Complexity and Classification Analysis Of Protocognitive Agents

Adnan Mahmud

August 13, 2024

1 Initial Data

The initial dataset comprises five distinct neuronal spike trains—technically proteinoid spikes, though modeled as neurons—each representing the firing patterns of individual neurons over time. These spike trains are characterized by a series of temporal points, where each point signifies the precise moment of a neuronal firing event; the data is structured as follows:

1. **Time series data:** Each spike train is represented as a time series—where the x-axis denotes time (in seconds)—and the y-axis is binary (0 or 1), indicating the presence or absence of a spike at each time point.

2. **Temporal resolution:** The timing of spikes is recorded with high precision—typically to the microsecond level—allowing for a detailed analysis of neuronal firing patterns.

3. **Variable duration:** Each spike train may have a different total duration, reflecting the variability in recording lengths or neuronal activity periods.

4. **Sparse nature:** Consistent with typical neuronal firing patterns, the spike trains exhibit a sparse structure—with relatively few spikes distributed over the recorded time period.

2 Transformation Process

To extract deeper insights and reveal potential hidden structures within the spike train data, we apply a novel transformation process. This transformation converts the initially sequential time series data into a complex, multinodal graph structure.

Figure 1 displays the original spike train data for all five datasets. Each row represents a separate dataset, with time (in seconds) on the x-axis and spike occurrences represented by vertical black lines. Dataset 1 exhibits a notably dense spike pattern, while Datasets 2-5 show varying degrees of sparsity and rhythmicity in their spike patterns. Temporal clustering of spikes is evident in some datasets, particularly in Datasets 3-5.

The transformation process involves two key functions, F1 and F2, which operate as follows:

2.1 Function F1: Spiral Sampling

F1 is defined as:

$$x(t) = 10 + (10 - 2t)\cos(t \cdot \pi), \quad \text{where } t \in \{1, 2, 3, ..., 19, 20\} \tag{1}$$

This function serves as a non-uniform sampling mechanism, designed to select points from the original spike train in a spiraling inward fashion. As t ranges from 1 to 20, $x(t)$ generates a series of values that oscillate between 0.5 and 20, with a general inward trend.

The key aspects of F1 are:

1. **Non-linear sampling:** Unlike uniform sampling, F1 provides a non-linear distribution of sampling points, potentially revealing patterns at different temporal scales.

2. **Bounded output:** The function's output is bounded between approximately 0.5 and 20, ensuring that the sampling remains within a predefined range regardless of the input spike train's duration.

3. **Decreasing periodicity:** The cosine term introduces an oscillatory behavior with decreasing amplitude, mimicking a spiral pattern when visualized.

2.2 Function F2: Significant Digit Extraction

F2 is defined as:

$$d = \lfloor (10^n) \cdot (x_i - \lfloor x_i \rfloor) \rfloor, \quad \text{such that } (10^n)x_i \geq 1 \text{ and } (10^{n-1})x_i < 1 \tag{2}$$

Where x_i represents a time point from the original spike train.

This function extracts the first significant digit after the decimal point for each selected time point. The key aspects of F2 are:

1. **Scale invariance:** By focusing on the first significant digit, F2 captures a scale-invariant property of the time points, potentially revealing patterns that are independent of the absolute time scale.

2. **Digit-based clustering:** Points with the same first significant digit are grouped together, creating a natural clustering mechanism based on this mathematical property.

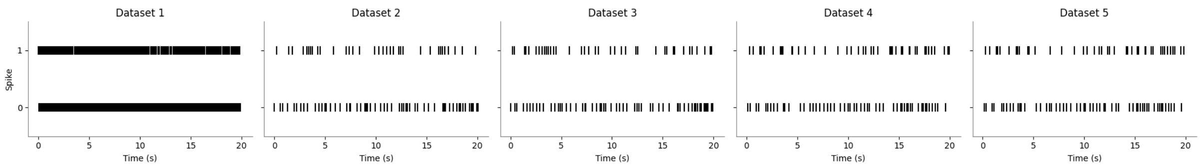

Figure 1: Raw Spike Trains Datasets

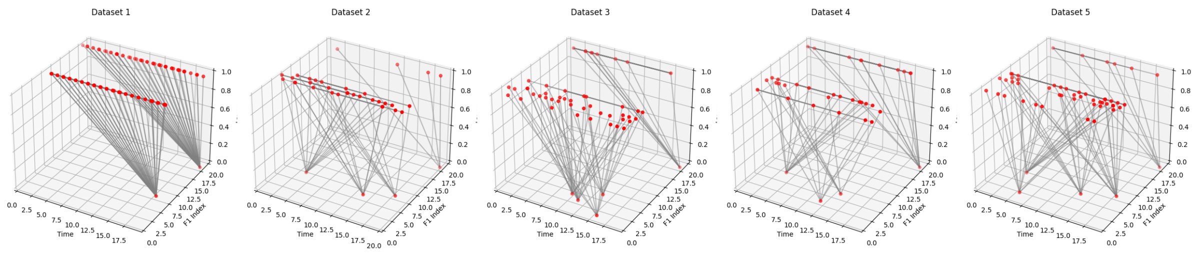

Figure 2: Transformed (Multinodal Graph Transformation) Spike Trains Datasets

3 Transformation Procedure & Effects

Figure 2 illustrates the results of applying our transformation process to each dataset, converting the linear spike trains into complex, three-dimensional graph structures. Each subplot represents a transformed dataset, where the x-axis represents time, the y-axis represents the F1 index, and the z-axis (vertical) represents the connection layer. Red nodes indicate the primary nodes selected by the F1 function, while grey edges connect the primary nodes to their corresponding secondary nodes. Table 1 outlines the mechanical steps of the transformation.

The transformation fundamentally alters the representation of the spike train data:

1. **Dimensionality Increase:** The original 1-dimensional time series is transformed into a 3-dimensional structure (time, F1 index, and connection layer).

2. **Topology Change:** The linear structure of the time series is converted into a complex graph with multiple interconnected nodes.

3. **Multi-scale Representation:** The combination of F1's spiral sampling and F2's digit extraction creates a multi-scale representation, potentially revealing both large-scale temporal patterns and fine-grained similarities between spike timings.

4. **Pattern Emergence:** The graph structure may reveal clusters or patterns of spike timings that were not readily apparent in the original linear representation.

5. **Information Condensation:** While some temporal information is lost in the transformation, the resulting structure condenses information about similarities and patterns in spike timing across different time scales.

This transformation offers a novel perspective on neuronal spike train data, potentially uncovering hidden structures and relationships that may provide new insights into neuronal firing patterns and their underlying mechanisms.

4 Complexity Metrics and Meta-Metric

4.1 Individual Complexity Metrics

To quantify the complexity of the transformed spike train data, we employ a set of graph-based metrics. These metrics capture various aspects of the multinodal graph structure, providing insights into the intricacy and patterns within the neuronal firing data. We then combine these metrics into a single meta-metric to obtain an overall measure of complexity.

4.2 Meta-Metric for Overall Complexity

To combine these individual metrics into a single measure of overall complexity, we define a meta-metric. This meta-metric is designed to provide a nuanced ranking of complexity that avoids artificial bounds and reflects the continuous nature of complexity. The meta-metric is calculated as follows:

1. **Normalization:** Each metric is normalized using z-scores to ensure comparability across

Table 1: Spike Train Analysis Algorithm

Input: Original spike train dataset S, Functions $F1$ and $F2$

Output: Graph G representing analyzed spike train

Algorithm:

1. Initialize empty sets:
   $P \leftarrow \emptyset$ (Set of primary nodes)
   $D \leftarrow \emptyset$ (Set of secondary nodes)
   $E \leftarrow \emptyset$ (Set of edges)
2. For each $t \in \{1, 2, ..., 20\}$:
   a) Compute $x(t) \leftarrow F1(t)$
   b) Find $p = \max\{s \in S : s \leq x(t)\}$ (closest point not exceeding $x(t)$)
   c) Add to primary nodes: $P \leftarrow P \cup \{p\}$
3. For each $p \in P$:
   Compute $d_p \leftarrow F2(p)$ (Extract first significant digit after decimal)
4. For each $s \in S$:
   a) Compute $d_s \leftarrow F2(s)$
   b) For each $p \in P$:
      If $d_s = d_p$:
         Add to secondary nodes: $D \leftarrow D \cup \{s\}$
         Add edge: $E \leftarrow E \cup \{(p, s)\}$
5. Construct graph: $G \leftarrow (P \cup D, E)$
6. Return G

different scales:

$$z_i = \frac{x_i - \mu_i}{\sigma_i} \tag{3}$$

where x_i is the original metric value, μ_i is the mean, and σ_i is the standard deviation of metric i across all datasets.

2. **Weighted Sum:** A weighted sum of the normalized metrics is computed:

$$S = \sum_{i=1}^{8} w_i z_i \tag{4}$$

where w_i are the weights assigned to each metric[1], reflecting their relative importance in determining overall complexity.

3. **Sigmoid Transformation:** The weighted sum is transformed using a sigmoid function to map the scores to a range between 0 and 1:

$$\text{Meta-Metric} = \frac{1}{1 + e^{-S}} \tag{5}$$

The sigmoid transformation ensures that the meta-metric asymptotically approaches 0 for extremely simple systems and 1 for extremely complex systems, without ever reaching these exact values. This reflects the idea that complexity exists on a continuous spectrum, and allows for meaningful comparisons between datasets while leaving room for potentially more or less complex datasets in future analyses.

4.3 Analysis of Complexity Metrics

To analyze the complexity of the transformed spike train data across our five datasets, we computed eight individual metrics and a meta-metric for each dataset. Table 3 and Figure 3 provides a comprehensive comparison of these metrics across datasets.

Complexity is not solely determined by graph size — Dataset 1 ranks fourth despite being the largest; local connectivity, particularly the clustering coefficient, is a strong contributor to overall complexity; balance between metrics is crucial — Dataset 2 achieves the highest complexity through balanced

[1] The weights w_i are as follows: Number of Nodes: 0.05, Number of Edges: 0.05, Average Degree: 0.10, Clustering Coefficient: 0.10, Graph Density: 0.20, Degree Entropy: 0.20, Number of Connected Components: 0.10, and Average Resistance: 0.20. These weights were chosen to balance the contribution of each metric, with slightly higher emphasis on degree entropy and average resistance as they capture more nuanced aspects of the graph structure.

Table 2: Individual Complexity Metrics

Metric	Description	Equation
Number of Nodes (N)	Counts the total number of nodes in the graph.	$N = \|V\|$, where V is the set of vertices in the graph
Number of Edges (E)	Counts the total number of connections in the graph.	$E = \|E\|$, where E is the set of edges in the graph
Average Degree ($\bar{k}$)	Measures the average number of connections per node.	$\bar{k} = \frac{2E}{N}$
Clustering Coefficient (C)	Quantifies the degree to which nodes in the graph tend to cluster together.	$C = \frac{1}{N}\sum_{i=1}^{N} C_i$, where C_i is the local clustering coefficient of node i
Graph Density (ρ)	Measures how close the graph is to being complete.	$\rho = \frac{2E}{N(N-1)}$
Degree Entropy (H)	Quantifies the complexity of the degree distribution.	$H = -\sum_k P(k)\log P(k)$, where $P(k)$ is the probability of a node having degree k
Number of Connected Components (CC)	Counts the number of disconnected subgraphs within the overall graph structure.	N/A
Average Resistance (R)	Measures the overall connectivity structure within components.	$R = \frac{1}{\|CC\|}\sum_{c \in CC} \frac{1}{\binom{\|V_c\|}{2}} \sum_{i<j \in V_c}(L^+_{ii} + L^+_{jj} - 2L^+_{ij})$, where L^+ is the pseudoinverse of the Laplacian matrix, and V_c is the set of vertices in component c

Table 3: Comprehensive Complexity Metrics Comparison

Rank	Dataset	Nodes	Edges	Avg Degree	Clustering Coef.	Density	Components	Degree Entropy	Avg Resistance	Meta Metric
1	Dataset 2	37	83	4.4865	0.7240	0.1246	4	1.8104	0.8273	0.5789
2	Dataset 4	37	78	4.2162	0.6855	0.1171	4	1.5879	0.9039	0.5353
3	Dataset 5	54	96	3.5556	0.3215	0.0671	5	1.4095	1.3010	0.4826
4	Dataset 1	100	446	8.9200	0.4278	0.0901	2	0.9759	1.0745	0.4568
5	Dataset 3	53	81	3.0566	0.3040	0.0588	5	1.3389	1.3095	0.4461

metrics; connected components play a nuanced role in complexity; degree entropy is a critical factor, suggesting the importance of diverse node connections. Dataset 2's high complexity could represent rich, structured neuronal interactions; Dataset 1 might represent high but uniform activity; a balance between local clustering and overall connectivity might indicate efficient information processing; multiple connected components in Datasets 3 and 5 could suggest distinct sub-networks.

The importance of individual metrics may vary depending on specific neuronal phenomena; further analysis is needed on the temporal evolution of complexity measures; there is potential for correlating complexity measures with specific neuronal functions or stimuli responses; a tailored analysis focusing on specific aspects of neuronal dynamics or comparative studies is necessary.

The insights gained from this analysis provide a foundation for understanding the structural complexity of transformed spike train data. Further and tailored analysis will depend on the overall narrative of the paper, potentially focusing on specific aspects of neuronal dynamics or comparative studies across different experimental conditions.

5 Classification Model for Spike Train Analysis

We developed a binary classification model aimed at predicting neuronal spikes, in order to extract insights into the temporal dynamics of neuronal firing. .

5.1 Model Architecture

We implemented a feedforward neural network using dense layers for our classification task. The architecture of our model is as follows:

- Input Layer: Corresponds to the number of engineered features (detailed in Section 5.2).
- Hidden Layers: Four dense layers with 128, 64, 32, and 16 neurons respectively.
- Output Layer: A single neuron with sigmoid activation for binary classification.

All hidden layers utilize the Rectified Linear Unit (ReLU) activation function, which introduces nonlinearity and helps mitigate the vanishing gradient problem during training.

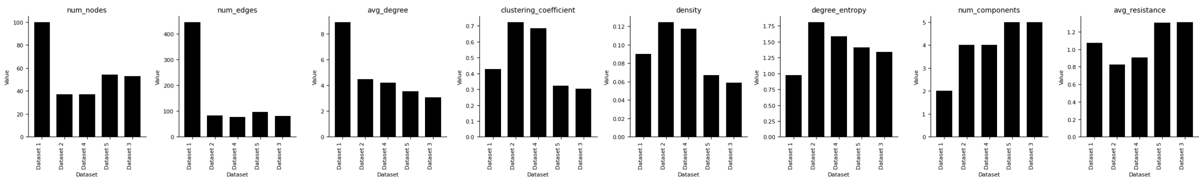

Figure 3: Comparison of Complexity Metrics Across Datasets

5.2 Feature Engineering

Table 4 presents a comprehensive and technical overview of all features used in our spike train classification model. These features are designed to capture complex temporal dynamics and structural characteristics of the neuronal activity.

The complete feature vector $\mathbf{x}_i$ for each time point i is constructed as follows:

$$\mathbf{x}_i = [t_i, \Delta t_i, ISI_i, CV_{ISI,i}, \mu_{S,i,3}, \sigma_{S,i,3}, \mu_{S,i,5}, \sigma_{S,i,5}, \mu_{S,i,10}, \sigma_{S,i,10}, F_{sin,5,i}, F_{cos,5,i}, F_{sin,10,i}, F_{cos,10,i}, F_{sin,20,i}, F_{cos,20,i}]$$

This 16-dimensional feature vector provides a rich representation of the spike train data—encompassing temporal, statistical, and spectral characteristics. By leveraging this comprehensive feature set, our classification model can capture intricate patterns in neuronal firing behavior—enabling accurate spike prediction in our protocognitive agent data.

5.3 Model Performance Analysis

5.3.1 Classification Metrics

As tabulated in Table 5, the model achieves an accuracy of 70.41%, i.e., correctly classifying about 7 out of 10 instances. While this indicates better-than-chance performance, there is room for improvement. With a precision of 70.59%, the model demonstrates a good ability to avoid false positives. When the model predicts a spike, it is correct approximately 71% of the time. The recall of 55.81% suggests that the model identifies about 56% of all actual spikes. This relatively lower recall indicates that the model misses a significant portion of true spikes. The F1 score of 0.6234 provides a balanced measure of the model's performance, considering both precision and recall. This score indicates moderate performance but also highlights the potential for enhancement.

5.3.2 Receiver Operating Characteristic (ROC) Curve

Figure 4 illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) at various classification thresholds. Our model achieves an Area Under the Curve (AUC) of 0.73, indicating a moderate discriminative ability. This AUC value suggests that the model performs better than random guessing (AUC = 0.5) in distinguishing between spike and non-spike events.

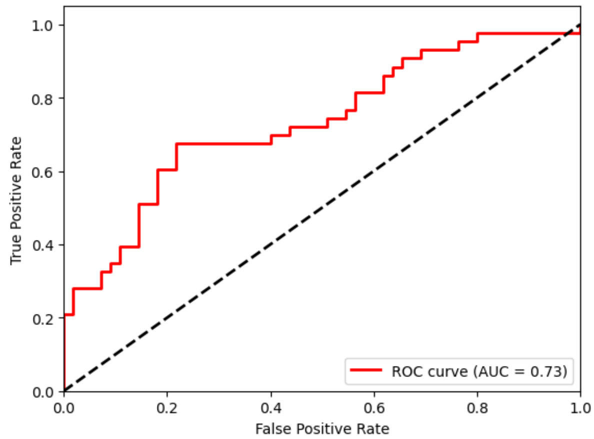

Figure 4: Receiver Operating Characteristic (ROC) Curve

The curve's shape reveals that our model achieves a relatively high true positive rate at lower false positive rates, as evidenced by the steep initial rise. This characteristic is desirable, especially in the context of neuronal spike detection where minimizing false positives while maintaining sensitivity is crucial.

5.3.3 Interpretation and Implications

The model's performance metrics reveal several key insights:

1. **Balanced Performance:** The similar values of precision and accuracy suggest a relatively balanced performance, which is crucial in spike train analysis where both false positives and false negatives can significantly impact interpretations.

2. **Conservative Predictions:** The higher precision compared to recall indicates that the model is somewhat conservative in its spike predictions. It prioritizes avoiding false positives over capturing all spikes.

3. **Missed Spikes:** The lower recall value suggests that the model is missing a considerable num-

Meta Feature	Feature	Mathematical Definition	Description
Temporal	Time (t_i)	t_i	Original timestamp of each data point, preserving absolute temporal information.
	Time Difference (Δt_i)	$\Delta t_i = t_i - t_{i-1}$	Interval between consecutive data points, capturing local temporal dynamics and identifying irregularities in sampling or firing patterns.
	Inter-Spike Interval (ISI_i)	$ISI_i = \begin{cases} t_i - t_j & \text{if } S_i = 1 \text{ and } S_j = 1 \\ 0 & \text{otherwise} \end{cases}$	Time difference between consecutive spikes, where S_i is the spike indicator (0 or 1) at time i, and j is the index of the previous spike. Provides insights into rhythmicity and variability of neuronal firing.
Statistical	Coefficient of Variation of ISIs ($CV_{ISI,i}$)	$CV_{ISI,i} = \frac{\sigma_{ISI,i,w}}{\mu_{ISI,i,w}}$	Measure of variability in spike timing, calculated over a rolling window of size w. $\sigma_{ISI,i,w}$ and $\mu_{ISI,i,w}$ are the standard deviation and mean of ISIs in the window ending at time i. Quantifies regularity or irregularity of spike patterns.
	Rolling Mean ($\mu_{S,i,w}$)	$\mu_{S,i,w} = \frac{1}{w}\sum_{j=i-w+1}^{i} S_j$	Average of spikes over a rolling window of size w (w = 3, 5, 10), capturing local spike density at different temporal scales.
	Rolling Standard Deviation ($\sigma_{S,i,w}$)	$\sigma_{S,i,w} = \sqrt{\frac{1}{w-1}\sum_{j=i-w+1}^{i}(S_j - \mu_{S,i,w})^2}$	Standard deviation of spikes over a rolling window of size w (w = 3, 5, 10), quantifying local variability in spike patterns.
Spectral	Sine Transformation ($F_{sin,p,i}$)	$F_{sin,p,i} = \sin\left(\frac{2\pi t_i}{p}\right)$	Sine component of the Fourier transformation for periods p (p = 5, 10, 20). Captures oscillatory behavior and periodic patterns in the spike train data.
	Cosine Transformation ($F_{cos,p,i}$)	$F_{cos,p,i} = \cos\left(\frac{2\pi t_i}{p}\right)$	Cosine component of the Fourier transformation for periods p (p = 5, 10, 20). Complements the sine component in identifying and quantifying periodic components across different time scales.

Table 4: Comprehensive Feature Set for Spike Train Classification

Metric	Value
True Positives	24
False Positives	10
True Negatives	45
False Negatives	19
Accuracy	0.7041
Precision	0.7059
Recall	0.5581
F1 Score	0.6234

Table 5: Classification Performance Metrics

ber of actual spikes. This could lead to underestimation of neuronal activity in certain analyses.

4. **Potential for Improvement:** The moderate F1 score and AUC indicate that while the model performs better than random guessing, there is substantial room for improvement. This could potentially be achieved through feature engineering, model architecture refinement, or increased training data.

5. **Context-Dependent Utility:** The model's current performance may be more suitable for applications where minimizing false positives is prioritized over capturing every spike. However, for studies requiring high sensitivity to neuronal activity, further optimization would be beneficial.

In the context of analyzing protocognitive agents, these results suggest that our model can, potentially, provide valuable insights into spike patterns, but—of course—caution should be exercised when making definitive claims about neuronal activity based solely on these predictions.

A Appendix

This appendix provides additional visualizations of our spike train classification model's performance, offering deeper insights into its behavior during training and its predictive accuracy.

Figure 5 presents the confusion matrix of our model's predictions on the test set.

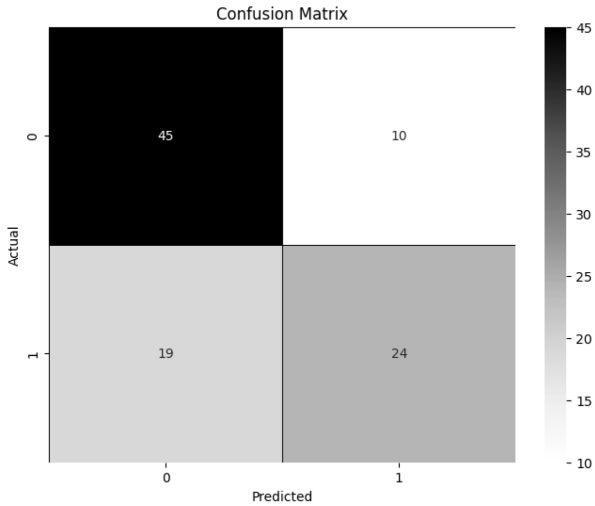

Figure 5: Confusion Matrix of Spike Prediction Model

Figure 6 shows the evolution of accuracy and loss for both training and validation sets over the course of model training.

These plots provide several key insights:

- **Accuracy (left plot):** The training accuracy (red line) shows a rapid increase in the early epochs, followed by a more gradual improvement. The validation accuracy (black line) remains relatively stable, suggesting that the model generalizes well to unseen data.

- **Loss (right plot):** The training loss (red line) decreases steadily throughout the training process, indicating continuous improvement in the model's performance on the training data. The validation loss (black line) shows some fluctuations but generally remains stable, further supporting the model's generalization capability.

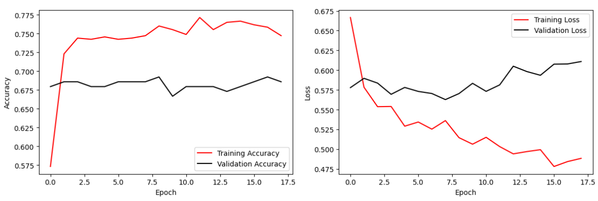

Figure 6: Training and Validation Accuracy and Loss

- **Overfitting assessment:** The gap between training and validation metrics is relatively small, suggesting that the model is not severely overfitting to the training data.

These visualizations complement the quantitative metrics presented in the main text, providing a more comprehensive view of the model's learning process and predictive performance.