1.Problem Statement:

Efficient PMU (Phasor Measurement Unit) data Compression in WAMS (Wide Area Monitoring Systems). Develop a generalized technique for data compression by eliminating spatial and temporal redundancy from the data and preserving critical information related to power systems events and disturbances.

2.Introduction:

Modern power systems rely heavily on Wide Area Monitoring Systems (WAMS) to capture and analyze a vast array of Phasor Measurement Unit (PMU) data. But because PMUs sample at high frequencies, the resulting massive amount of data presents serious storage, transmission, and processing challenges. The sheer amount of data that PMUs continuously produce overloads the capacity of the network, stresses storage infrastructure, and places a significant computational load on monitoring systems. Furthermore, by adding to the already onerous storage constraints, regulatory compliance, which mandates the preservation of historical PMU data, exacerbates these difficulties. The inability to handle massive amounts of data in real-time can lead to delays that threaten power grid stability in critical situations. Periods of noise or negligible fluctuations within these massive datasets can mask important system analysis, resulting in inefficient decision-making.

By drastically lowering the amount of data, compression is a powerful tool that can be used to maximize storage utilization and minimize bandwidth requirements for data transmission. This simplified method makes managing massive data flows easier and significantly reduces the processing load on monitoring systems. Importantly, data compression does more than just truncate information—rather, it does so sparingly, guaranteeing that important information about power system events and disturbances is preserved. In addition to making data more manageable and economical, this fine balance between data size and fidelity is essential to maintaining data integrity. These compression techniques enable utilities and grid operators to make informed decisions, preserve grid stability, and guarantee the overall reliability of power systems by improving the usefulness and operational efficiency of WAMS. The intelligent use of compression techniques enhances WAMS's capabilities and creates an atmosphere that supports quick and accurate decision-making, all of which help power grids remain stable and resilient in the face of changing circumstances.

3.Possible Approaches for Data Compression

1.Slack Referenced Encoding (SRE):

- Description: Utilizes Slack Referenced Encoding, prioritizing data fidelity.

- Compression Ratio (CR): Achieves a maximum compression ratio of 10.12.

-Efficiency: Despite achieving moderate compression, the technique may not be highly efficient due to the comparatively lower compression ratio.

2. Wavelet Packet Transform (WPT):

- Description: Applies Wavelet Packet Transform methodology.

- Compression Ratio (CR): Yields a lower compression ratio of 2.

- Fidelity: Demonstrates an impressive Root Mean Square Error (RMSE) of 3.68 × 10^-6, indicating high fidelity in data reconstruction despite the lower compression ratio.

3. Standard Compression Algorithms (e.g., szip algorithm):

- Description: Examines conventional compression algorithms like the szip algorithm.

- CR for Voltage and Frequency Data: Reports compression ratios around 2.77 for voltage data and 3.77 for frequency data.

- Evaluation: The achieved compression ratios are relatively low, considering the application to voltage and frequency data.

4. Other Techniques (e.g., Embedded Zero Tree Wavelet-Based Denoising and Compression):

- Description: Introduces various other techniques, such as embedded zero tree wavelet-based denoising and compression.

- Evaluation: While these techniques have been proposed, a comprehensive analysis of their efficiency and performance may be lacking, necessitating further scrutiny and comparative evaluation against established methods.

4. Methodology:

As shown below in fig 1. shows the detail flow chart of the compression technique which I am going to use to remove the temporal and spatial redundancy from the input data.

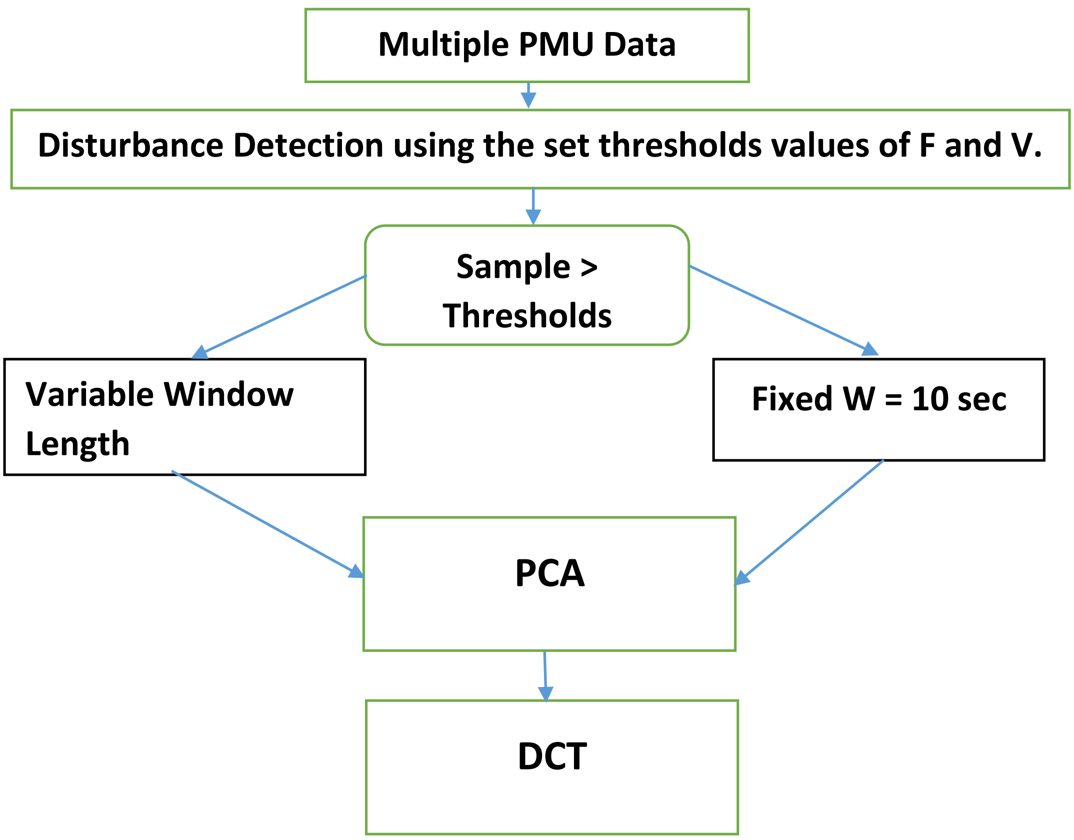


Fig1. Flow chart of compression technique

4.1. Statistical Change Detection (SCD):

1. Variable Window Length: Employs a variable window length based on specific disturbance triggering criteria for enhanced detection during PMU measurements.

- Traditional Criteria versus Less Conservative Criteria:

  - Recommends less conservative criteria than the traditional NERC standards, targeting even minor variations in PMU data.

  - Criteria include voltage ≤ 99% of nominal voltage and frequency ≤ 59.94 Hz or ≥ 60.06 Hz (±0.1% variation).

2. Disturbance Sample Evaluation:

- Triggering Mechanism: Evaluates each incoming sample Pi(n) as a disturbance sample if its deviation from the average value over a 10-second window exceeds the specified thresholds.

- Statistical Variance (SV) Calculation: Computes the statistical variance of disturbance samples over a moving window of 3 cycles, assuming disturbances will last for at least 3 cycles.

3. Adaptive Disturbance Detection:

- Dynamic Window Adjustment: Users can modify the window length for SV calculation based on the PMU reporting rate.

- Disturbance Termination: Considers a disturbance over when all samples in the moving window fall below the trigger threshold.

4.2. Principal Component Analysis:

Principal component analysis, or PCA, is a dimensionality reduction technique that is frequently used to reduce the dimensionality of large data sets, by reducing a large set of variables to a smaller one that still retains the majority of the information in the larger set.

Accuracy naturally suffers when a data set has fewer variables; however, the secret to dimensionality reduction is to compromise a little on accuracy in favor of simplicity. Smaller data sets facilitate exploration and visualization, and because they require fewer processing variables, machine learning algorithms can analyze data points more quickly and easily. PCA's basic principle is to minimize the number of variables in a data set while maintaining as much information as feasible.

4.2.1. Standardization:

1. In Principal Component Analysis (PCA), standardization refers to the process of scaling the variables to a common scale before performing the PCA. This step is crucial because variables in different units or scales might disproportionately influence the analysis.

2. Standardization involves transforming the variables so that they have a mean of zero and a standard deviation of one. It equalizes the importance of each variable in the PCA, ensuring that no single variable dominates the analysis simply because of its scale or unit.

3. Standardizing variables prior to PCA helps in achieving a more balanced and accurate analysis. It ensures that all variables contribute fairly to the creation of principal components, focusing the PCA on the underlying relationships between variables rather than their individual scales or units.

$$x = \frac{value - mean}{standard\ deviation} \tag{1}$$

4.2.2. Covariance Matrix Computation:

In Principal Component Analysis (PCA), the covariance matrix plays a crucial role in understanding the relationships between variables. It's a square matrix that captures the variance of variables and their pairwise relationships.

The covariance matrix is computed by taking the variables in the dataset and finding the covariance values between each pair of variables. The diagonal elements of the covariance matrix represent the variance of individual variables, while the off-diagonal elements represent the covariance between pairs of variables.

In our Case, the number of Variables = No. of Columns in data

If Number of variables = 3, this is how the matrix will be:

In our case the variable = 274.

$$D = \begin{matrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{matrix} \qquad (2)$$

The above covariance matrix can also be written as:

$$D = \begin{matrix} Var(x) & Cov(x,y) & Cov(x,z) \\ Cov(x,y) & Var(y) & Cov(y,z) \\ Cov(x,z) & Cov(y,z) & Var(z) \end{matrix} \qquad (3)$$

The steps involved in computing the covariance matrix for PCA are:

1. Centering the data: Subtracting the mean of each variable from the respective data points, resulting in zero-mean variables.

2. Calculating the covariance: Finding the average of the products of the differences between the values of two variables and their respective means. It can be calculated as:

$$Cov(X,Y) = \frac{\sum_{i=1}^{n}(Xi-\overline{X})(Yi-\overline{Y})}{n-1} \qquad (4)$$

3. Matrix representation: Collecting these covariance values into a matrix where the diagonal elements represent the variance of each variable, and the off-diagonal elements represent the covariance between variable pairs.

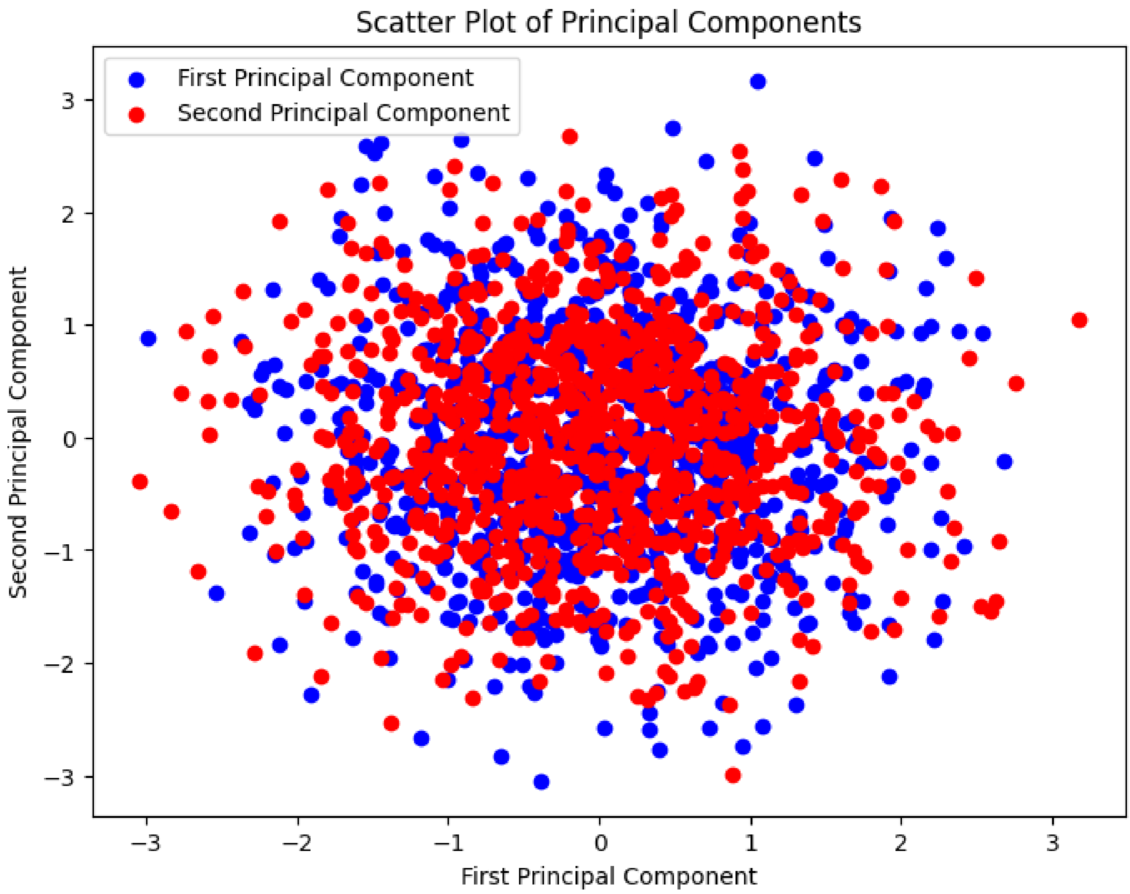
The covariance matrix provides essential information for PCA by indicating the strength and direction of relationships between variables. In PCA, the principal components are derived from the eigenvectors and eigenvalues of this covariance matrix. The eigenvectors represent the directions of maximum variance (principal components), and the corresponding eigenvalues signify the magnitude of variance in those directions.

4.2.3. Principal Components:

Eigenvectors and eigenvalues serve as fundamental mathematical elements used to derive principal components from the covariance matrix. Before going into these mathematical concepts, let's grasp the essence of principal components.

Principal components represent novel variables derived as linear blends of the original variables. These combinations are meticulously formulated to ensure the new variables, or principal components, lack correlations among themselves while encapsulating the bulk of the information present in the original variables. In essence, even though a 10-dimensional dataset generates 10 principal components, PCA endeavors to encapsulate the maximum information in the initial components, sequentially accommodating residual information in subsequent components. This approach is akin to the representation depicted in the scree plot showcased below.

In our case, we have reduced the dimensions of our data from 274 variables to 4 variables or let’s say 4 principal component, where each maximum data is variance is retained by first pc and rest of them will store the same in decreasing order.



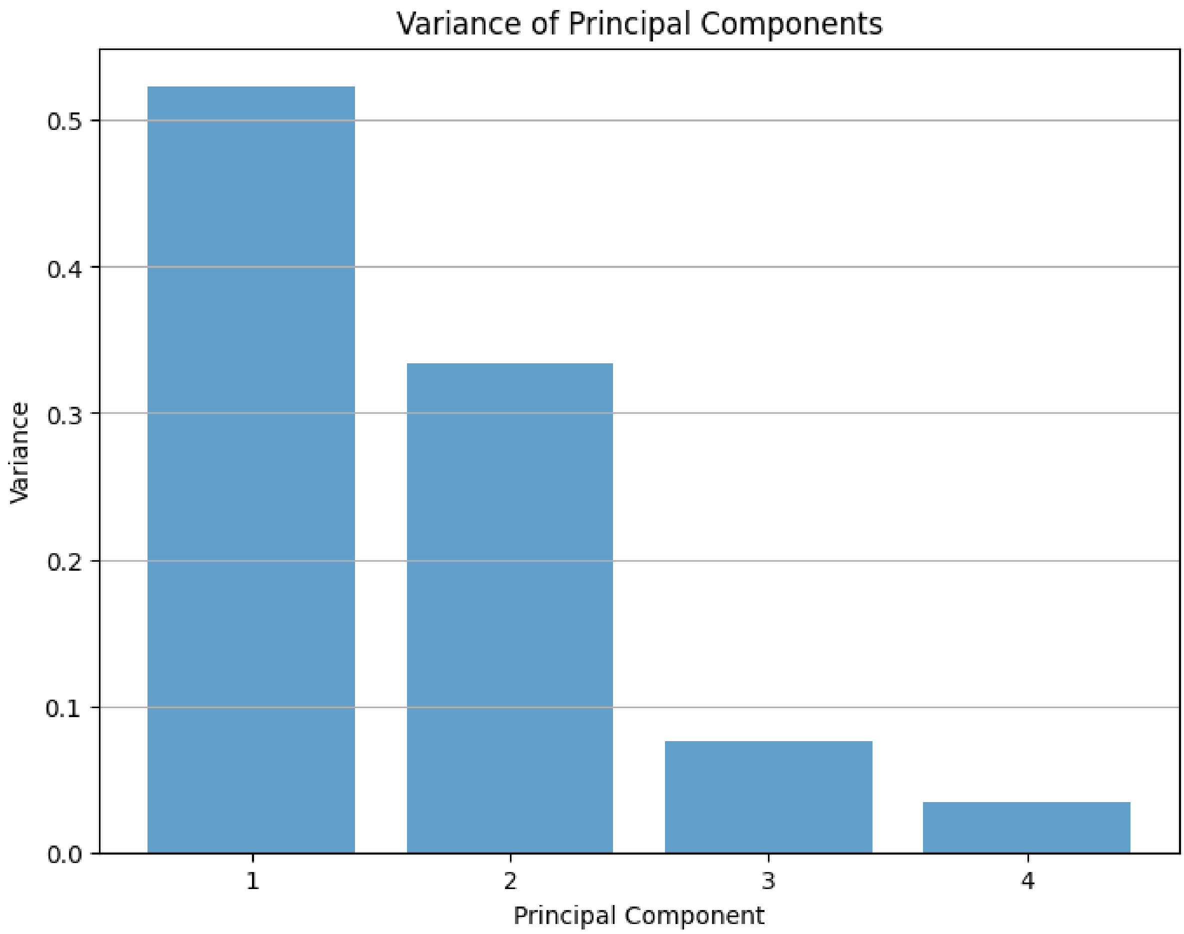
Plot1. Scattering of principal components

The above plots shows the scattering of first and second principal components. Here, I have created a plot of 1000 samples spread over 2 dimensions, each with values distributed around 0 (mean) with a standard deviation of 1.The range of values can vary, but the majority of the values fall within approximately -3 to +3 standard deviations from the mean for a standard normal distribution.

The variance stored by the principal components is as follows:

array ([0.52331883, 0.33437727, 0.0761982, 0.03467531])

in plot2, it can be seen that how the variance varies across the next principal components. This percentage represent the relative importance of each principal component in capturing the overall variance present in the data.



Plot1. Variance of principal components

4.3. Discrete Cosine Transform:

The Discrete Cosine Transform (DCT) is a mathematical technique used to compress Phasor Measurement Unit (PMU) data. DCT is applied for PMU data compression in the following way:

1. DCT as a Compression Technique:

- Transform-Based Method: DCT is a signal processing technique that converts a signal from its original domain (e.g., time domain) into the frequency domain.

- Frequency Representation: PMU data comprises various signals representing power system variables sampled over time. DCT transforms these signals into frequency components.

- Energy Compaction: DCT tends to concentrate the signal's energy into a small number of coefficients. These coefficients represent the most critical frequency components of the signal.

2. Application in PMU Data Compression:

- Signal Decomposition: PMU data, consisting of multiple signals from different measurements, is broken down into segments.

- DCT Calculation: The DCT algorithm is applied to each segment (signal) independently.

-Frequency Coefficients: DCT computes coefficients that represent the signal's frequency content. These coefficients highlight important frequency components in the signal.

- Threshold and Compression: Following the DCT computation, a threshold is set to filter out coefficients below a certain magnitude. Coefficients with magnitudes below this threshold are considered less significant and are discarded.

- Retaining Significant Components: Only the coefficients above the threshold, termed as 'significant' coefficients, are kept. These significant coefficients contain the most relevant frequency information for signal reconstruction.

3. Reconstruction and Decompression:

- Reconstruction from Coefficients: To reconstruct the original signal, the significant DCT coefficients are utilized.

- Inverse Transform: Applying the inverse DCT (IDCT) on the retained coefficients reconstructs the compressed signal.

5. Conclusion and Future Work:

Compression of the PMU data is basically consists of two stages, where in each stage we remove redundancies (repetitiveness of the data) from the data. In first stage we have used Principal Component Analysis which is a Machine Learning algorithm to remove the spatial Redundancy from the data. Basically we have represented a huge amount of data in few principal components, which contains or stores most of the variance of the data and retaining most of the valuable information from it.

In future work, I am going to work on removing the temporal redundancy from the data using transforms such as discrete cosine transforms or discrete wavelet transform. Will also find a way to handle a large set of data using the method of sliding window and variable window length, so that there is more efficient compression of the disturbance data can be possible thus maintaining the data fidelity.

6. References:

[1] L. Souto, J. Meléndez and S. Herraiz, "Comparison of Principal Component Analysis Techniques for PMU Data Event Detection," 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9281512.

[2] https://electricgrids.engr.tamu.edu/electric-grid-test-cases/synthetic-pmu-data/

[3]https://www.smartgrid.gov/document/pmus_and_synchrophasor_data_flows_north_america/

[4] P. H. Gadde, M. Biswal, S. Brahma and H. Cao, "Efficient Compression of PMU Data in WAMS," in IEEE Transactions on Smart Grid, vol. 7, no. 5, pp. 2406-2413, Sept. 2016, doi: 10.1109/TSG.2016.2536718. [5] D. Salomon, Data Compression: The Complete Reference. New York, NY, USA: Springer, 2004.

[5] L. Souto, J. Meléndez and S. Herraiz, "Comparison of Principal Component Analysis Techniques for PMU Data Event Detection," 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9281512.