identify patterns or anomalies. there is a clustered data is stored for future analysis or visualization. Rare Pattern Mining, techniques are applied to identify rare or unusual patterns within the clustered data. These patterns might indicate anomalies or interesting behaviors. The final analysis results, including identified patterns and insights, are stored for future reference or reporting.

4 Discussion

The dataset utilized in this study comprises 10,800 frames, distributed in three different datasets, which were collected from three distinct cameras. This comprehensive collection of frames ensures a diverse range of scenes and activities, providing a robust basis for analyzing and detecting anomalies. By integrating footage from multiple cameras, the dataset captures a variety of perspectives and contexts, enhancing the depth and accuracy of the rare pattern mining process.

The study utilizes TPU v2-8 running plateform is a high-performance processing unit specifically designed for machine learning workloads. It features a configuration with 8 TPU cores, each engineered to deliver high-throughput, low-latency computation. In terms of processing power, each individual TPU v2 core provides 45 teraflops (TFLOPS) for 16-bit floating-point operations, which totals 180 TFLOPS when utilizing all eight cores together. This capacity allows the TPU v2-8 to handle the intensive computations typical in machine learning models efficiently.

Weighted K-means Figures 2 illustrate the implementation of Weighted K-means on Data sets 1, 2, and 3, respectively. Each image is reshaped into a 1D array, and KMeans clustering is applied with specified

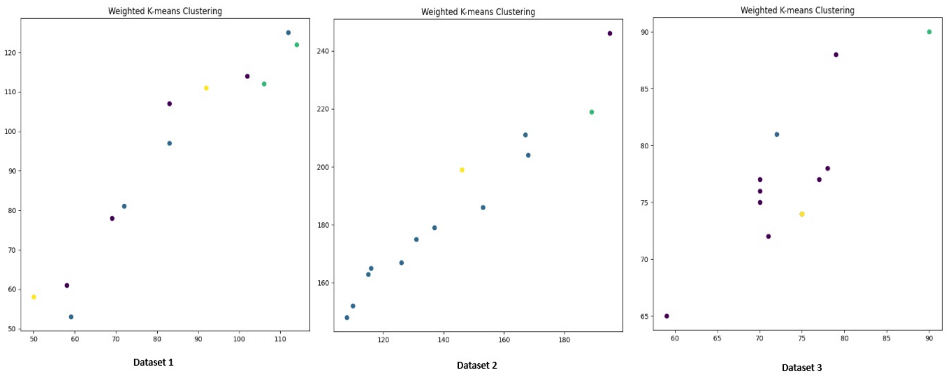

Figure 2: Weighted K-means on Data set 1, 2, and 3

parameters: `num_clusters=4` (number of clusters), `max_iter=50` (maximum iterations per clustering run), and `n_init=10` (number of initializations to find the best cluster configuration). Finally, images are saved into directories corresponding to their assigned clusters, allowing for organization based on visual similarities within the data. The figures include scatter plots that visualize how rare and frequent frames are distributed.

Fuzzy C-means Figures 3 showcase the ongoing application of the Fuzzy C-means to Data sets 1, 2, and 3, correspondingly. Each image is reshaped to a 1D array, and clustering is performed with parameters such as `num_clusters=4` (number of clusters), `error=0.005` (convergence threshold), and `maxiter=1000` (maximum iterations). The images are then saved into directories according to their assigned clusters, facilitating organization based on visual similarity within the data. The figures showcase scatter plots highlighting the spatial distribution of frames classified as rare or frequent.

K-medoids Figures 4 illustrate the consistent use of K-medoids on Data sets 1, 2, and 3. Each image is reshaped into a 1D array, and clustering is performed with parameters such as `num_clusters=4` (number of clusters) and `max_iter=300` (maximum iterations). Finally, the images are saved into directories corresponding to their assigned clusters, facilitating organization based on visual similarity within the data. The visualizations emphasize the distinction between frames that appear frequently and those that are rare.

Mini-batch K-means Figures 5 exemplify the continuous utilization of Mini-batch K-means on Data sets 1, 2, and 3, respectively. Each image is reshaped into a 1D array, and clustering is applied with parameters such as `num_clusters=4` (number of clusters) and `batch_size=100` (number of samples per batch). Finally, images are saved into directories corresponding to their assigned clusters, facilitating organization by visual similarity within the data. These figures provide a visual representation of how rare and frequent frames are arranged.

Bisecting K-means Figures 6 exemplify the implementation of Bisecting K-means on Data sets 1, 2, and 3, respectively. Each image is flattened into 2D arrays and iteratively splitting clusters based on variance. Each image is assigned to one of the resulting clusters. The parameter `num_clusters` specifies the desired number of final clusters, set to 4 in the code. The `KMeans` algorithm is used with `n_clusters=2` to split clusters, while `init='k-means++'` ensures efficient centroid initialization. Additionally, `max_iter=100` limits iterations, `n_init=10` runs the algorithm multiple times for stability, and `random_state=0` ensures reproducibility. The images are then saved in separate directories based on their cluster labels. The figures illustrate how rare and frequent frames are distributed.

Kernel K-means Figures 7 shows the implementation of the Kernel K-means to Data sets 1, 2, and 3, correspondingly. Each image is flattened into 2D arrays. The parameter `gamma` controls the spread of the RBF kernel, with a lower value indicating a wider influence, set to 0.1 in the code. The `num_clusters` parameter specifies the number of clusters for the KMeans algorithm, which is set to 4 for the image dataset. The `KMeans` algorithm is configured with `max_iter=50` for the maximum number of iterations and `n_init=5` to run the algorithm 5 times with different initializations to ensure stability. The figures visually represent the organization of rare and frequent frames.

K-means++ Figures 8 show the utilization of K-means++ on Data sets 1, 2, and 3, respectively. Each image is flattened into 2D arrays. The `num_clusters` parameter defines the number of clusters for the KMeans algorithm, set to 4 in this case, which determines the number of image groupings. The KMeans algorithm is initialized using `init='k-means++'` to improve the selection of initial centroids, with `max_iter=300` allowing up to 300 iterations and `n_init=10` running the algorithm 10 times to ensure optimal results. The `random_state=0` ensures reproducibility of the clustering results by controlling the random initialization process. The figures provide a visual representation of how rare and frequent frames are structured.

Figure 3: Fuzzy C-means on Data set 1, 2, and 3

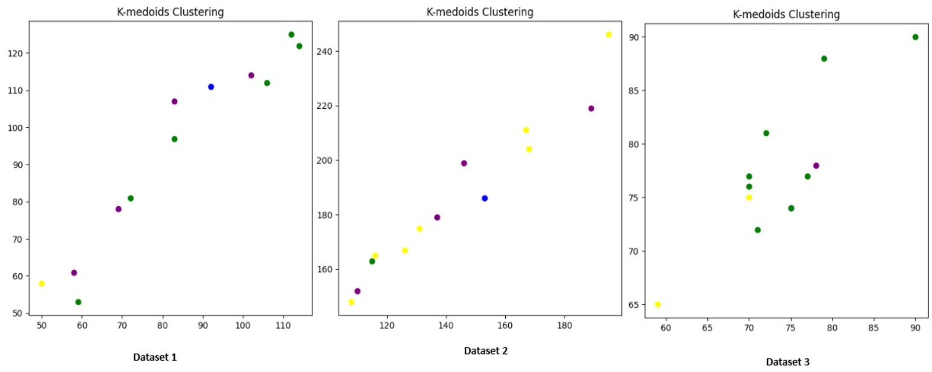

Figure 4: K-medoids on Data set 1, 2, and 3

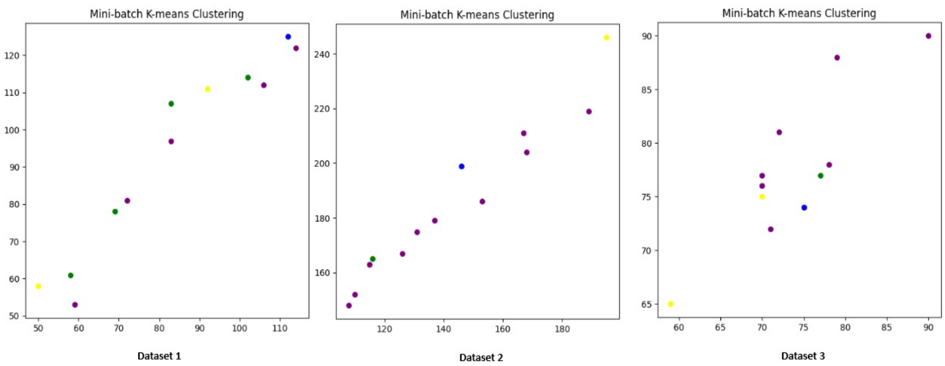

Figure 5: Mini-batch K-means on Data set 1, 2, and 3

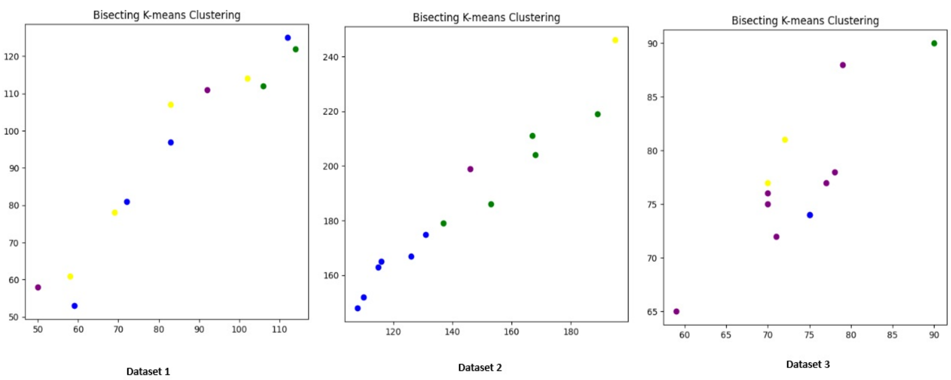

Figure 6: Bisecting K-means on Data set 1, 2, and 3

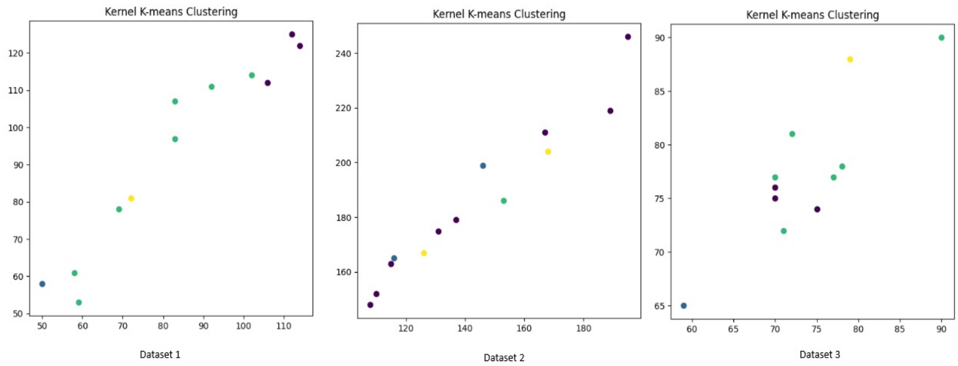

Figure 7: Kernel K-means on Data set 1, 2, and 3

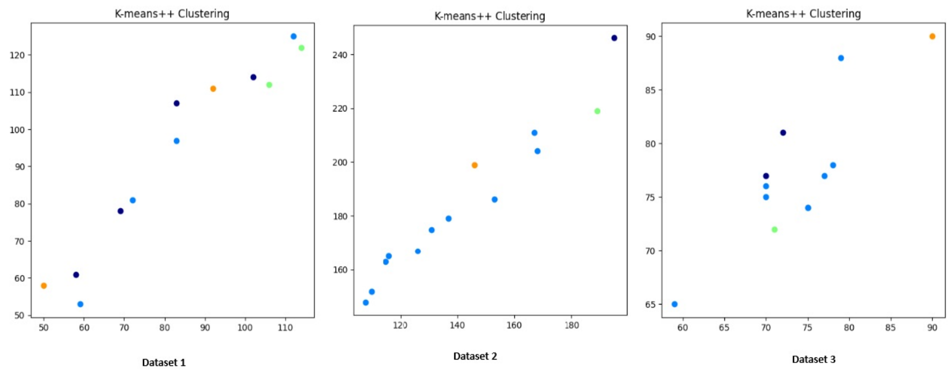

Figure 8: K-means++ on Data set 1, 2, and 3

5 Results

Table1 provides a comparative analysis of different clustering techniques for identifying rare patterns in data. Each row represents a unique clustering algorithm, while the columns show various performance metrics to assess the efficacy of each technique. The clustering techniques analyzed include Kernel K-means (Ikotun et al., 2023), Mini-batch K-means (Wang et al., 2020), K-means++ (Zulfauzi et al., 2023), K-medoids (PAM) (Best et al., 2022), Weighted K-means (Premkumar et al., 2024), Bisecting K-means (Chen et al., 2021), and Fuzzy C-means (J. Li et al., 2021). Each of these techniques has its unique approach to clustering data, which can impact its performance in detecting rare patterns. The Frames Extracted column indicates the total number of frames used in the study, with a consistent value of 10,800 across all techniques. This provides a standardized dataset size, ensuring that each technique has an equal opportunity to identify patterns. The Human Subject Extracted column lists the number of patterns (1,273) identified by Mask-RCNN-Resnet50-fpn-3x model, serving as a reference to evaluate the clustering algorithms' accuracy. The Actual Rare Patterns column displays the ground truth, with 37 rare patterns confirmed by human experts. This value remains constant across techniques and serves as a baseline to assess the predicted rare patterns.

The Predicted Rare Patterns column shows the number of rare patterns each algorithm identified, with all techniques predicting 37 patterns. This alignment with the actual rare pattern count suggests that each algorithm was calibrated to match the expected number of patterns. However, the true effectiveness of each algorithm lies in the accuracy and error rates in identifying these patterns. The True Predictions column captures the number of correct predictions, indicating each algorithm's success in identifying genuine rare patterns. For instance, Fuzzy C-means achieved the highest number of correct predictions, suggesting it performed well in accurately identifying rare patterns. In contrast, techniques like Kernel K-means and Mini-batch K-means have lower true prediction counts, indicating a less precise identification capability. Finally, the False Predictions column displays the count of incorrect rare pattern identifications. A lower false prediction count implies greater accuracy, as fewer incorrect patterns were mistakenly labeled as rare. Fuzzy C-means again stands out with only 4 false predictions, suggesting it is the most reliable algorithm among the techniques tested. Bisecting K-means also performs well with 9 false predictions, while other methods, like Kernel K-means and K-medoids, have higher false prediction counts, indicating a tendency to misidentify non-rare patterns as rare. The resultant sample figures of rare patterns are detailed in Figure 9.

Figure 9: Rare patterns extracted from surveillance videos

Table 1: Comparison of Clustering Techniques for Rare Pattern Detection

Clustering techniques	Frames extracted	Human subject extracted	Actual rare patterns	Predicted rare patterns	True Predictions	False Predictions
Kernel K-means	10,800	1273	37	37	17	20
Mini-batch K-means	10,800	1273	37	37	19	18
K-means++	10,800	1273	37	37	21	16
K-medoids (PAM)	10,800	1273	37	37	21	16
Weighted K-means	10,800	1273	37	37	24	13
Bisecting K-means	10,800	1273	37	37	28	9
Fuzzy C-means	10,800	1273	37	37	33	4

Figure 10 provides a comparative evaluation of various clustering techniques. The evaluation is based on two key metrics: the percentage of correct predictions and the percentage of missing predictions. The analysis reveals a trend where the accuracy of correct predictions generally increases as we move from Kernel K-means (Ikotun et al., 2023) to Fuzzy C-means (J. Li et al., 2021). Specifically, Kernel K-means (Ikotun et al., 2023) and Mini-batch K-means (Wang et al., 2020) demonstrate lower rates of correct predictions (46 percent and 51 percent, respectively), while more advanced techniques like K-means++ (Zulfauzi et al., 2023) and K-medoids (PAM) (Best et al., 2022) show moderate improvement with a 57 percent accuracy rate. The accuracy improves with Weighted K-means (Premkumar et al., 2024) (65 percent) and Bisecting K-means (Chen et al., 2021) (76 percent), ultimately peaking with Fuzzy C-means (J. Li et al., 2021), which achieves an 89 percent correct prediction rate. On the other hand, the occurrence of missing predictions decreases progressively across the methods, starting from 54 percent with Kernel K-means (Ikotun et al., 2023) and reducing to 11 percent with Fuzzy C-means (J. Li et al., 2021), indicating that Fuzzy C-means (J. Li et al., 2021) is the most effective in minimizing incorrect clustering.

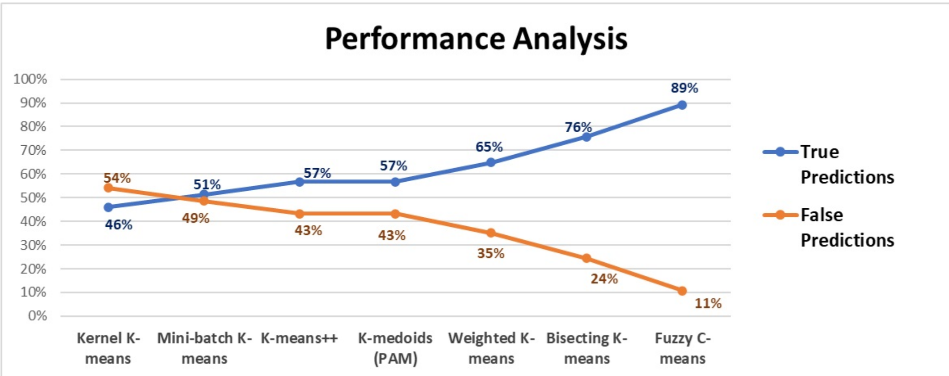

Figure 10: Average Performance

6 Conclusion

The analysis shows that Fuzzy C-means (J. Li et al., 2021) is the most robust clustering method among those evaluated, offering the highest accuracy in correct predictions while also significantly minimizing missing predictions. This suggests that Fuzzy C-means (J. Li et al., 2021) is well-suited for complex clustering tasks, particularly in scenarios where precision is critical. In contrast, Kernel K-means (Ikotun et al., 2023) and Mini-batch K-means (Wang et al., 2020) are less effective, showing lower accuracy and a higher rate of missing predictions. Therefore, in practical applications, particularly those involving intricate data like video analysis, Fuzzy C-means (J. Li et al., 2021) would be the preferred choice due to its superior performance. Looking forward, several enhancements could further improve clustering outcomes. One approach is to explore hybrid models that combine the strengths of different clustering algorithms, potentially enhancing accuracy and reducing errors even further.

References

Amin, J., Anjum, M. A., Ibrar, K., Sharif, M., Kadry, S., & Crespo, R. G. (2023). Detection of anomaly in surveillance videos using quantum convolutional neural networks. *Image and Vision Computing*, *135*, 104710.

Best, L., Foo, E., & Tian, H. (2022, September). Utilising k-means clustering and naive bayes for iot anomaly detection: A hybrid approach. In *Secure and trusted cyber physical systems* (pp. 177–214).

Chen, Y., Liu, W., Zhao, H., Cao, S., Fu, S., & Jiang, D. (2021, February). Bisecting k-means based fingerprint indoor localization. *Wireless Networks*, *27*, 3497–3506.

De Donato, L., Marrone, S., Flammini, F., Sansone, C., Vittorini, V., Nardone, R., … Bernaudin, F. (2023). Intelligent detection of warning bells at level crossings through deep transfer learning for smarter railway maintenance. *Engineering Applications of Artificial Intelligence*, 106405.

Fan, Z., Xia, W., Liu, X., & Li, H. (2021). Detection and segmentation of underwater objects from forward-looking sonar based on a modified mask rcnn. *Signal, Image and Video Processing*, *15*, 1135-1143.

Gao, H. (2022). A peer-to-peer distributed bisecting k-means. In *Proceedings of the 2022 14th international conference on machine learning and computing (icmlc '22)* (pp. 170–177). ACM.

Guo, J., Lu, S., Jia, L., Zhang, W., & Li, H. (2024). Encoder-decoder contrast for unsupervised anomaly detection in medical images. *IEEE Transactions on Medical Imaging*, *43*, 1767-1778.

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023, April). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, *622*, 178–210.

Izakian, H., & Pedrycz, W. (2013). Anomaly detection in time series data using a fuzzy c-means clustering. In *2013 joint ifsa world congress and nafips annual meeting (ifsa/nafips).* IEEE.

Jain, P. K., & Pamula, R. (2018). Two-step anomaly detection approach using clustering algorithm. In *Proceedings of the international conference on advanced computing networking and informatics* (pp. 513–520). Springer. doi: 10.1007/978-3-319-99981-$4_5$3

Kaur, N., Rani, S., & Kaur, S. (2024). Real-time video surveillance based human fall detection system using hybrid haar cascade classifier. *Multimedia Tools and Applications*, *1*, 3.

Li, J., Izakian, H., Pedrycz, W., & Jamal, I. (2021, March). Clustering-based anomaly detection in multivariate time series data. *Applied Soft Computing*, *100*, 106919.

Li, X., Guan, S., Deng, S., & Li, M. (2022). Improved weighting k-means algorithm based on covariance matrix. In *Proceedings of the 2022 6th asian conference on machine learning (acml).* IEEE.

Liu, N. (2024). Cctv cameras at home: Temporality experience of surveillance technology in family life. *Journal of New Media & Society*, 1-19.

Lu, H., Xu, H., Wang, Q., Gao, Q., Yang, M., & Gao, X. (2024). Efficient multi-view k-means for image clustering. *IEEE Transactions on Image Processing*, *33*, 851-862.

Luo, K., Liu, C., Chen, B., Li, X., Peng, D., & Yuan, Z. (2024, August). MFCD: A Deep Learning Method with Fuzzy Clustering for Time Series Anomaly Detection. In *Web and big data (apweb-waim 2024)* (pp. 62–77).

Premkumar, M., Sinha, G., Ramasamy, M. D., Sahu, S., Subramanyam, C. B., Sowmya, R., … Derebew, B. (2024, March). Augmented weighted k-means grey wolf optimizer: An enhanced metaheuristic algorithm for data clustering problems. *Scientific Reports*, *14*.

Su, R., Guo, Y., Wu, C., Jin, Q., & Zeng, T. (2024, June). Kernel correlation–dissimilarity for multiple kernel k-means clustering. *Pattern Recognition*, *150*, 110307.

Sureja, N., Chawda, B., & Vasant, A. (2022, June). An improved k-medoids clustering approach based on the crow search algorithm. *Journal of Computational Mathematics and Data Science*, *3*, 100034.

Thiyagarajan, S. K., & Murugan, K. (2023). Arithmetic optimization-based k means algorithm for segmentation of ischemic stroke lesion. *Soft Computing*, 1-14.

Ullah, W., Hussain, T., Ullah, F. U. M., Lee, M. Y., & Baik, S. W. (2023). Transcnn: Hybrid cnn and transformer mechanism for surveillance anomaly detection. *Engineering Applications of Artificial Intelligence*, 106173.

Wang, Z., Zhou, Y., & Li, G. (2020, May). Anomaly detection by using streaming k-means and batch k-means. In *2020 5th ieee international conference on big data analytics (icbda).* IEEE. doi: 10.1109/ICBDA49040.2020.9101212

Xiao, B., Wang, Z., Liu, Q., & Liu, X. (2018). SMK-means: An Improved Mini Batch K-means Algorithm Based on Mapreduce with Big Data. *TSP*, *1*(1), 1–5.

Yasin, A., Tahir, S. B., & Frnda, J. (2023). Anomaly prediction over human crowded scenes via associate-based data mining and k-ary tree hashing. *International Journal of Intelligent Systems*, *2023*, Article ID 9822428, 12 pages.

Zulfauzi, I. A., Dahlan, N. Y., Sintuya, H., & Setthapun, W. (2023, November). Anomaly detection using k-means and long-short term memory for predictive maintenance of large-scale solar (lss) photovoltaic plant. *Energy Reports*, *9, Supplement 12*, 154–158.