Evaluating Extended Automata for Sequential Regression

Job Vonck Supervisor: S.E. Verwer

Delft University of Technology - CSE3000 Research Project

Introduction

Sequential regression predicts the next value in a sequence. It is used in many real-world applications such as energy forecasting and traffic prediction. These problems often contain repeating patterns. Automata can model such patterns using **states, transitions, and loops**.

Extended Regression Automata (ERA) allow us to use automata for regression:

- **Guards:** conditions that decide which transition is taken
- **Predictions:** each state q predicts a value $P(q)$

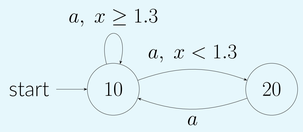

Figure 1. Example ERA with guards and predictions.

RQ1: Can extended automata outperform standard regression baselines?
RQ2: Do automata learned with BIC outperform those learned with AIC?

Methodology

- Sequential data is converted into **sliding-window traces**
- These traces are given to **FlexFringe** [1], which learns an automaton using:
  - **Splits:** create more specific states using guards
  - **Merges:** combine similar states and can create loops
- Each split or merge is evaluated using an **information criterion**:
  - **AIC:** $\Delta\text{AIC} = 2k + n \ln\left(\frac{\text{RSS}_{\text{before}}}{\text{RSS}_{\text{after}}}\right)$
  - **BIC:** $\Delta\text{BIC} = k \ln(n) + n \ln\left(\frac{\text{RSS}_{\text{before}}}{\text{RSS}_{\text{after}}}\right)$

Parameters: k = model complexity, n = number of data points, RSS = prediction error.

Baselines: Persistence, Regression Tree, Random Forest Regressor

Datasets:

- **Noisy sine**: a simple periodic signal, with some noise
- **Mackey-Glass**: a periodic function from mathematical biology
- **Hourly electricity consumption**: dataset containing energy consumption

Results: Noisy Sine

All methods except persistence achieve an MAE of ≈ 0.05, matching the expected error due to the added noise. **BIC achieves the lowest error** across all metrics and produces a **6-state looping automaton** (versus 8 states for AIC).

Metric	Persistence	Reg. Tree	Rand. Forest	BIC	AIC
MAE	1.0103	0.0495	0.0489	**0.0460**	0.0475
RMSE	1.0131	0.0572	0.0554	**0.0535**	0.0551
MAPE (%)	99.80	4.89	4.82	**4.52**	4.80

Table 1. Noisy sine results (lower is better).

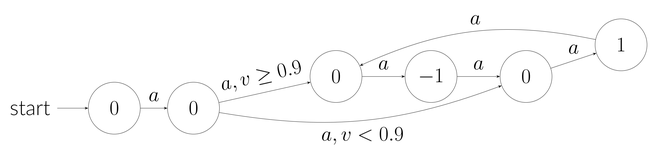

Figure 2. Automaton produced by BIC for noisy sine.

Results: Mackey-Glass & Electricity

On the larger datasets, **Random Forest consistently outperforms** the automata. The learned automata become tree-shaped, with few merges. This led to overfitting since the models followed individual training traces instead of learning reusable loop-based patterns.

Dataset	Win.	Persist.	RT	RF	BIC	AIC
Mackey-Glass	4	0.0647	0.0054	**0.0050**	0.0100	0.0082
Mackey-Glass	8	0.0616	0.0060	**0.0029**	0.0056	0.0055
Mackey-Glass	48	0.065	0.0021	**0.0012**	0.0175	0.0223
Electricity	24	7.03	6.70	**4.51**	7.53	7.82

Table 2. MAE results for larger datasets (lower is better).

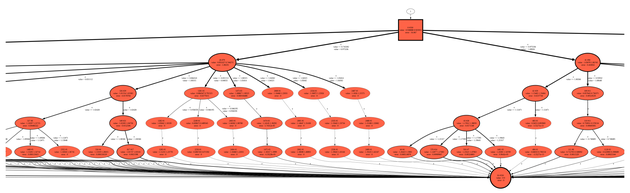

Figure 3. Subsection of automaton produced by BIC for the Mackey-Glass dataset with a window size of 4.

Why Do Loops Fail to Form?

The main issue is a **bias towards splitting** in the learning process:

- **Splitting** reduces RSS by creating more specialised states
- **Merging** can increase RSS because it combines states with different predictions
- On complex datasets, the RSS gain from splitting often dominates the AIC/BIC complexity penalty
- Result: a **tree-shaped automaton** with few merges, no useful loops, and a higher risk of overfitting

Conclusions

RQ1: Can automata outperform baselines?

- **Simple data (noisy sine):** BIC achieves the best score similar to Random Forest
- **Complex data:** Random Forest wins because the automata formed tree-shaped models instead of finding a pattern.

RQ2: BIC vs. AIC?

- Accuracy mixed across all datasets. Both are able to beat each other in specific cases.
- BIC produces more compact models (199 vs. 301 states, Mackey-Glass window 4)
- BIC up to **4× faster** (6s vs. 21s, Mackey-Glass window 8)

Future work:

- Investigate other learning methods like **RTI**

References

[1] Sicco Verwer and Christian Hammerschmidt. "FlexFringe: Modeling Software Behavior by Learning Probabilistic Automata". In: *Logical Methods in Computer Science* Volume 21, Issue 3 (24, 2025), p. 9295. DOI: `10.46298/lmcs-21(3:31)2025`.