Symbolic Time Series Analysis for Anomaly Detection

Ibrahim RIDENE ibrahim.ridene.20@gmail.com
Lorenzo DUNAU lorenzo.dunau@gmail.com

Master MVA, ENS Paris-Saclay
Dec 17 2024

GITHUB link of the project: **click here**

Contents

1 Introduction and contributions

This project aims to provide a comprehensive exploration of symbolic time series analysis (STSA) for anomaly detection, as introduced in [1].
Section 2 begins with the foundational techniques used in anomaly detection, including statistical methods like Principal Component Analysis (PCA) and neural network-based approaches such as Multilayer Perceptron Neural Networks (MLPNN) and Radial Basis Function Neural Networks (RBFNN). These methods provide a comparative baseline for understanding the strengths and limitations of STSA.

Following this, we delve into symbolic techniques, specifically the Symbolic False Nearest Neighbors (SFNN) and Wavelet Space (WS) methods. These approaches highlight how symbolic representations capture the coarse-grained dynamical behavior of complex systems. The D-Markov machine is then introduced, providing a framework for anomaly detection based on finite-state machine modeling.

With these methods established, Section 3 describes the dataset used in our experiments, which involves time series data from a nonlinear electronic system called Duffing Problem. The experiments, presented in Section 4, evaluate the performance of the D-Markov machine and symbolic techniques under various configurations, including the impact of the Markov order D. The results showcase the potential of symbolic dynamics in early anomaly detection and robustness to noise.

Finally, in Section 5, we will conclude concerning our approach by providing an overview of what has been done, what the limitations are, and possible future work.

All the code and plots can be found on GitHub: **Click here**.

During the implementation of this project, the contributions of each student are as follows:

- Ibrahim RIDENE
  - Implementation of the Wavelet Space (WS) partitioning algorithm
  - Implementation of the D-Markov Machine algorithm from scratch by generating states, computing stationary vector and calculating the anomaly mesures based on KL divergence.
  - Implementation of the Principal Component Analysis (PCA) pipeline
  - Implementation of the Radial basis Function Neural Networks (RBFNN) pipeline
  - Implementation of the Multilayer Perceptron Neural Network (MLPNN) pipeline
  - Implementation of the Data generation pipeline based on Duffing Equation.
- Lorenzo DUNAU
  - Implementation of the data preprocessing pipeline.
  - Implementation of the Symbolic False Nearest Neighbors (SFNN) algorithm.
  - Implementation delay impact pipeline for SFNN

In this project, **no source code was available for the paper**, so we implemented everything from scratch and replicated the experiments entirely based on the descriptions provided. Initially, we focused on reproducing the original experiments as accurately as possible. Furthermore, we demonstrated the delay impact of anomaly detection for SFNN. Subsequently, we extended the study to evaluate the impact of different symbolization methods (for WS) on anomaly detection performance. Additionally, we analyzed the influence of the Markov order D on the results, providing insights beyond the scope of the original article.

2 Method

In this section, we present the various anomaly detection methods introduced in the paper, including both traditional techniques like Principal Component Analysis (PCA) and neural networks, as well as symbolic methods such as SFNN, WS, and the D-Markov machine. These approaches are explored in the context of their ability to identify anomalies in nonlinear dynamical systems.

2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is used to detect anomalies by projecting high-dimensional time series data into a lower-dimensional subspace defined by the eigenvectors of the covariance matrix. The sampled time series of size l is reshaped into a $d \times n$ data matrix, from which the covariance matrix is computed. The resulting feature matrix is given by:

$$\tilde{M} = \left[\sqrt{\frac{\lambda_1}{\sum_{k=1}^{n} \lambda_k}} v_1, \ldots, \sqrt{\frac{\lambda_q}{\sum_{k=1}^{n} \lambda_k}} v_q \right],$$

where λ_i are the eigenvalues, v_i are the corresponding eigenvectors, and q is chosen such that

$$\frac{\sum_{i=q+1}^{n} \lambda_i}{\sum_{i=1}^{n} \lambda_i} < Z,$$

where Z is a predefined threshold.

Under nominal (stationary) conditions, the matrix $\tilde{M}_{\text{nom}}$ is computed. At subsequent time steps $t_1, t_2, \ldots$, new feature matrices $\tilde{M}_1, \tilde{M}_2, \ldots$ are obtained using the same parameters (l, d, n, q). The anomaly measure at each time step is defined as:

$$\mathcal{M}_k = d(\tilde{M}_k, \tilde{M}_{\text{nom}}),$$

where $d(\cdot, \cdot)$ is an appropriately defined distance function that quantifies the deviation of the observed data from the nominal condition.

2.2 Multilayer Perceptron Neural Network (MLPNN)

The MLPNN method in the paper is designed to detect anomalies by training a neural network using the mean-square error criterion. The input time series data is segmented and transformed into input vectors of dimension d, with the target output vector set to zero. The network's weight matrix $\mathbf{w}^k(n)$ is updated iteratively using the backpropagation algorithm:

$$\mathbf{w}^k(n+1) = \mathbf{w}^k(n) - \alpha^k \mathbf{g}^k(n),$$

where $\mathbf{g}^k(n)$ is the error gradient and α^k is the learning rate. Anomaly measures are computed as the distance between the performance vectors of the nominal condition and test cases at various slow-time epochs.

2.3 Radial Basis Function Neural Networks (RBFNN)

The RBFNN method in the paper detects anomalies using a radial basis function defined as:

$$f(y) = \exp\left(-\frac{\sum_k |y_k - \mu|^\alpha}{N \theta_\alpha} \right),$$

where μ is the center and θ_α is the α-th central moment of the dataset. For anomaly detection, the mean μ and moment θ_α are calculated as:

$$\mu = \frac{1}{N} \sum_{k=1}^{N} y_k, \quad \theta_\alpha = \frac{1}{N} \sum_{k=1}^{N} |y_k - \mu|^\alpha.$$

Anomalies are quantified by comparing the RBF output of the nominal condition, f_{nom}, with the output f_k at slow-time epochs t_k, using the distance metric:

$$\mathcal{M}_k = d(f_{\text{nom}}, f_k),$$

where $d(\cdot, \cdot)$ is an appropriately defined distance function.

2.4 Symbolic False Nearest Neighbors (SFNN)

The Symbolic False Nearest Neighbors (SFNN) method optimizes the symbolic partitioning of time series by avoiding topological degeneracies in the reconstructed state space. According to Takens' theorem, the state space can be reconstructed using d consecutive observations from the time series, where, at each time step, the state vector is represented as:

$$(y(n), y(n+\delta), y(n+2\delta), \ldots, y(n+(d-1)\delta))$$

This provides a d-dimensional representation of the state space. The challenge is determining the appropriate value of d. A small d may lead to poor partitions, with more false nearest neighbors due to the insufficient representation of the underlying dynamics. Conversely, increasing d beyond a certain point becomes unnecessary, as chaotic processes typically operate in lower-dimensional spaces.

To address this, the False Nearest Neighbors (FNN) method is employed. The idea is to start with a small value for d and create a partition of the state space. For each point in the reconstructed state space, its nearest neighbor is identified. If a significant number of points have nearest neighbors with different labels (symbolic assignments), this suggests that the dimensionality d is too low, and thus, the dimension is increased. This process is repeated until the number of false nearest neighbors falls below a satisfactory threshold, indicating that the chosen dimension adequately captures the system's dynamics.

The partitioning of the state space is performed using the k-Means algorithm, which clusters the points based on their distances in the d-dimensional space.

2.5 Wavelet Space (WS) Method

The Wavelet Space (WS) method leverages the time-frequency analysis of the time series data to extract relevant features for anomaly detection. The time-localized signal is first analyzed to identify the dominant frequencies, followed by wavelet decomposition to generate coefficients corresponding to these frequencies. A mother wavelet, such as Daubechies (db1), is chosen based on the dynamical behavior of the system. The wavelet coefficients are stacked and partitioned into symbolic segments, with the number of segments equal to the alphabet size. The symbolic sequences are then used to detect anomalies by comparing transitions in the nominal and test conditions.

2.6 D-Markov Machine

The D-Markov machine is a finite-state automaton constructed as a D-th order Markov chain. It captures patterns in symbolic time series data by constructing states and transitions based on a sliding window of length $D+1$ over a sequence of symbols. The states are defined as:

$$q(D, s) = \{S^- \in S^- : S_D^- = s\},$$

where S_D^- represents the last D symbols of the past sequence S^-. The set of all states is given by:

$$Q(D) = \{q(D, s) : s \in S_D\},$$

with $|Q(D)| = |A|^D$, where A is the symbol alphabet.

The transition probabilities between states q_j and q_k are computed using frequency counts:

$$\pi_{jk} = \frac{N(s_{i_1} \cdots s_{i_D} s)}{N(s_{i_1} \cdots s_{i_D})},$$

where $N(\cdot)$ represents the frequency of occurrences of the corresponding symbol sequences.

The stationary probability vector $\mathbf{p}_{\text{nom}}$, representing the steady-state probabilities of the states under nominal conditions, is obtained as the left eigenvector of the transition matrix Π_{nom} corresponding to the unit eigenvalue.

Anomalies are detected by comparing the stationary probability vector of the test condition $\mathbf{p}_k$ at slow-time epochs t_k with the nominal vector $\mathbf{p}_{\text{nom}}$. The anomaly measure is defined as:

$$\mathcal{M}_k = d(\mathbf{p}_k, \mathbf{p}_{\text{nom}}),$$

where $d(\cdot, \cdot)$ is an appropriately defined distance function, such as the KL divergence or cosine similarity.

3 Data

The data utilized in this project is synthetic and generated based on the Duffing equation, which describes a nonlinear dynamical system. This equation models the time evolution of an electronic oscillator under the influence of a damping parameter β, a driving amplitude A, and a driving frequency ω, with zero initial conditions. The Duffing equation is given as:

$$\ddot{y} + \beta\dot{y} + y + y^3 = A\cos(\omega t),$$

where $\ddot{y}$ represents the second derivative of y with respect to time (acceleration), $\dot{y}$ is the first derivative (velocity), and y is the displacement. For each β value in the range $[0.10, 0.35]$, the system evolves over 27 seconds with a sampling rate of 100 Hz, resulting in a time series of 2700 data points. Data generation equations and plots are described in the appendix 6.1

- Preprocessing Steps

- **Normalization**: The generated time series is normalized using the z-score method:

$$z = \frac{x - \mu}{\sigma},$$

  where x is the original data point, μ is the mean, and σ is the standard deviation of the time series. This ensures that all data are centered around zero with unit variance.

- **Segmentation**: Each normalized time series is divided into 270 non-overlapping segments of length 10:

$$\text{Segment}_i = \{y_{10i}, y_{10i+1}, \ldots, y_{10i+9}\}, \quad i = 0, \ldots, 269.$$

  These segments provide localized snapshots of the system's dynamics, essential for symbolic representation and feature extraction.

- Observations:

The synthetic data captures the dynamics of the Duffing oscillator as β varies. For lower β, the system exhibits periodic behavior, while higher β values lead to more complex, potentially chaotic dynamics. On the other hand, normalization standardizes the time series across different β values, ensuring that anomalies arise purely from changes in system dynamics and not scale differences. Then, Segmenting the data into smaller chunks enhances the ability to detect localized deviations, which are critical for methods like symbolic representation and D-Markov machines.

4 Results

4.1 Numerical Simulation

To simulate the experiment in [1], we fixed the different hyperparameter values (order of Markov D, range of values β, number of test samples...) to the default ones defined in Section 4 of the paper. We also fixed the PCA threshold to Z=0.05. Additionally, the defined distance metric for PCA, MLPNN and RBFNN to MSE and for D-Markov machine methods (SFNN and WS) to KL divergence. We implemented the SAX (Symbolic Aggregate approXimation) symbolization technique [2] for the D-Markov machine methods.

The plot of the result of the evolution of the anomaly measures based on the value of β shows that:

- All five plots show gradual increase in the anomaly measure M for b in the approximate range of 0.10–0.25, followed by an abrupt increase in the anomaly measure in the vicinity of b 0.29 when a (possible) bifurcation takes place.

- The performance of the MLP and RBF neural network methods is better than that of the PCA method, which aligns with the results of the paper.

- The performance of the D-Markov methods with WS partitioning is significantly superior to that of the remaining three methods, which is also aligns with the results of the paper.

- The best performance was detected for the D-Markov Machine with WS partitioning which is clearly superior to the remaining three pattern recognition techniques from the perspectives of early detection of anomalies.

- The only difference between the results of our experiment and those found in the paper is that the paper might use a normalized threshold or saturation mechanism to cap anomaly measures after the bifurcation point at $\beta = 0.29$, whereas our implementation lacks such a mechanism, causing a further evolution of values.

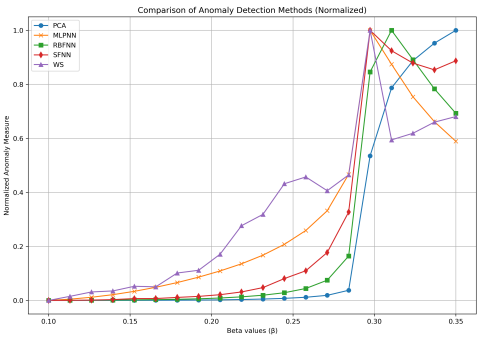

Figure 1: Results of our simulation

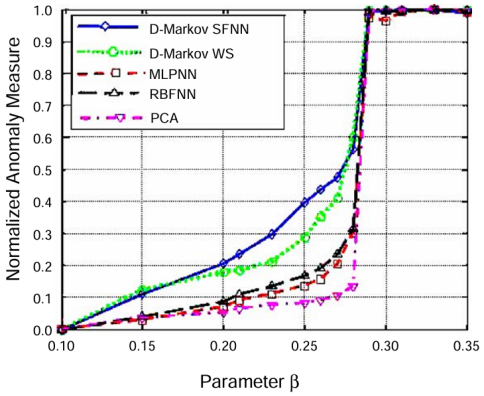

Figure 2: Results of the paper

Figure 3: Comparison between our results and those presented in the paper

4.2 Impact of delay

There appears to be an optimal delay when applying the Symbolic False Nearest Neighbors (SFNN) method. Interestingly, there is no straightforward correlation between delay and performance, but empirical results suggest that a delay of 18 is optimal. This observation can be explained by considering the pulsation (frequency) of the signal.

For this explanation, we use a Duffing signal with an angular frequency of $\omega = 5\,\text{rad/s}$ and a sampling rate of 100 samples per second. With these parameters, we obtain 125 samples per period. This corresponds to approximately $7 \times$ delay, where the chosen delay ensures that the symbolic partitioning accounts for the periodicity of the original signal.

In this context, the delay ensures that the reconstructed state space effectively captures the dynamics of the signal. Specifically, a delay of 18 leads to a partitioning that aligns well with the periodic components of the signal, making the symbolic sequences more meaningful for analysis.

This is a key observation: when choosing an appropriate delay, it is crucial to consider the underlying frequency characteristics of the signal to ensure that the partition reflects the natural periodicity.

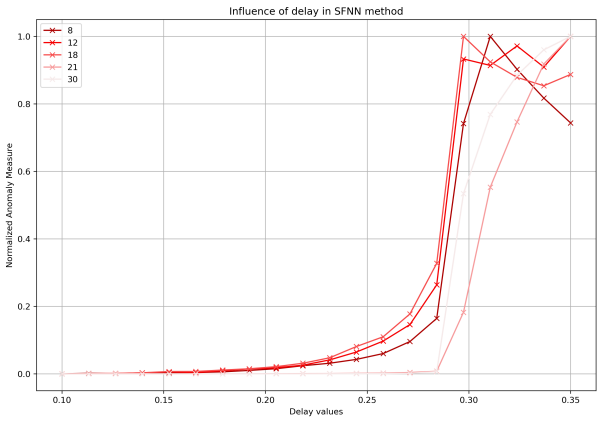

Figure 4: Influence of delay

4.3 Impact of type of Symbolization

In this section, we compare the performance of two symbolization techniques, **SAX** and **Linear** implemented for the Wavelet Space (WS) method, which was chosen due to its superior performance in early detection of anomalies based on the previous section. The comparison is illustrated in the figure, showing anomaly measures for varying values of β.

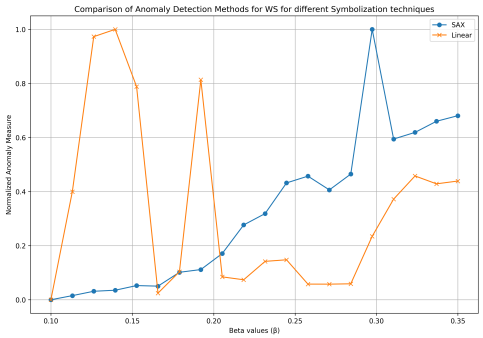

Figure 5: Impact of Symbolization technique on anomaly detection for WS method

The SAX symbolization technique produces a smoother anomaly measure curve. This is because SAX uses a **Gaussian distribution** to determine breakpoints, aligning the symbolic sequences with the global statistical properties (mean and variance) of the wavelet coefficients. By preserving global information, SAX ensures that the symbolic sequences are less sensitive to noise and outliers in the data, resulting in more consistent anomaly measures.

In contrast, linear symbolization divides the coefficient range into **equal-width bins**, disregarding the statistical distribution of the data. Wavelet coefficients, which often follow non-uniform distributions, are poorly represented by linear partitioning. This leads to increased sensitivity near bin boundaries, causing frequent symbol changes and introducing artifacts in the symbolic sequences. Consequently, the anomaly measure curve for linear symbolization exhibits numerous peaks and variability, as seen in the plot.

Overall, the results highlight the advantage of SAX symbolization in capturing the underlying dynamics of the system with less noise and greater consistency, making it better suited for robust anomaly detection in the WS method.

4.4 Impact of Markov Order D

This section analyzes the performance of the Wavelet Space (WS) method for anomaly detection with varying Markov orders D, as shown in the figure. The Markov order D determines the size of the state window used to construct the transition matrix in the D-Markov machine. Higher values of D capture more complex temporal dependencies in the symbolic sequences, which can influence the anomaly detection capabilities.

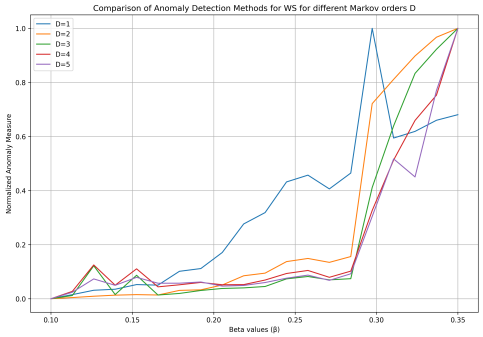

Figure 6: Impact of Markov order on anomaly detection for WS method

For small D values (e.g., $D = 1$), the anomaly measures show early increases, allowing for the detection of the bifurcation point at $\beta = 0.29$ sooner than higher D models. This is because smaller

D values are highly sensitive to immediate changes in the symbolic sequences, making them effective at early detection. However, as D increases, the model incorporates longer temporal contexts, which smooths out short-term variations and delays the detection of anomalies. While higher D values (e.g., $D = 4$ or $D = 5$) provide more stable anomaly measures, they lose sensitivity to early deviations due to the following factors:

- Increased Temporal Context: Longer state sequences reduce sensitivity to short-term changes, prioritizing stability over early detection.
- Overfitting to Normal Dynamics: Larger D values capture finer details of the nominal condition, potentially masking the onset of anomalies.
- Noise Suppression: Higher D values smooth out noise but can suppress subtle deviations signaling early anomalies.
- Transition Matrix Sparsity: The state space becomes sparser with higher D, reducing sensitivity to small changes in dynamics.

Overall, while $D = 1$ excels at early anomaly detection, higher D values offer greater robustness and stability at the cost of delayed detection. The choice of D depends on the specific requirements of the anomaly detection task, balancing sensitivity and stability. Based on the results, $D = 3$ or $D = 4$ provides a good trade-off for robust anomaly detection in the Duffing system.

5 Conclusion

In this project, we successfully implemented the methods described in the paper from scratch, replicating the experiments and analyzing the results. Our implementation was compared against the results reported in the paper, highlighting both consistencies and differences. Notably, the absence of a thresholding or saturation mechanism in our implementation led to subtle discrepancies in the anomaly measures after the bifurcation point at $\beta = 0.29$. Despite this, the overall trends and insights were consistent with those reported in the paper, validating the robustness of our implementation.

Also, we demonstrated that an optimal delay of 18 effectively captures the signal's periodic dynamics, aligning symbolic sequences with its natural frequency. This highlights the importance of choosing delay values that reflect the signal's inherent periodicity for meaningful partitioning and anomaly detection.

Additionally, we explored the impact of symbolization techniques on the Wavelet Space (WS) method, which demonstrated superior performance in early anomaly detection. The SAX symbolization method provided smoother and more consistent anomaly measures due to its alignment with the statistical properties of the data, while the linear symbolization method introduced artifacts and noise, resulting in less reliable detection. This analysis highlighted the critical role of symbolization in enhancing anomaly detection capabilities.

Finally, we examined the influence of the Markov order D in the D-Markov machine. Smaller D values, such as $D = 1$, were effective at early detection of anomalies due to their sensitivity to short-term changes. However, increasing D improved stability and robustness by capturing longer temporal dependencies, albeit at the cost of delayed anomaly detection. The trade-off between sensitivity and stability underscores the importance of selecting an appropriate Markov order based on the specific requirements of the application.

Overall, this project provided valuable insights into the implementation and performance of advanced anomaly detection methods, as well as the role of symbolization and Markov order in enhancing detection capabilities. These findings pave the way for further exploration and optimization of these methods in more complex and diverse applications.

References

[1] Shin C. Chin, Asok Ray, and Venkatesh Rajagopalan. "Symbolic time series analysis for anomaly detection: A comparative evaluation". In: *Signal Processing* 85.9 (2005), pp. 1859–1868. DOI: 10.1016/j.sigpro.2005.03.014.

[2] Jessica Lin et al. "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms". In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD '03)*. ACM, 2003, pp. 2–11. DOI: 10.1145/882082.882086.

6 Appendix

6.1 Data Generation

The iterative procedure for solving the Duffing equation is as follows:

$$\frac{d^2y}{dt^2} = -\beta\frac{dy}{dt} - y - y^3 + A\cos(\omega t),$$

which is numerically solved using a finite time step $\Delta t = \frac{1}{\text{sampling rate}}$. The updates at each iteration i are:

$$\frac{dy}{dt}[i] = \frac{dy}{dt}[i-1] + \frac{d^2y}{dt^2}[i-1] \cdot \Delta t,$$

$$y[i] = y[i-1] + \frac{dy}{dt}[i] \cdot \Delta t.$$

This iterative process is initialized with $y[0] = 0$ and $\frac{dy}{dt}[0] = 0$ and repeated for all β values in the specified range. The full implementation and code are included in the duffing_data.py file for reproducibility.
The figure below shows the generated results for three typical values of β ($\beta_{\text{nominal}} = 0.10$, $\beta_{\text{bifurcation}} = 0.29$, $\beta_{\max} = 0.35$).

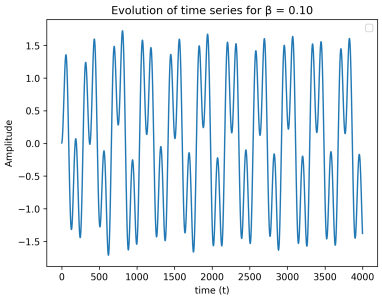

Figure 7: $\beta_{\text{nominal}} = 0.10$

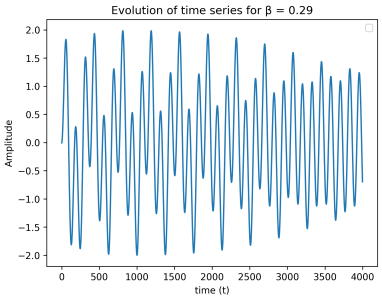

Figure 8: $\beta_{\text{bifurcation}} = 0.29$

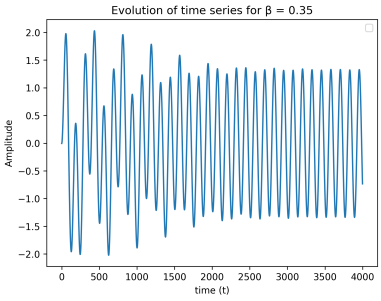

Figure 9: $\beta_{\max} = 0.35$

Figure 10: Time series shape based on some β value