

TDK-dolgozat

Ferenczi Martin

Ferenczi Martin

Six-sigma approach-based visibility graphs

Témavezető: Dr. Abonyi János, egyetemi tanár

HUN-REN-PE Complex Systems Monitoring Research Group

Six-sigma approach-based visibility graphs

Készítette: Ferenczi Martin
Adattudomány mesterképzés

Témavezető: Dr. Abonyi János

In this study, an innovative approach is introduced for time series analysis based on visibility graphs, augmented with the principles of Six Sigma. While visibility graphs offer a compact view of time series, preserving crucial characteristics, their integration with statistical process control tools of Six Sigma results in a more informative representation than previous methodologies. The proposed method enhances traditional approaches, providing a more organized and insightful analysis for accurate results. Incorporating the Six Sigma methodology brings a systematic strategy to the analysis process. With the data from our analysis, a decision tree was trained to identify rule violations, making the most of the information provided by the visibility graphs and Six Sigma methods.

800 time-series are analyzed, each 100 points long, categorizing them into labeled networks for detecting Western Electric and Nelson rule violations using Network and Violation Metrics. The dataset comprises 400 series each for Western Electric and Nelson rules. This approach allows for a deeper exploration of the time-series, revealing hidden relationships and valuable insights. The decision tree model was trained and subsequently achieved a detection accuracy of 91.976%.

Contents

Nomenclature	4
1 Introduction	5
2 Methodology of SPC-based Time Series Analysis	8
2.1 Definition of Visibility Graph	8
2.2 Classic HVG-based Approach of Time Series Analysis	11
2.3 Analytical Depth of Network Metrics in Visibility Graphs	12
2.4 Integrating the Core of Six Sigma Methodology	14
2.4.1 Western Electric Rules	15
2.5 Labeled Visibility Graphs in Time Series Analysis	17
2.6 Decision Tree Model for Process Control Analysis	18
2.7 SPC Principles in Visibility Graph-Based Time Series Analysis .	19
2.8 Transitioning from the Classical HVG-based Approach to the "Six Sigma-Based" Approach	20
2.9 The Six Sigma-Based Visibility Graph Method	21
3 Outcomes and Toolset Efficacy	26
3.1 Implementation and Analysis	26
3.1.1 Evaluation of Classifier Performance	33
3.2 Toolbox Capabilities and Implementation	36
4 Conclusions	38
Bibliography	41

Nomenclature

μ	Mean of a data set
σ	Standard deviation of a data set
$\{x_i\}_{i=1}^N$	Time series data set
$G(V, E)$	Visibility graph with nodes V and edges E
L	Distance of the control limits from the mean in terms of the number of standard deviations
N	Total number of observations in the series
t_i	Time at the i -th time instance
x_i	The value of the series at the i -th time instance
y_i	Data value at the i -th time instance

1. Chapter

Introduction

Visibility graphs are powerful tools in time series analysis, transforming sequential data into expansive network topologies. This transformation enables both granular and holistic analyses of data characteristics [1]. Employing various algorithms such as the Horizontal Visibility Graph and Multiplex Visibility Graph, time series and even images can be converted into complex networks, enriching the analysis with a multidimensional perspective [2]. This methodological diversity meets different analytical needs and facilitates the discovery of intricate data patterns.

The Horizontal Visibility Graph, which will be the focal point of this paper, revolutionizes the analysis of time series by leveraging graph-theoretical structures. This approach not only enhances traditional time series analysis but also bridges the gap with nonlinear dynamics and graph theory, allowing for a comprehensive application of network theory to time series data [3].

The use of visibility graphs has spread across disciplines, from financial markets to fMRI data analysis, evidencing their versatility [4]. By transforming time series into network models, these graphs illuminate patterns and dynamic interrelations that are critical for forecasting and offer a rich representation of the evolution of system over time.

Despite their advantages, visibility graphs are not without challenges. Specifically, the requirement to use lower and upper visibility graphs separately can be limiting, as it may obscure certain correlations and necessitate greater computational effort. Our research aims to overcome these hurdles by proposing a unified method that integrates the insights from both graph types, thereby enhancing data analysis efficiency and depth [5].

Central to our methodology is the process of quantization, which discretizes

continuous data, simplifying it for analysis through horizontal visibility graphs. By labeling quantized data, we construct a network that embodies the quantized values and their temporal relationships, presenting a comprehensive and nuanced view of the time series dynamics.

Our study aims to address the dichotomy of employing lower and upper visibility graphs by introducing an approach that combines quantization with visibility graphs, aiming for a more integrated analysis that can capture subtle patterns effectively.

The contributions of the study are as follows:

- We introduce a new quantization process that simplifies continuous time series data into distinct, finite values, paving the way for the construction of horizontal visibility graphs. This crucial step isolates key characteristics of the data, streamlining the subsequent analysis.
- Our construction of horizontal visibility graphs from labeled, quantized data opens a new pathway for analyzing the complex patterns found in time series. This method offers a thorough and detailed perspective of the behavior of data.
- We also develop a labeled network that, together with the horizontal visibility graphs, provides a comprehensive picture of the time series data. This network represents both the simplified data points and their chronological interconnections, delivering an in-depth understanding of the flow of data and structure.
- An extensive evaluation using a large dataset validates our approach. By applying our method to 800 different time-series, we demonstrate its capability to effectively identify specific rule violations, highlighting the robustness of our technique.
- The study presents a novel methodology that diverges from the traditional separate use of lower and upper visibility graphs. Our approach constructs visibility graphs based on zone information derived from statistical process control data, offering a fresh perspective for analysis that circumvents the need for separate visibility graphs.

Further sections will thoroughly detail our method, the comprehensive testing it underwent, and the significant findings that enhance the field of time series analysis. The insights gained here set a new direction for future research in this area.

The structure of the document is organized as follows: Chapter 2 introduces the SPC-based Visibility Graph for Time Series Analysis, starting with the basic definition of visibility graphs and progressing through classical and Six Sigma-enhanced approaches for time series analysis. In Chapter 3, we present the results, including toolbox capabilities and their implementation, data analysis and transformation specifics, and a detailed performance analysis of our methods. Chapter 4 provides concluding remarks, summarizing the key findings and outlining potential areas for future research. Finally, the bibliography lists the sources referenced throughout the study.

2. Chapter

Methodology of SPC-based Time Series Analysis

A time series can be formally represented as a sequence of data points

$$\{x_i\}_{i=1,\dots,N} \quad (2.1)$$

where:

- x_i denotes the value of the series at the i -th time instance.
- The index i runs from 1 to N , indicating the time order of the data points, with N being the total number of observations in the series.

2.1 Definition of Visibility Graph

The visibility graph $G(V, E)$ is constructed from a time series by mapping each data point (t_i, x_i) , where t_i represents specific moments in time, to a node in the graph. Edges between nodes, denoted as e_{ij} , are established based on a visibility criterion, which is defined by a direct line of sight between data points.

The visibility criterion is defined more formally as follows: two arbitrary data points (t_i, x_i) and (t_j, x_j) will have visibility, and consequently will be connected by an edge e_{ij} in the graph, if and only if for any other data point (t_k, x_k) located temporally between them, the following condition is satisfied:

$$x_k < x_i + (x_j - x_i) \frac{t_k - t_i}{t_j - t_i} \quad (2.2)$$

This criterion ensures that a straight line drawn from (t_i, x_i) to (t_j, x_j) does not intersect any data point (t_k, x_k) that lies between them on the time axis.

The steps to construct the visibility graph from a time series are as follows:

1. Assign each time-stamped data point (t_i, x_i) to a unique node in the graph.
2. Iteratively check each pair of nodes (t_i, x_i) and (t_j, x_j) against the visibility criterion to determine if an edge e_{ij} should be drawn between them.
3. After evaluating all pairs of nodes, the resulting graph encapsulates the structural and relational information of the time series, making it ready for further graph-theoretical analysis.

This approach translates the sequential data into a network structure, facilitating the analysis of time series through graph theory. It enables the identification of underlying patterns and dynamics, using network metrics and algorithms.

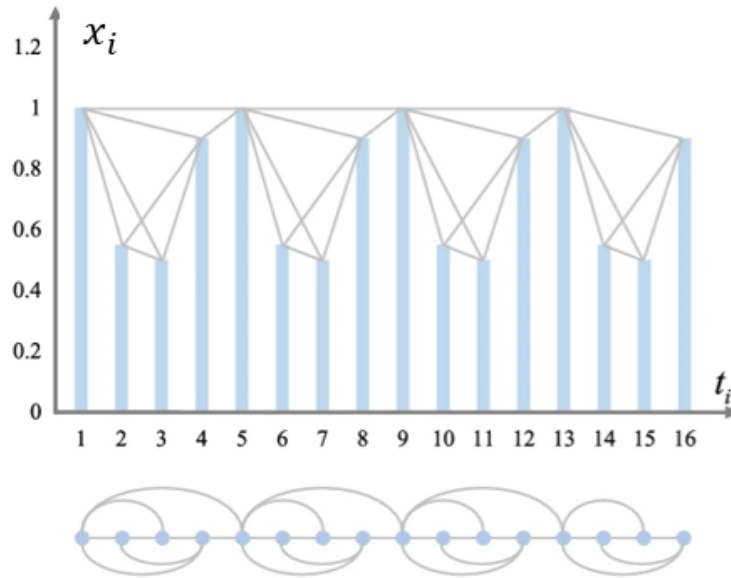


Figure 2.1: An example of a visibility graph constructed from a time series.

The Horizontal Visibility Graph (HVG) is a variant of the visibility graph, optimized for the analysis of time series data by emphasizing immediate fluctuations and relative magnitudes.

For a time series defined as a set of ordered pairs (t_i, x_i) , the HVG is built using the following rules:

1. Represent each data point (t_i, x_i) by a node v_i in the graph $G_h(V, E)$.
2. Connect two nodes v_i and v_j with an edge e_{ij} if and only if every data point (t_k, x_k) between them satisfies $x_k \leq \min(x_i, x_j)$ for all t_k such that $i < k < j$.

This edge criterion reflects a purely horizontal line of sight, giving rise to the nomenclature Horizontal Visibility Graph. It ensures that nodes v_i and v_j are connected if a horizontal line from x_i to x_j does not intersect any data points that have a higher value than both x_i and x_j .

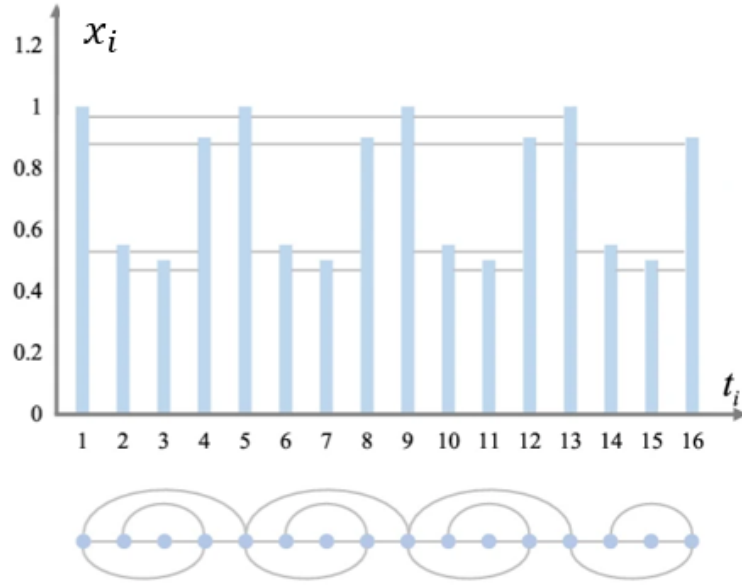


Figure 2.2: An example of a horizontal visibility graph constructed from a time series.

The Horizontal Visibility Graph (HVG) approach offers a streamlined and effective framework for analyzing time series data, accentuating pivotal features such as the relative magnitudes of data points. This methodology inherently diminishes the intricacy of the raw data, paving the way for graph-theoretical techniques to uncover and measure key aspects of the time series, including

cyclic patterns, directional tendencies, and stochastic properties. The HVG is particularly beneficial for:

- Identifying critical points in the time series: Local extrema in the data are intuitively represented by the nodal degrees within the HVG, providing immediate insight into significant fluctuations.
- Elucidating the data point distribution: The degree distribution of the HVG can offer clues to the statistical distribution of the time series values, which may be indicative of inherent stochastic processes or deterministic behaviors.
- Versatile applications across disciplines: In fields such as finance, meteorology, or even physiological signal processing, the prominence of peaks and troughs is crucial for interpreting the data. The HVG method excels in these areas by isolating these features for further analysis.

Moreover, the simplification process of HVG effectively filters out non-essential variability, often perceived as noise, thereby rendering a distilled visual representation. This simplification facilitates a more straightforward identification of structural patterns and dynamical trends within the time series. Nonetheless, it is essential to recognize that while the HVG method highlights certain aspects of the data, it may also obscure others, particularly nuances that occur at finer scales, which may be relevant in some analytical contexts. Therefore, it should be employed as a complementary tool within a broader analytical framework for time series investigation.

2.2 Classic HVG-based Approach of Time Series Analysis

In the classic approach, the data undergoes a series of transformations to adapt it for network theory techniques. This method consists of three main stages:

1. **Time Series Representation:** This initial stage uses the pre-defined time series data. This data forms the basis for the subsequent analysis.
2. **Visibility Graph Transformation:** The second stage involves the transformation of the time series data into a visibility graph. Each data

point in the series is mapped to a node within the graph. Nodes are interconnected based on the principle of direct visibility.

3. **Network Analysis:** The final stage entails analyzing the constructed visibility graph using various network analysis techniques. This includes computing key structural properties of the graph, such as degree distribution, centrality measures, clustering coefficients, and path lengths, to identify patterns and dynamics within the time series.

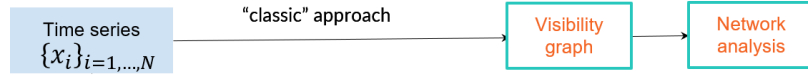


Figure 2.3: Schematic representation of the visibility graph transformation and subsequent network analysis in the classic approach to time series analysis.

2.3 Analytical Depth of Network Metrics in Visibility Graphs

The visibility graph derived from the time series, is subjected to an in-depth network analysis to uncover structural intricacies and dynamic patterns. This analysis encompasses several key metrics, each offering unique insights into the topology of the graph and the underlying time series data:

- **Degree and Degree Distribution:** The degree of a node in a network is defined as the number of connections (edges) it has with other nodes. This metric evaluates the connectivity of each node, quantifying its number of direct links within the graph. Understanding the degree distribution is fundamental in deciphering the structure of the network, pinpointing nodes with significant influence or connectivity.
- **Centrality Measures:** These metrics identify pivotal nodes within the network, including:
 - *Degree Centrality:* Highlights the importance of the node based on its linkage count.

- *Betweenness Centrality*: Depicts the role of the node in facilitating information flow through the network.
- *Closeness Centrality*: Measures the efficiency of the node in reaching other nodes, emphasizing its accessibility.
- **Clustering Coefficient**: This metric reveals propensity of nodes to form tightly-knit groups, illuminating the graph’s local clustering patterns.
- **Path Length**: Represents the average shortest distance between node pairs, offering insights into the network’s overall interconnectedness and ease of traversal.
- **Edge Weights**: These weights, representing temporal intervals between observations ($w_{i,i+l} = t_{i+l} - t_i$), capture the persistence of specific states in the time series. The figure below elucidates this concept.

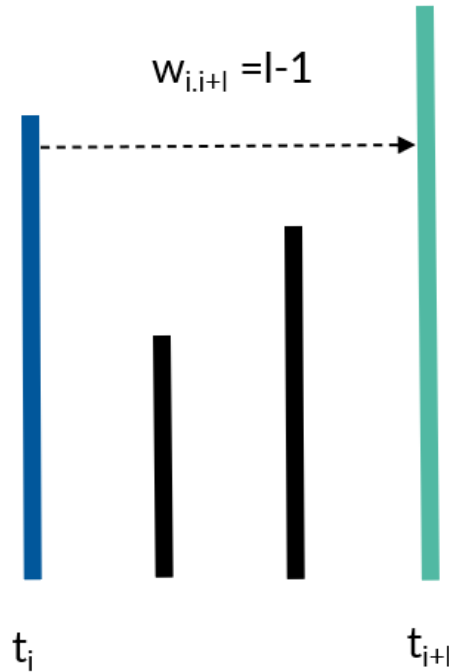


Figure 2.4: Interpretation of the edge weights in the network.

- **Node Degrees and Community Structures:** The in-degree and out-degree of nodes offer insights into the dynamics of trends and critical points within the time series. Additionally, identifying community structures through these degrees reflects the modularity of the network and distinct dynamic regimes or states of processes.

These metrics collectively contribute to a nuanced understanding of the time series data, enabling comprehensive analysis of system behavior, prediction of future states, and identification of anomalies. The subsequent figures offer visual interpretations of these concepts, further elucidating their significance in network analysis.

2.4 Integrating the Core of Six Sigma Methodology

At the heart of our Six Sigma-based approach is the Statistical Process Control (SPC) chart, an indispensable tool for monitoring, controlling, and improving process performance over time. The SPC chart graphically represents the variability of a process across a time series, enabling a detailed examination of observations taken at successive time intervals.

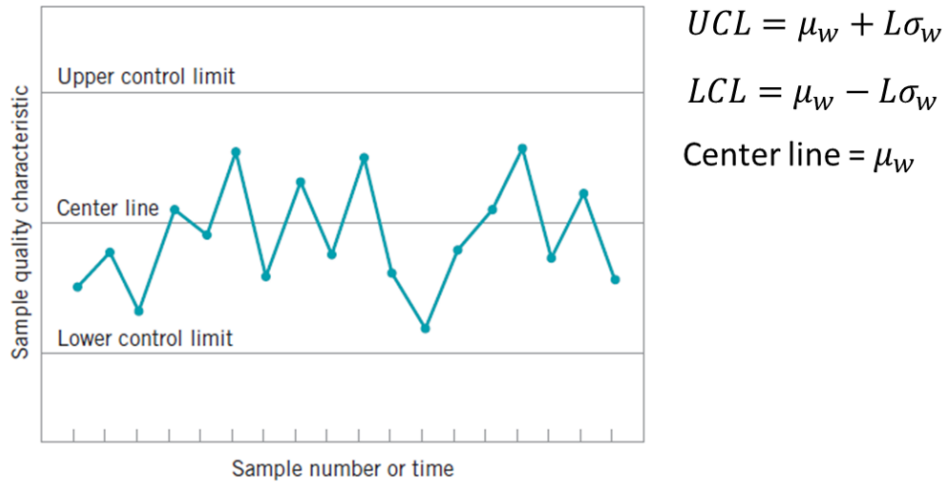


Figure 2.5: Simple Control Chart

The image above illustrates a simple control chart, which is a type of SPC chart. As you can see, it includes the main elements of a control chart:

- A Center Line (CL), representing the process mean (μ_w).
- Upper Control Limit (UCL) and Lower Control Limit (LCL), which define the boundaries of acceptable process variation.

The control limits are determined based on the statistical characteristics of the process:

$$UCL = \mu_w + L\sigma_w \quad (2.3)$$

$$LCL = \mu_w - L\sigma_w \quad (2.4)$$

where μ_w is the process mean and σ_w is the standard deviation, a measure of process variability. The multiplier L is usually set based on the desired confidence level for the control limits; in many practical applications, L is set to 3 to encapsulate 99.73% of the normal process variation, assuming the process follows a normal distribution. This setting aids in distinguishing between common-cause variation, which is inherent in the process, and special-cause variation, which may indicate a process change requiring investigation.

2.4.1 Western Electric Rules

The Western Electric rules provide a structured approach to analyzing time series data. These rules are instrumental in detecting significant patterns that may indicate a need for process intervention. In our methodology, we employ these rules as follows.

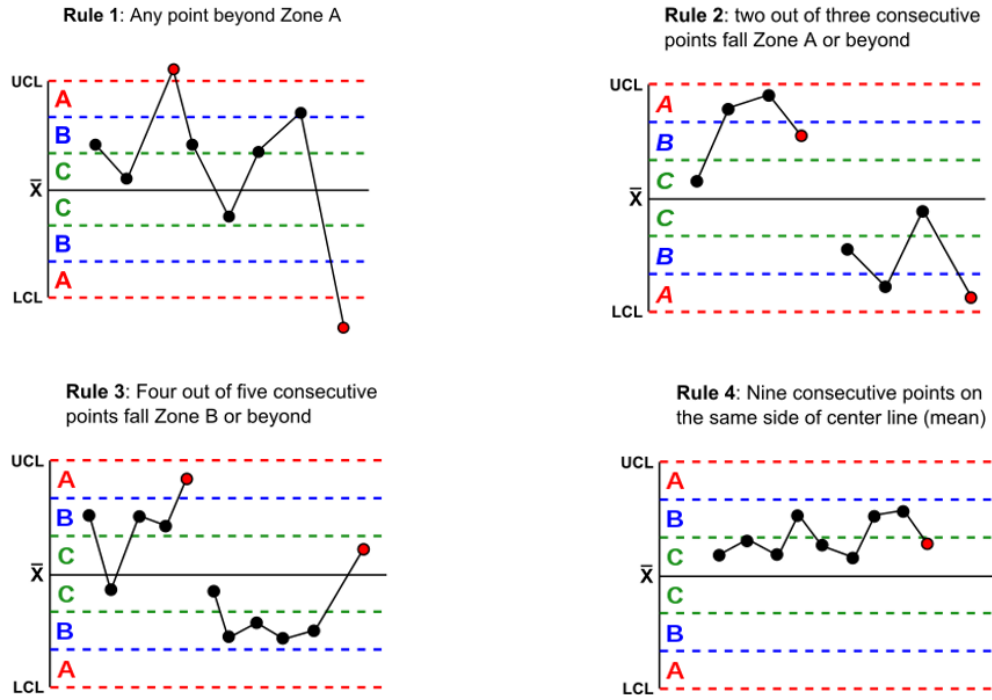


Figure 2.6: Illustration of Western Electric rules applied in visibility graphs

The Western Electric rules consist of four primary guidelines, each designed to flag potential issues in time series data:

- **Rule 1:** A single data point beyond Zone A.
- **Rule 2:** Two out of three consecutive points in Zone A or beyond.
- **Rule 3:** Four out of five consecutive points in Zone B or beyond.
- **Rule 4:** Nine consecutive points on the same side of the centerline.

2.5 Labeled Visibility Graphs in Time Series Analysis

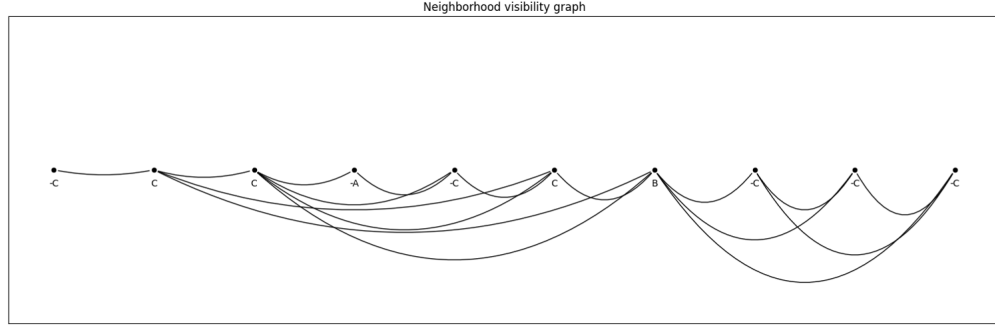


Figure 2.7: Illustration of a labeled visibility graph

Labeled visibility graphs represent a synthesis of graph theory and statistical process control (SPC) within the framework of time series analysis. In these graphs, denoted as $G_{vis} = (V, E, D)$, the set E is inherited from the construction of the horizontal visibility graph (HVG), where each edge represents visibility between data points in the time series. The set D , however, is a novel addition that imbues each node with a label indicating its 'zone' as per SPC chart classifications.

The 'zone' information encapsulated within D provides a powerful mechanism for categorizing the operational states of the system under analysis. Each zone label derives from predefined SPC zones, which segment the continuous fluctuation of data points into discrete, statistically significant regions. This zonal information is critical for:

- Characterizing the behavior of the time series data within the context of quality control and process stability.
- Identifying regime shifts or trends that may signal deviations from expected performance parameters.
- Enhancing the interpretability of the graph structure by correlating the topological features with the process states defined by SPC zones.

By integrating the edge set of HVG with SPC zone labels, the labeled visibility graph G_{vis} provides a multifaceted view of the time series data. This

approach not only retains the temporal and sequential relationships present in the raw data but also adds a layer of process-related insight, thereby enriching the analysis and offering a more comprehensive understanding of the underlying system dynamics.

2.6 Decision Tree Model for Process Control Analysis

In our process control analysis, a decision tree model was employed, integrating both graph metrics and violation metrics to enhance its predictive power and provide a comprehensive understanding of process behavior.

- **Integration of Metrics:**
 - *Violation Metrics:* Quantify occurrences and severity of deviations from expected process behavior, serving as early warnings for potential quality issues.
 - *Graph Metrics:* Provide a deeper understanding of the network structure of the data, revealing patterns and trends beyond traditional statistical measures.
- **Model Construction:** The decision tree commences its analysis by evaluating key violation metrics such as the count of points exceeding the Upper Control Limit (UCL) and the Lower Control Limit (LCL). These metrics are pivotal in flagging potential issues indicative of a process deviation. As the model delves deeper, it incorporates graph metrics like average degree connectivity and degree assortativity, offering insights into the structural relationships within the process data.
- **Classifier Training and Evaluation:** The classifier was trained using the extracted network and violation metrics, focusing on identifying rule violations within the time series data. Its performance was evaluated using metrics such as accuracy, precision, recall, and F1-score, ensuring its efficacy in detecting embedded rule violations.

2.7 SPC Principles in Visibility Graph-Based Time Series Analysis

Following the foundational concepts of time series representation through visibility graphs, our approach seeks to enhance this analysis by integrating quantization techniques. The quantized time series is then subjected to analysis using the principles of Statistical Process Control (SPC).

SPC, a method traditionally used for monitoring and controlling processes to ensure quality and stability, provides a robust framework for interpreting the patterns within time series data. By adopting the SPC principles of pattern recognition and interpretation, our methodology extends the capabilities of visibility graphs in uncovering the underlying dynamics of the data.

In recent years, there has been a significant interest in incorporating machine learning techniques with SPC to leverage the strengths of both areas. For instance, studies have shown that machine learning can enhance the interpretative power of SPC charts [6], improve anomaly detection in manufacturing processes, and effectively reduce false alarm rates in process control [7]. Furthermore, the integration of SPC and engineering process control (EPC) has been explored, revealing challenges in control chart pattern recognition due to EPC adjustments embedding disturbance patterns within the charts. Advanced soft computing techniques, such as Extreme Learning Machine (ELM) and Random Forest (RF), have been proposed to address these challenges, demonstrating effectiveness in recognizing various disturbance patterns in SPC-EPC processes [8]. Additionally, the integration of manufacturing, inspection, and after-sales service data using machine learning algorithms has been explored for quality management in product manufacturing. This approach, employing one-class classification algorithms to address class imbalance and symbolic aggregate approximation for data segmentation, has demonstrated high efficacy in detecting defective engines in heavy machinery before shipment [9].

Our contribution to this evolving field lies in the integration of SPC patterns and machine learning methods within the framework of visibility graphs. By doing so, we not only adopt the quantization and pattern interpretation strategies of SPC but also harness the analytical prowess of modern machine learning techniques to offer a comprehensive analysis tool for time series data.

The subsequent sections will elaborate on the specifics of our approach,

including the development of labeled visibility graphs, the application of machine learning for classification and prediction, and the results of our extensive evaluation.

2.8 Transitioning from the Classical HVG-based Approach to the "Six Sigma-Based" Approach

Moving from the classical visibility graph approach to a Six Sigma-based method involves several key differences in time series analysis. Initially, the transformation of time series data into a network structure through the classical method provides foundational groundwork. The Six Sigma-based method builds upon this by incorporating statistical process control (SPC) and data classification techniques, thus refining and extending the capabilities for analysis.

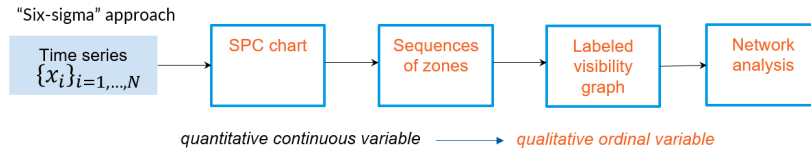


Figure 2.8: Illustration of the Six Sigma-based visibility graph methodology.

Key Differences:

- The integration of SPC charts introduces control limits and statistical thresholds for enhanced process monitoring.
- SPC zone-based discretization categorizes continuous data into distinct zones, which will be discussed in detail in the following sections.
- The construction of labeled visibility graphs from the discretized data allows for the classification of data points, providing a groundwork for further analysis.
- Machine learning integration uses network and violation metrics extracted from the visibility graphs to train models for predicting violations.

- A focus on proactive quality control through predictive modeling aids in preemptive process improvements.

These steps represent the high-level process flow of the Six Sigma-based approach, setting the stage for a more comprehensive exploration of each element in the subsequent sections.

2.9 The Six Sigma-Based Visibility Graph Method

1. **SPC Chart with Enhanced Zones:** An enhanced SPC chart includes additional zones to provide a more nuanced view of the process variation:
 - **Zone A** ($A+$, $A-$): Points falling between 2σ and the control limits (UCL or LCL), established at 3σ .
 - **Zone B** ($B+$, $B-$): Points lying between 1σ and 2σ away from the centerline, representing a moderate distance from the mean.
 - **Zone C** ($C+$, $C-$): Points within 1σ of the centerline, indicating minimal variation from the mean.
 - **Beyond Control Limits** ($UCL+$, $LCL-$): Points that exceed the UCL ($UCL+$) or fall below the LCL ($LCL-$), suggesting potential issues with the process.

The designation of "+" or "-" after the zone indicates whether the points are above or below the mean, respectively. These enhanced zones enable a deeper analysis of process performance, allowing for the detection and categorization of variations within specific ranges.

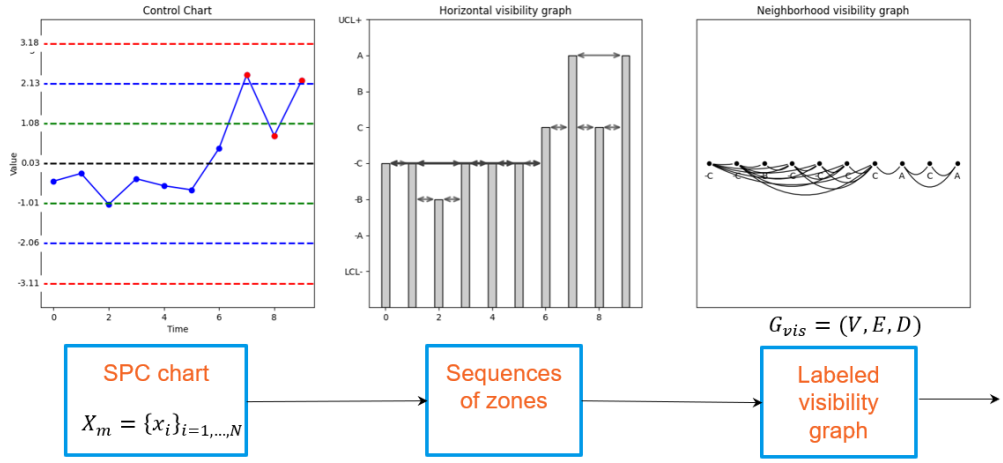


Figure 2.9: SPC chart division into zones with labeled data points.

2. **Discretization of Time Series Data:** The next step involves the discretization of time series data instances according to a specific rule system, which categorizes continuous data into the defined SPC zones. The discretization process is governed by the following categorization:

$$\hat{x}_i = \begin{cases} A+ & \text{if } x_i \text{ is between } 2\sigma \text{ and the UCL} \\ A- & \text{if } x_i \text{ is between } 2\sigma \text{ and the LCL} \\ B+ & \text{if } x_i \text{ is between } 1\sigma \text{ and } 2\sigma \text{ above the mean} \\ B- & \text{if } x_i \text{ is between } 1\sigma \text{ and } 2\sigma \text{ below the mean} \\ C+ & \text{if } x_i \text{ is within } 1\sigma \text{ of the mean and above} \\ C- & \text{if } x_i \text{ is within } 1\sigma \text{ of the mean and below} \\ UCL+ & \text{if } x_i \text{ exceeds the UCL} \\ LCL- & \text{if } x_i \text{ falls below the LCL} \end{cases} \quad (2.5)$$

This categorization transforms the continuous data points into specific tags, facilitating a more structured analysis.

3. **Constructing the Horizontal Visibility Graph:** After the SPC chart is updated with the discretized data, we proceed to construct the

horizontal visibility graph (HVG). This approach encapsulates the zonal information within each node, providing insights about data point trends without requiring their precise values. The HVG is developed from this newly created SPC chart, allowing for an in-depth analysis of the process dynamics.

4. **Construction of Labeled Visibility Graphs:** Utilizing the sequences derived from the horizontal visibility graph (HVG), we construct labeled visibility graphs, denoted as $G_{vis} = (V, E, D)$. In these graphs, each node corresponding to a data point in the time series is explicitly tagged with a label that represents its respective 'zone' within the process parameters. These zones are typically derived from domain-specific knowledge or statistical characteristics of the data, such as control limits.

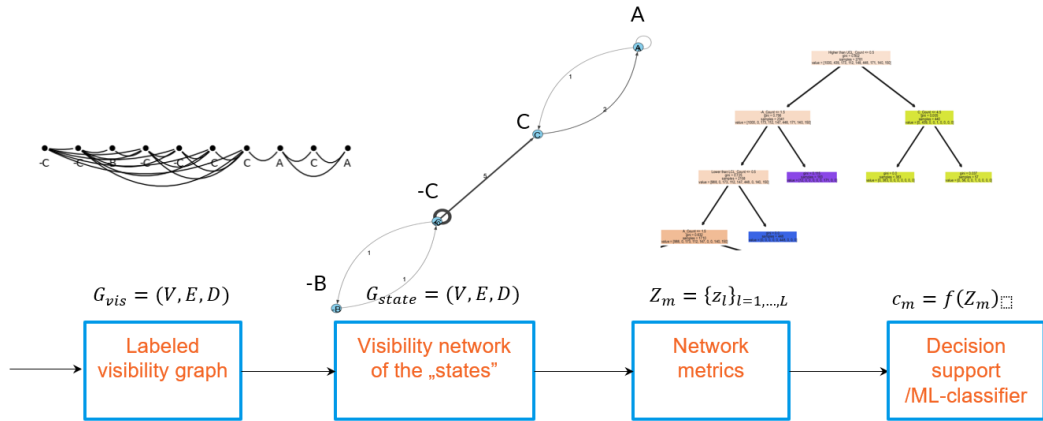


Figure 2.10: Construction of labeled visibility graphs from the horizontal visibility graph.

5. **Deriving State Networks from Labeled Visibility Graphs:** Once the labeled visibility graphs are established, we proceed to map these graphs to visibility networks of "states". Each state corresponds to a zone and its associated data points. These state networks offer a higher-level view, capturing the transitions between different zones.
6. **Extraction of Network Metrics:** We extract a comprehensive set of network metrics. These metrics are pivotal in quantifying the structure

and connectivity of the state network, thereby providing a comprehensive understanding of the dynamics of data. The metrics extracted include:

- **Density:** The measure of how many edges are in the network compared to the maximum possible number of edges between vertices.
- **Degree Assortativity:** The correlation coefficient of degree between pairs of linked nodes.
- **Average Clustering:** The average of the clustering coefficient across all nodes in the network.
- **Average Degree Connectivity:** The average number of connections per node.
- **Transitivity:** The overall probability for the network to have adjacent nodes interconnected.
- **Number of Communities:** The count of detected communities within the network.
- **Modularity:** The strength of division of a network into modules (also called groups, clusters or communities).
- **Diameter:** The longest shortest path between any two nodes in the network.

7. **Explanatory Variables:** In addition to the network metrics, we are adding other variables to enhance the predictive capability of the model. These variables provide an essential layer of detail that reflects process behavior beyond the structure revealed by network analysis alone.

- **Control Limit Variables:** Counts of observations below the Lower Control Limit (LCL) and above the Upper Control Limit (UCL), capturing the instances of statistically significant deviations from the process mean.
- **Zone-Specific Counts:** The frequency of data points within each of the designated SPC chart zones (A, B, C), and beyond control limits, delineating the extent of process variation within and outside of expected thresholds.

This comprehensive suite of explanatory variables, in conjunction with the network metrics, underpins our predictive modeling framework. They are instrumental in dissecting the contributing factors to the explained variable, *Violation*, indicative of compliance lapses within the process. The integration of these variables empowers our model to forecast 'Violations' with enhanced precision, thereby furnishing critical insights for advancing quality control measures and process optimization efforts.

8. **Application of Machine Learning for Classification and Prediction:** Following the development and analysis of the decision tree model, we further explore the application of machine learning techniques for classification and prediction. Leveraging the extracted network metrics as features, we utilize a supervised learning method to train a decision tree. This approach enables the trained classifier to effectively predict the violations in our dataset, enhancing the precision and reliability of our process control analysis.

This method combines statistical process control (SPC), network theory, and machine learning to provide a comprehensive tool for process analysis and improvement. By translating time-series data into a network context and applying advanced analytics, it allows for a deeper understanding of process dynamics and aids in making better decisions.

3. Chapter

Outcomes and Toolset Efficacy

We have developed a toolbox that is implemented in both Python and R. This toolbox is designed for data analysis and visualization, catering to different user preferences for either programming language.

3.1 Implementation and Analysis

To identify Western Electric rule violations, our analysis commenced with a dataset comprising 400 individual time series. Each series encompassed 100 observations, ensuring a consistent length across all data points. Importantly, each time series contained exactly one rule violation, providing a focused framework for our analysis.

The dataset was structured to systematically cover the four Western Electric rules across the 400 time series. Specifically, the first 100 time series were designed to violate the first rule of the Western Electric ruleset, the next 100 to breach the second rule, followed by the third 100 violating the third rule, and the final 100 breaking the fourth rule. This structured approach allowed for a comprehensive analysis of each rule's impact and characteristics within the dataset.

Furthermore, the time series data adhered to a normal distribution, a critical aspect that influences the interpretation and effectiveness of our analysis. The normal distribution of the data ensures that the results and insights derived are grounded in a common statistical framework, enhancing the reliability and applicability of our findings.

Statistical Process Control Chart Generation:

The generation of Statistical Process Control (SPC) charts in our analysis

began with a crucial preprocessing step: segmenting each time series. This segmentation was achieved through an algorithm that received several parameters: the time series itself, the mean and standard deviation of the time series, the size of the moving window, and the step size for the moving window. The default step size was set to 1, meaning the moving window advanced one point at a time, allowing for a detailed and gradual analysis of the time series.

The segmentation process involved calculating the Upper Control Limit (UCL) and Lower Control Limit (LCL) for each segment, based on the mean and standard deviation of the time series. These control limits are critical for identifying potential rule violations within each segment and are essential for the subsequent analysis.

Following the segmentation and control limit determination, our process involved iterating over each segment to identify rule violations. This step was essential, especially for training purposes, as it provided a ground truth against which the model's predictions could be evaluated. The identified violations for each segment were recorded, creating a comprehensive record of rule breaches.

The final step before generating the SPC charts was the visualization of control charts for each segment. This visualization process involved highlighting the rule violations on the chart, enhancing interpretability and providing a clear visual representation of where and how the rules were violated within each segment.

Initially, we generated SPC charts for each time series, applying various moving window sizes to capture different aspects of the data. We particularly focused on a 30-observation window size, as illustrated in Figure 3.1. This window size was chosen for its ability to provide a balanced view of both immediate and slightly longer-term trends in the data, making it an effective tool for our analysis.

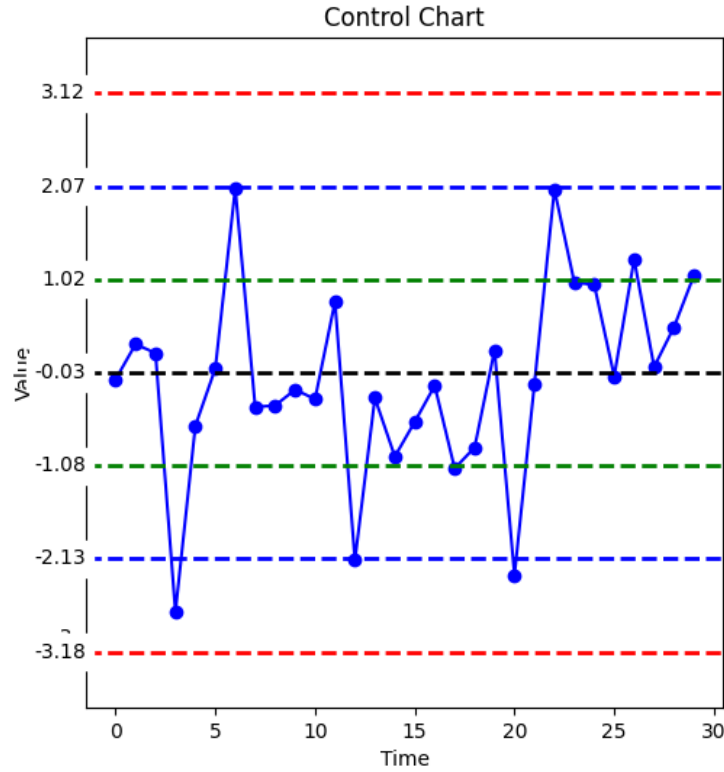


Figure 3.1: Control chart with 30 moving window sizes example.

Discretization of Time Series Data:

Following the generation of the SPC charts, the next critical step in our analysis was the discretization of the time series data. This process, performed by the function, involved transforming the continuous data of each segment into discrete categories. This function not only plots the segment but also carries out the essential task of discretization, aligning the data with the defined zones of the SPC chart.

The discretization process is crucial as it converts the continuous fluctuations of the time series into a format that is more manageable and interpretable for subsequent analysis steps. By categorizing the data points into distinct zones based on the SPC chart, we can more effectively identify patterns and anomalies within the time series.

The outcome of this discretization process is visually represented in the SPC charts, where the quantized data is clearly delineated. This visualization, as shown in Figure 3.2, provides a clear and intuitive understanding of how

the continuous data has been segmented into discrete categories, facilitating further analysis and interpretation.

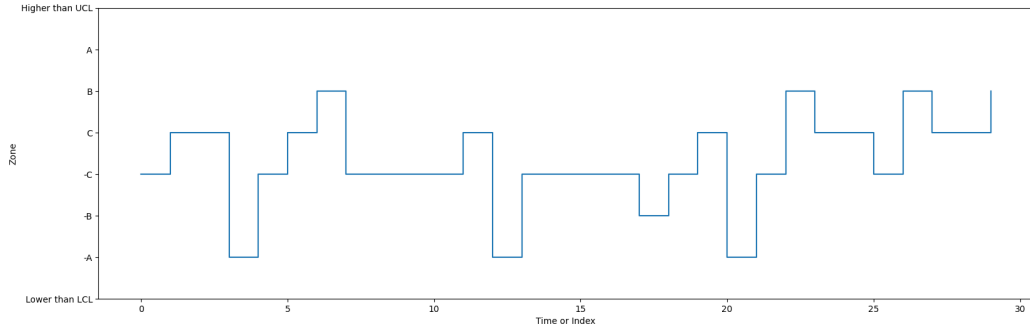


Figure 3.2: The chart after discretization.

Construction of Horizontal Visibility Graphs:

Following the discretization of the time series data, the next step in our analysis was the construction of horizontal visibility graphs (HVGs). This function plays a pivotal role in transforming the discretized data into a graphical format, where the relationships between different data points are visually represented.

Horizontal visibility graphs are a powerful tool for understanding the structural properties of time series data. In these graphs, each data point is represented as a node, and edges are drawn between nodes if they are 'visible' to each other, based on certain criteria defined by the horizontal visibility algorithm. This method of representation allows us to observe and analyze the underlying patterns and connections within the data.

Figure 3.3 showcases an example of a horizontal visibility graph constructed from one of our discretized time series segments. This figure illustrates how the categorized data is transformed into a network of interconnected nodes, offering insights into the structure and dynamics of the time series.

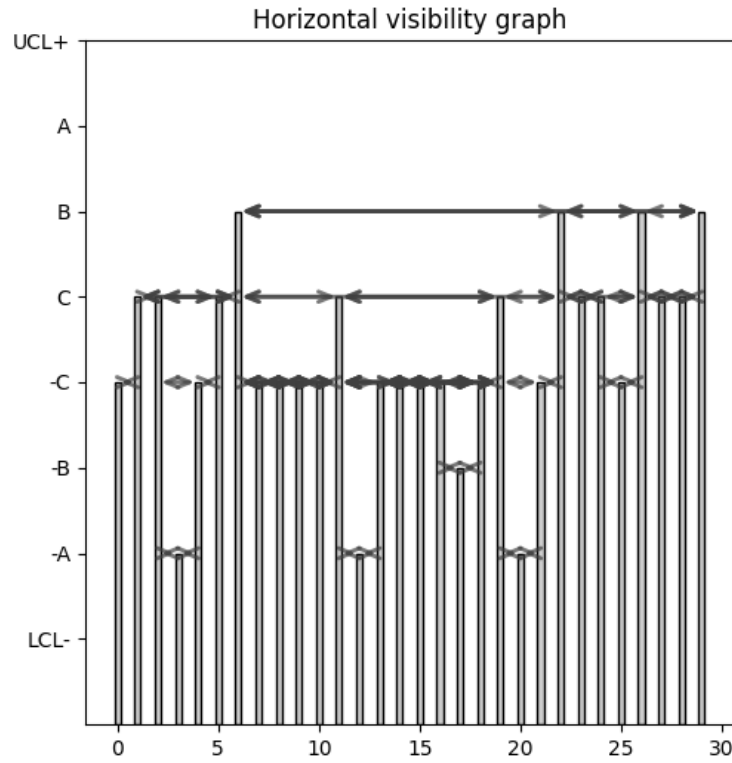


Figure 3.3: Horizontal visibility graph for the example 30 moving window.

Creation of Labeled Visibility Graphs:

After constructing the horizontal visibility graphs, we moved on to the creation of labeled visibility graphs. This process involved two key steps: first, generating the graph and then visualizing it with the plot function.

In labeled visibility graphs, each node is assigned a label corresponding to its zone in the SPC chart. This labeling significantly enhances the interpretability of the graph by providing additional context about each data point. For instance, nodes can be labeled as belonging to upper control limits, lower control limits, or other specific zones defined in the SPC chart. This categorization allows for a more nuanced understanding of the relationships and patterns within the time series data.

Following the creation of the graph, the plot function is used for visualization. This function not only displays the graph but also incorporates arrows to indicate the direction of transitions between different states or zones in the time series. This directional aspect adds another layer of analysis, helping to

understand the sequence and flow of events or conditions within the data.

Figure 3.4 illustrates an example of a labeled visibility graph created using this process. The figure demonstrates how the labeled nodes and directional arrows provide a comprehensive view of the transitions and interactions within the time series, offering deeper insights into its dynamics.

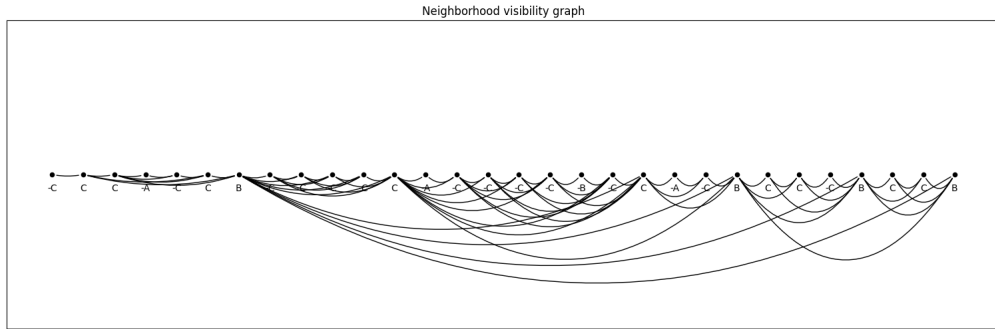


Figure 3.4: Labeled visibility graph.

Development of Visibility Network of States:

The culmination of our analysis was the development of the Visibility Network of States, a crucial step in understanding the complex dynamics of our time series data. This network graph is unique in that it illustrates the weighted relationships between different states or zones within the time series, providing a more nuanced view of the interactions and transitions.

To construct this network, we employed two different functions, each utilizing a distinct Python package. The first function uses the NetworkX package, while the second function, utilizes the iGraph package. Both functions take the labeled visibility graph, denoted as 'G_zone', as input. We are using two different package for better readability in different cases.

The 'G_zone' graph, which was previously created and labeled in the earlier steps of our analysis, serves as the foundation for the Visibility Network of States. The two functions mentioned above transform 'G_zone' into a weighted graph format. In this format, the edges between nodes (representing different states or zones) are assigned weights. These weights signify the strength or frequency of transitions between the states, offering valuable insights into the predominant patterns and trends within the time series.

Figure 3.5 presents an example of the Visibility Network of States created using these methods. This figure highlights the weighted relationships between different states, visually encapsulating the intricate dynamics of the time series data. The network graph serves as a powerful tool for visualizing and understanding the complex interactions and dependencies within the data, ultimately enhancing our analytical capabilities.

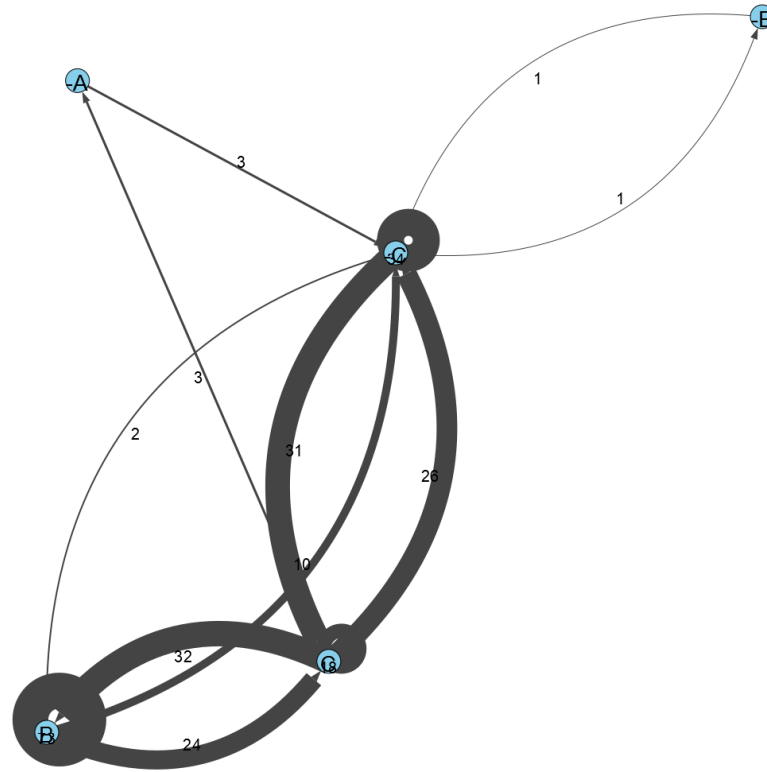


Figure 3.5: Visibility network of the states graph.

Upon completion of these steps, we systematically saved the network metrics and violation metrics.

Configuration of the Decision Tree Classifier:

Following the creation of the visibility graphs and networks, we configured the Decision Tree Classifier, leveraging the `DecisionTreeClassifier` of the Scikit-Learn library. This stage was vital in translating the complex data structures into actionable insights. Key configurations for the classifier included:

- **Minimum Samples per Leaf:** Set to 30, to prevent overfitting by

ensuring a sufficient number of samples in each leaf of the tree.

- **Maximum Features:** Restricted to 10, to prioritize the most relevant features at each decision node, enhancing the decision-making process.
- **Criterion:** Employed the Gini impurity as the split criterion, aiding in creating homogeneous nodes and effective data classification.

Visualization of Decision Tree:

To elucidate the decision-making process of the trained classifier, we visualized the decision tree structure. This visualization (Figure 3.6) provided insights into the decision paths and feature importance, crucial for understanding and refining the performance of the classifier.

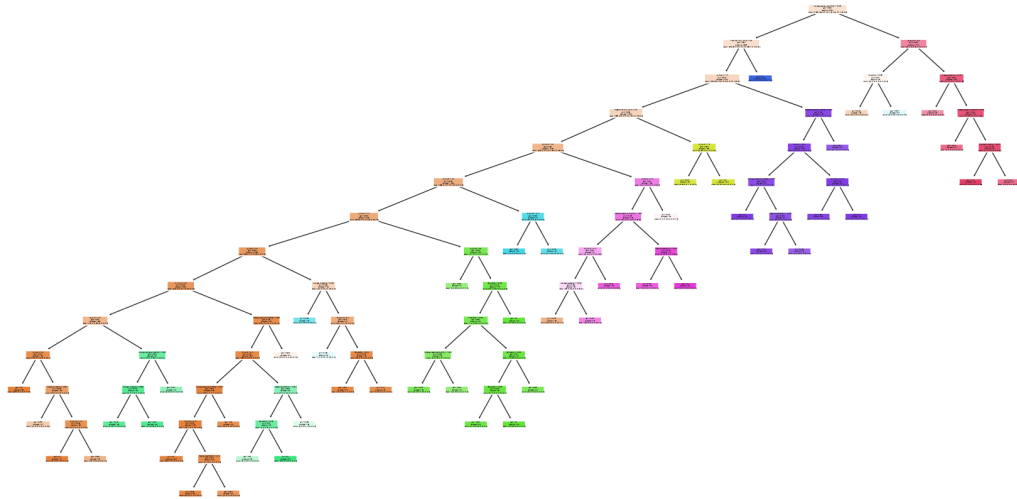


Figure 3.6: Visualization of the trained decision tree.

Concluding this phase, the training and configuration of the decision tree classifier underscored its critical role in enhancing the predictive capabilities of our toolbox, particularly in analyzing time series data for rule violation detection.

3.1.1 Evaluation of Classifier Performance

A critical component of our analysis was the evaluation of the performance of the classifier. To this end, a confusion matrix was constructed, which is a table

used to describe the performance of a classification model on a set of test data for which the true values are known.

Interpreting the Confusion Matrix:

The confusion matrix allows us to visualize the accuracy of the classifier in predicting rule violations. The matrix is divided into sections, each representing a different rule violation scenario, both above and below the mean, as per the Western Electric ruleset. The Y-axis represents the actual instances of rule violations, while the X-axis shows the predictions made by the classifier. The values along the diagonal of the matrix represent correct predictions, where the predicted category matches the actual category. Off-diagonal numbers represent misclassifications.

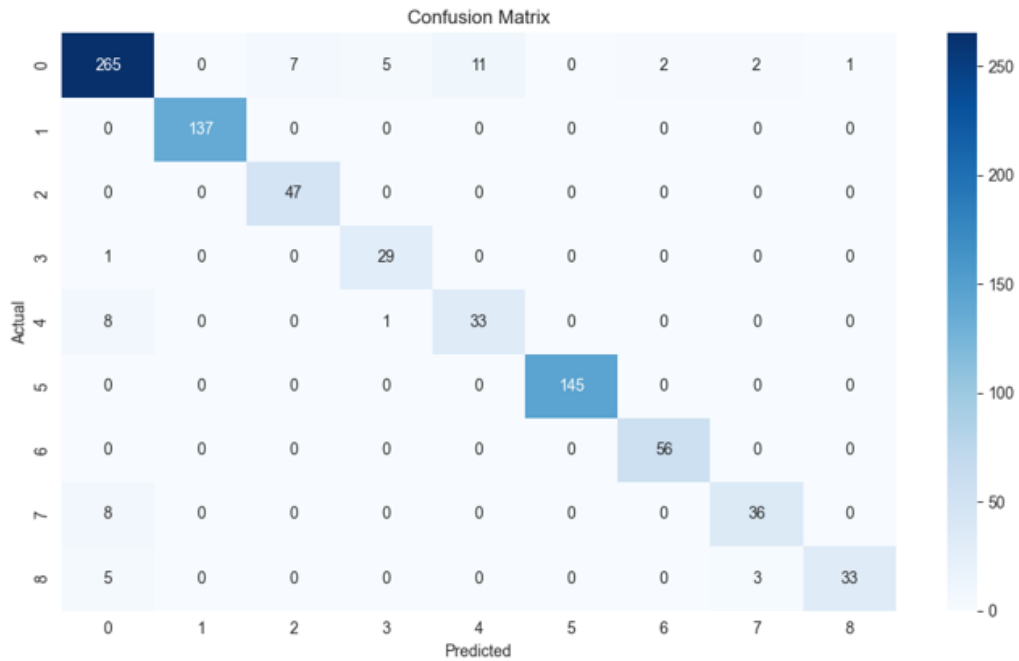


Figure 3.7: Confusion matrix illustrating the performance of the decision tree classifier.

Analysis of Confusion Matrix Results:

For instance, the value at the coordinate (0,0) in the matrix indicates the number of times the model correctly predicted a non-violation scenario. Similarly, the value at (1,1) shows how often the model accurately predicted a

first-rule violation above the mean. Misclassifications can be observed in off-diagonal values, such as (5,0), which would indicate the model incorrectly predicted a non-violation where there was actually a first-rule violation below the mean.

The confusion matrix is particularly telling for the performance of the model across different types of violations. For example, we can observe from the matrix that:

- The classifier is highly accurate in predicting non-violations (0) and first-rule violations above the mean (1).
- It also shows a commendable accuracy for third-rule violations under the mean (7).
- Misclassifications mainly occur adjacent to the true category, suggesting that while the model may confuse the type of rule violation, it generally predicts the presence of a violation correctly.

This matrix serves as a pivotal tool for fine-tuning our model. By analyzing the patterns of misclassification, we can identify areas where the model may require further training or where feature engineering may enhance its predictive accuracy.

Feature Importance Analysis:

To complement the predictive modeling, we executed a feature importance analysis using the Boruta algorithm alongside a Random Forest classifier. This method is an all-relevant feature selection technique, which aims to identify and confirm the most relevant attributes for the model's decision-making process.

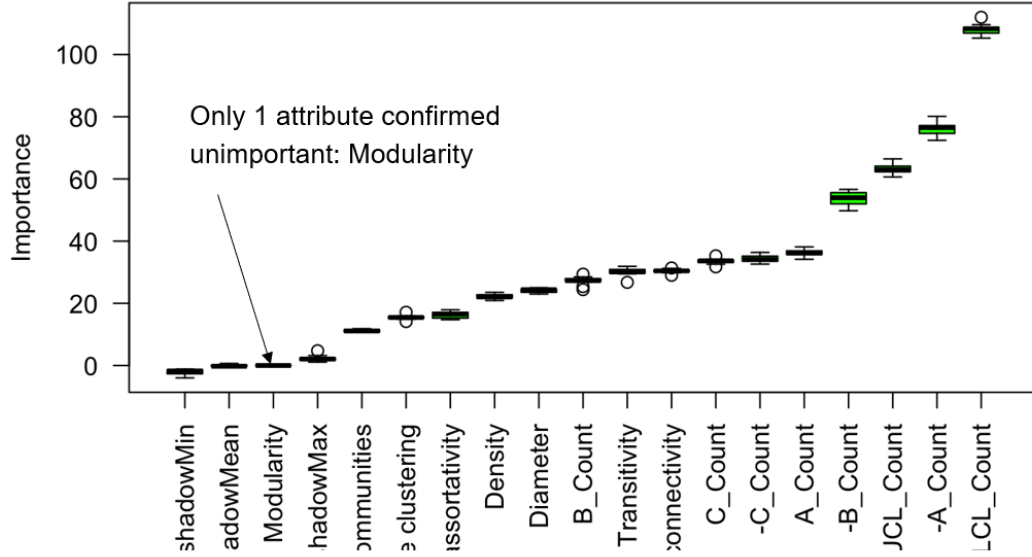


Figure 3.8: Feature importance scores obtained from the Random Forest classifier using the Boruta algorithm.

The Boruta analysis robustly concluded that, with the exception of Modularity, each attribute significantly contributes to the model’s accuracy. In particular, features like ‘C_Count’, ‘A_Count’, and ‘LCL_Count’ emerged as highly influential, indicating that the frequency of points within certain SPC zones plays a pivotal role in the model’s predictions. Notably, these counts correspond to the control limits and zones established in our SPC chart, reinforcing the importance of these zones in rule violation detection.

This feature importance analysis provides an in-depth understanding of the internal mechanics of the classifier and affirms the relevance of the selected features to the model’s overall predictive capability. The insights gained from this analysis enhance the analytical strength of our toolbox, supporting its application in process control and improvement initiatives with a data-driven approach.

3.2 Toolbox Capabilities and Implementation

Our toolbox, designed for time series analysis, incorporates several key functionalities:

- **SPC Chart Construction:** Capable of generating Statistical Process Control charts from time series data, incorporating detailed zonal structures including standard and specialized zones (A+, A-, B+, B-, C+, C-, UCL+, LCL-).
- **Discretization of Time Series Data:** Transforms continuous time series data into discrete categories based on the SPC chart zoning, facilitating structured analysis.
- **Horizontal Visibility Graph Creation:** Constructs horizontal visibility graphs using the discretized data, where nodes are represented based on the categorized zones.
- **Graph-Based Data Analysis:** Employs the visibility graphs for network analysis, extracting and saving network and violation metrics.
- **Integration of Machine Learning Techniques:** Incorporates machine learning methods for predictive modeling and analysis based on the structured data.

4. Chapter

Conclusions

In the course of this research, we have successfully integrated the principles of Statistical Process Control (SPC) with the analytical capabilities of Visibility Graphs (VisGraphs) to establish a robust methodological framework for time-series data analysis. This synergy has not only fortified the interpretability of process behaviors but also paved the way for novel analytical techniques.

The fusion of SPC and VisGraphs has culminated in several key achievements:

1. A novel quantization strategy that coherently transforms continuous time-series data into a discretized, manageable structure, effectively reducing its complexity while preserving intrinsic patterns.
2. The innovation of utilizing quantized data as categorical nodes within the Horizontal Visibility Graph (HVG), thereby enhancing the interpretability of the temporal data structure.
3. The elucidation of the architecture of network through the sequence of labels, which unraveled the probabilistic underpinnings of the data and facilitated the extraction of significant metrics, such as average run lengths.
4. The development and implementation of a decision-tree classifier that leverages these metrics, validating the precision and analytical acumen of the method.
5. Improved data visualization techniques within the VisGraphs that bolstered the understandability and allowed for immediate and intuitive comprehension of complex relationships within the data.

Conclusions

This integrative approach transcends traditional analysis and invites a variety of practical applications, particularly in industrial settings where time-series data is prevalent. The visual and analytical clarity achieved here opens up new avenues for predictive analytics and operational optimization.

Bibliography

- [1] Mutua Stephen, Changgui Gu, and Huijie Yang. Visibility graph based time series analysis. *PLOS ONE*, 10(11):1–19, 11 2015.
- [2] Tao Wen, Huiling Chen, and Kang Hao Cheong. Visibility graph for time series prediction and image classification: a review. *Nonlinear Dynamics*, 110(4):2979–2999, 2022.
- [3] Shijie Yan and Danling Wang. Time series analysis based on visibility graph theory. In *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 311–314, 2015.
- [4] Speranza Sannino, Sebastiano Stramaglia, Lucas Lacasa, and Daniele Marinazzo. Visibility graphs for fmri data: Multiplex temporal graphs and their modulations across resting-state networks. *Network Neuroscience*, 1(3):208–221, 2017.
- [5] Bartolo Luque, Lucas Lacasa, Fernando Ballesteros, and Jordi Luque. Horizontal visibility graphs: exact results for random time series, 2010.
- [6] Phuong Hanh Tran, Adel Ahmadi Nadi, Thi Hien Nguyen, Kim Duc Tran, and Kim Phuc Tran. *Application of Machine Learning in Statistical Process Control Charts: A Survey and Perspective*, pages 7–42. Springer International Publishing, Cham, 2022.
- [7] Shih-Hsiung Chou, Shing Chang, Tzong-Ru Tsai, Dennis KJ Lin, Yunfei Xia, and Yu-Siang Lin. Implementation of statistical process control framework with machine learning on waveform profiles with no gold standard reference. *Computers & Industrial Engineering*, 142:106325, 2020.

- [8] Yuehjen E Shao and Chih-Chou Chiu. Applying emerging soft computing approaches to control chart pattern recognition for an spc–epc process. *Neurocomputing*, 201:19–28, 2016.
- [9] Taehoon Ko, Je Hyuk Lee, Hyunchang Cho, Sungzoon Cho, Wounjoo Lee, and Miji Lee. Machine learning-based anomaly detection via integration of manufacturing, inspection and after-sales service data. *Industrial Management & Data Systems*, 117(5):927–945, 2017.