TADAM: Learning Timed Automata from Noisy Observations

Lénaïg Cornanguer, Pierre-François Gimenez
(equal contribution)

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY
Inria

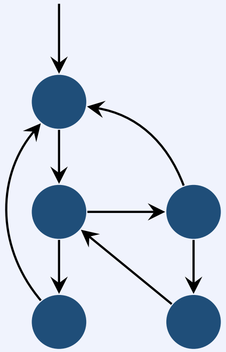
Introduction

communication protocol

system

industrial process

Introduction



communication protocol

system → behavior model

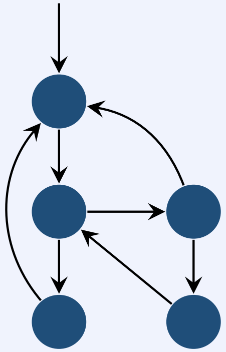
industrial process

Introduction

communication protocol

industrial process

system



behavior model

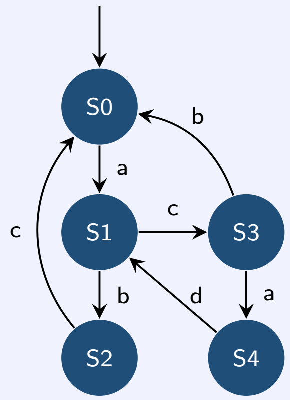
monitoring

model checking

data generation

...

Behavior model formalism



Automata formalism

Finite state automata (FSA)

Natural formalism for **discrete event system (DES)** modeling

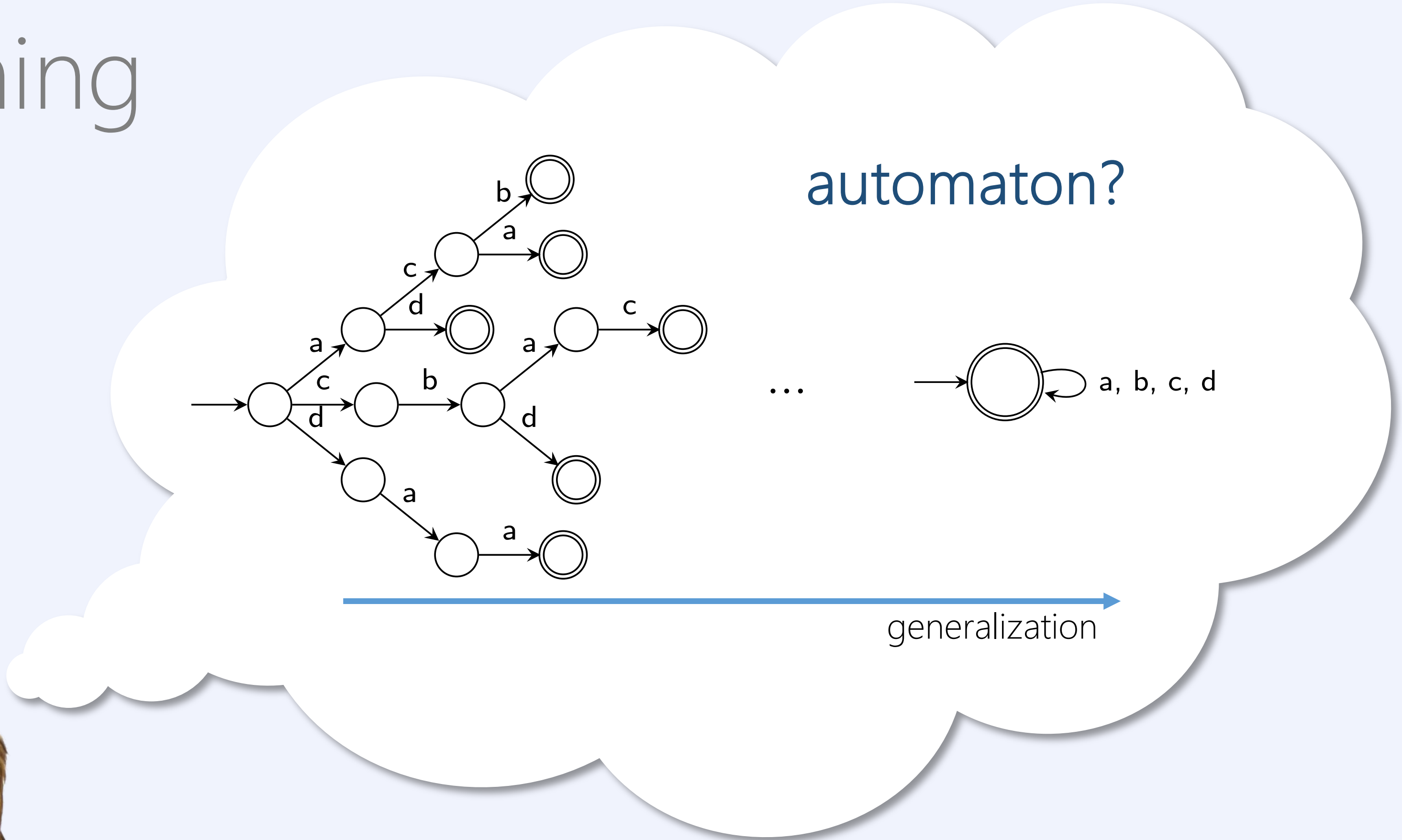
Human-understandable representation of the behavior of a system

Based on a **mathematical formalism** with **extensive literature** and with **software support**

Automata learning

data

a c b
a d
c b a c
a c a
d a a
c b d

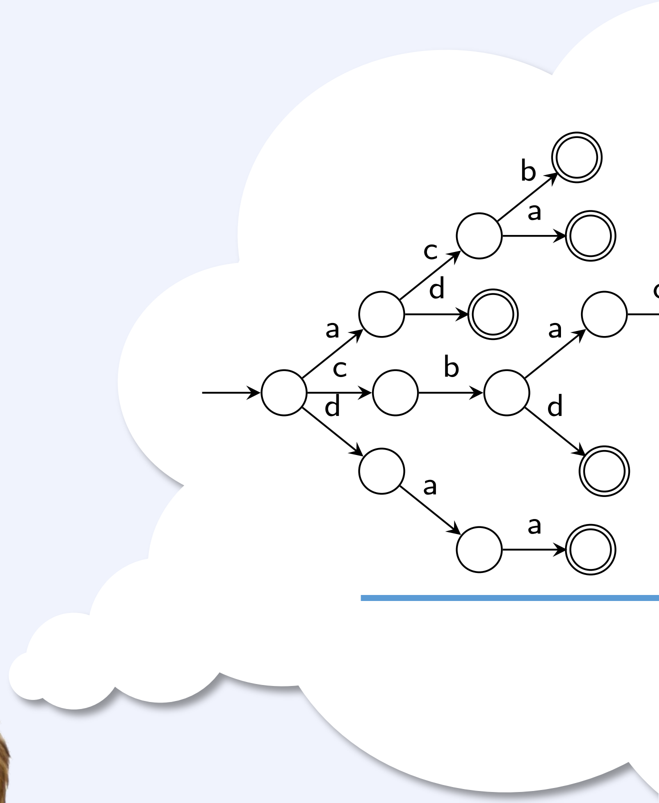


Automata learning

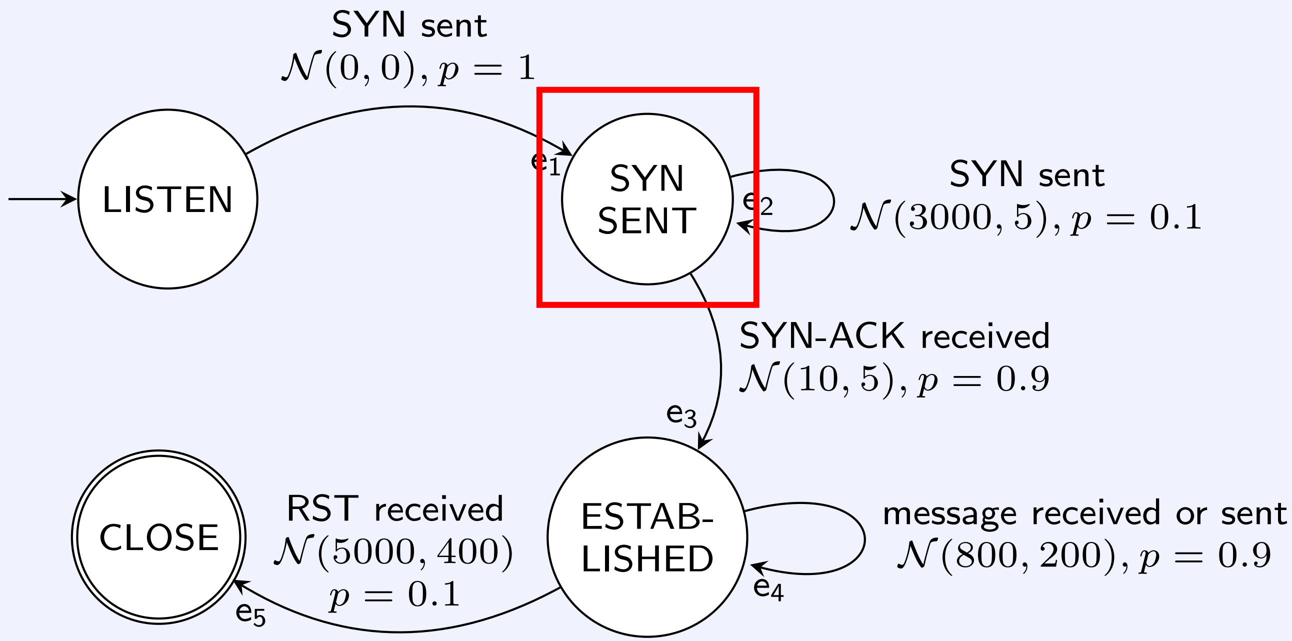
limited mesurement accuracy,
probe configuration error
…

noisy
data

a c b
a d
c b a c
a c a
d a a
c b d

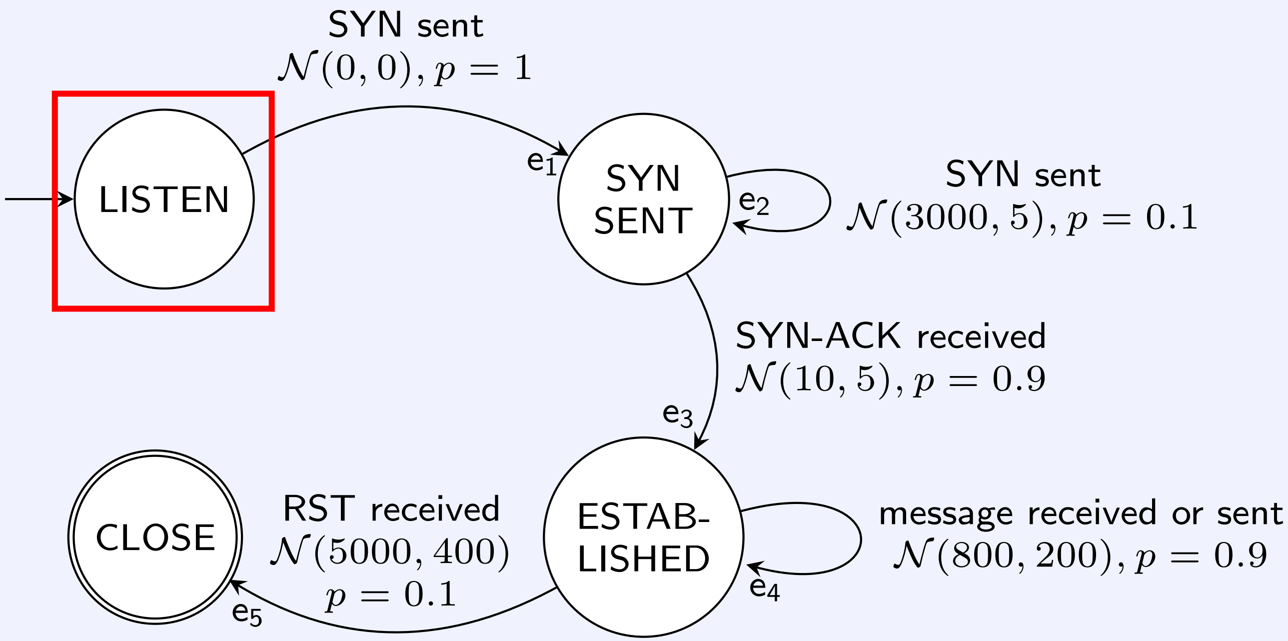


Probabilistic Real-Time Automaton



location
$q \in \mathcal{Q}$

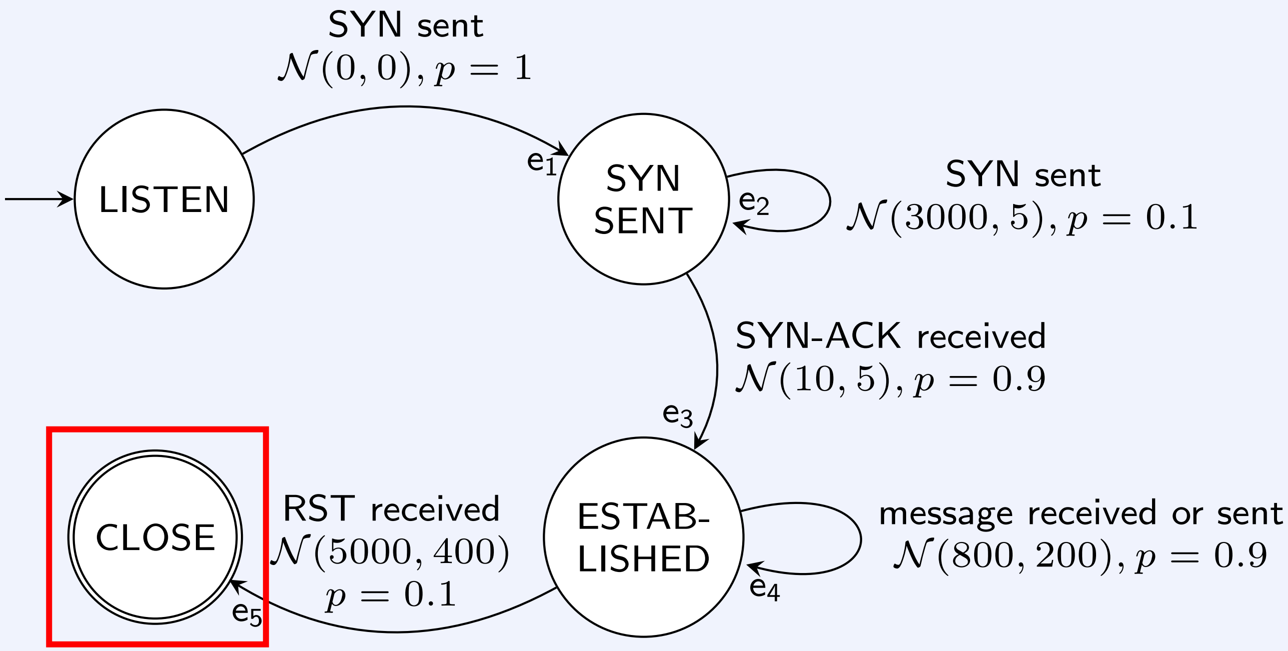
Probabilistic Real-Time Automaton



initial

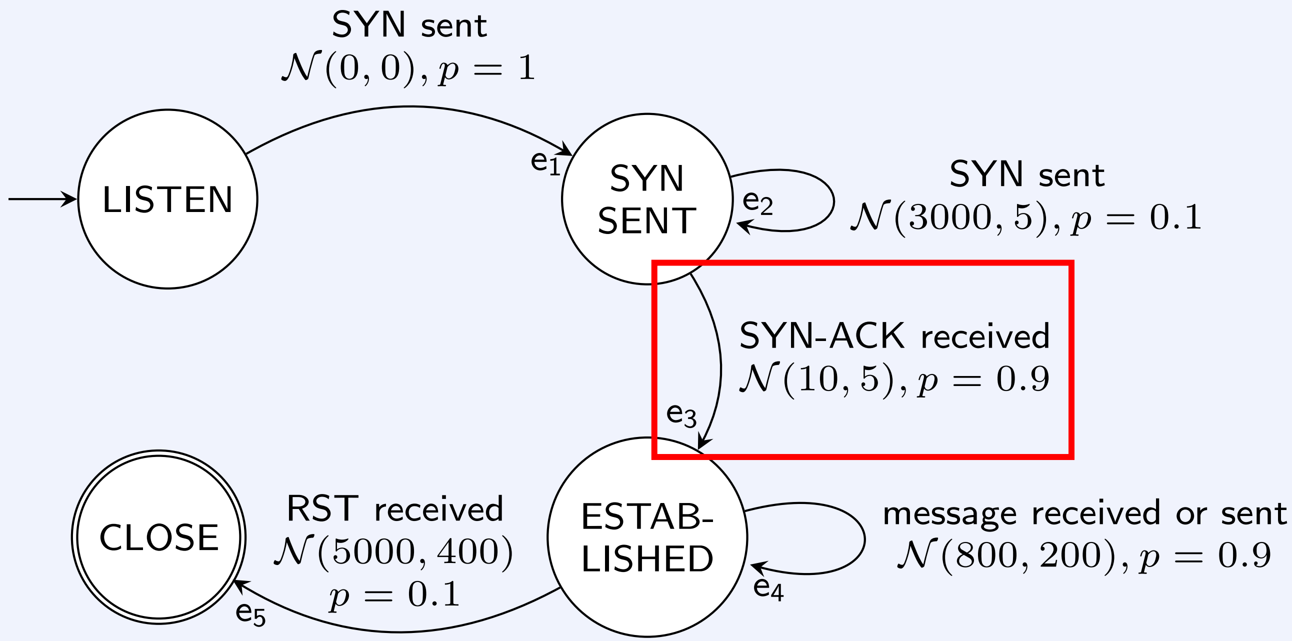
location

Probabilistic Real-Time Automaton



final location

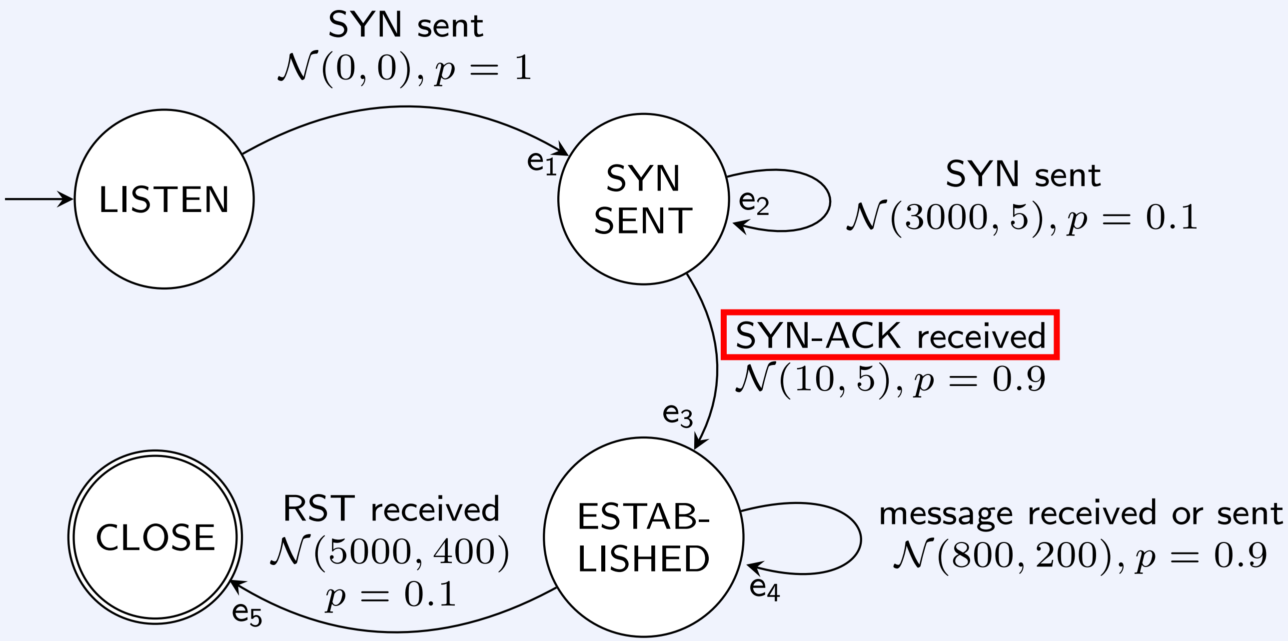
Probabilistic Real-Time Automaton



transition

$e \in \mathcal{E}$

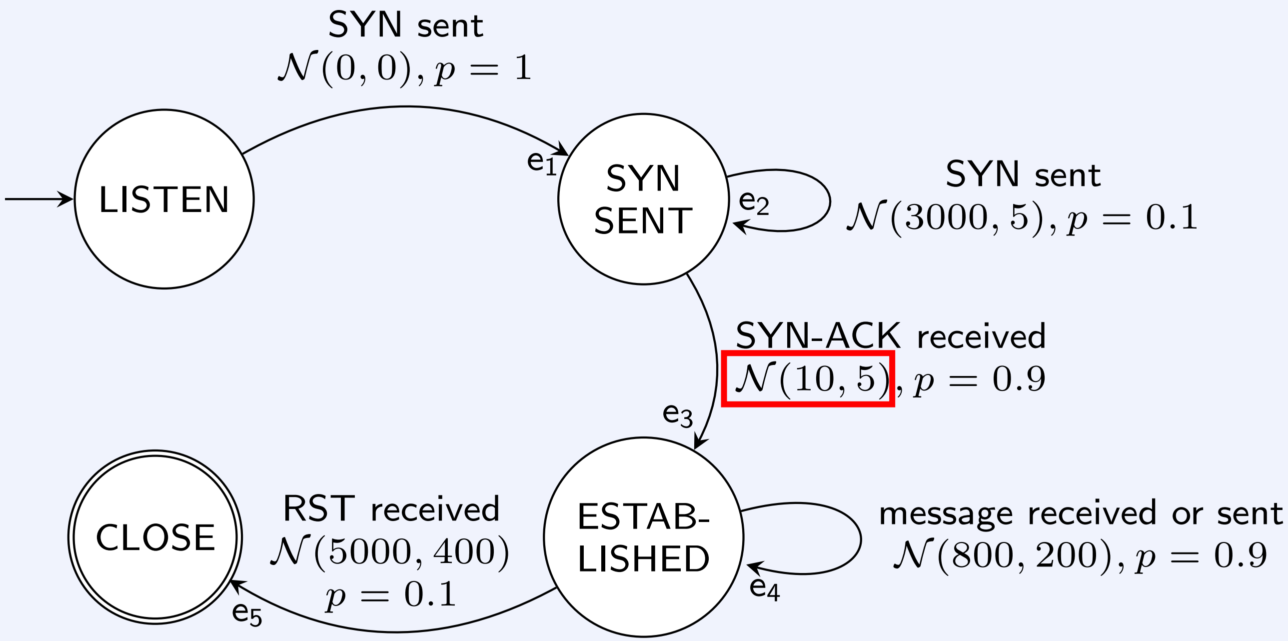
Probabilistic Real-Time Automaton



symbol

$a \in \Sigma$

Probabilistic Real-Time Automaton

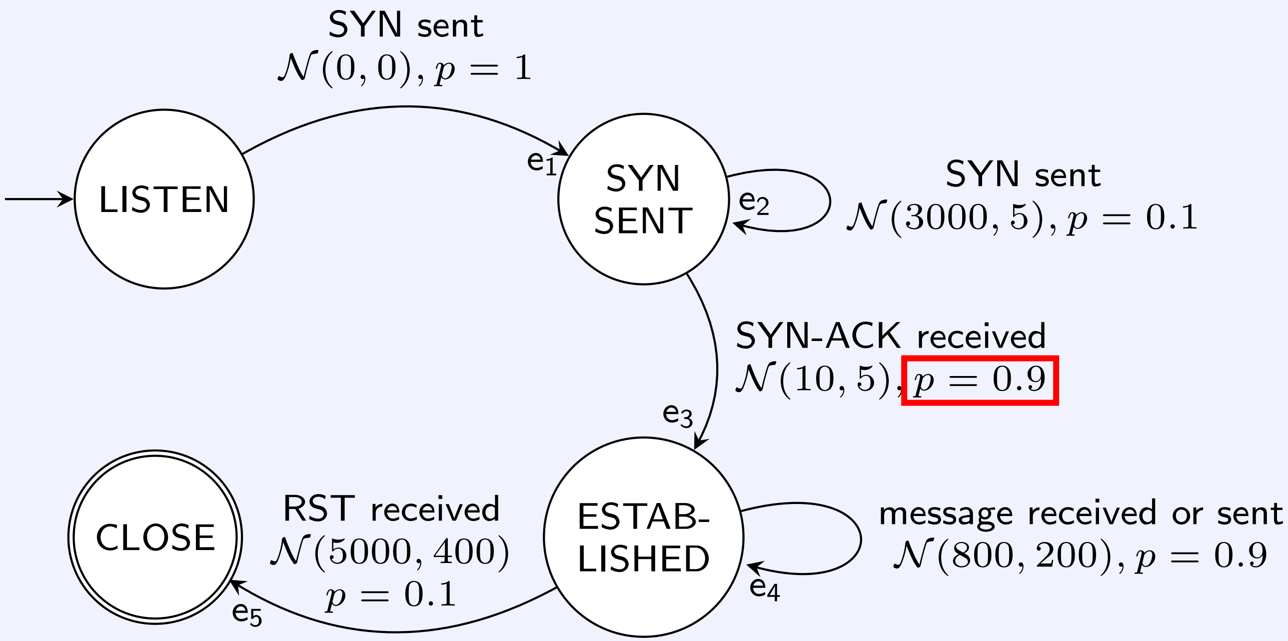


delay

distribution

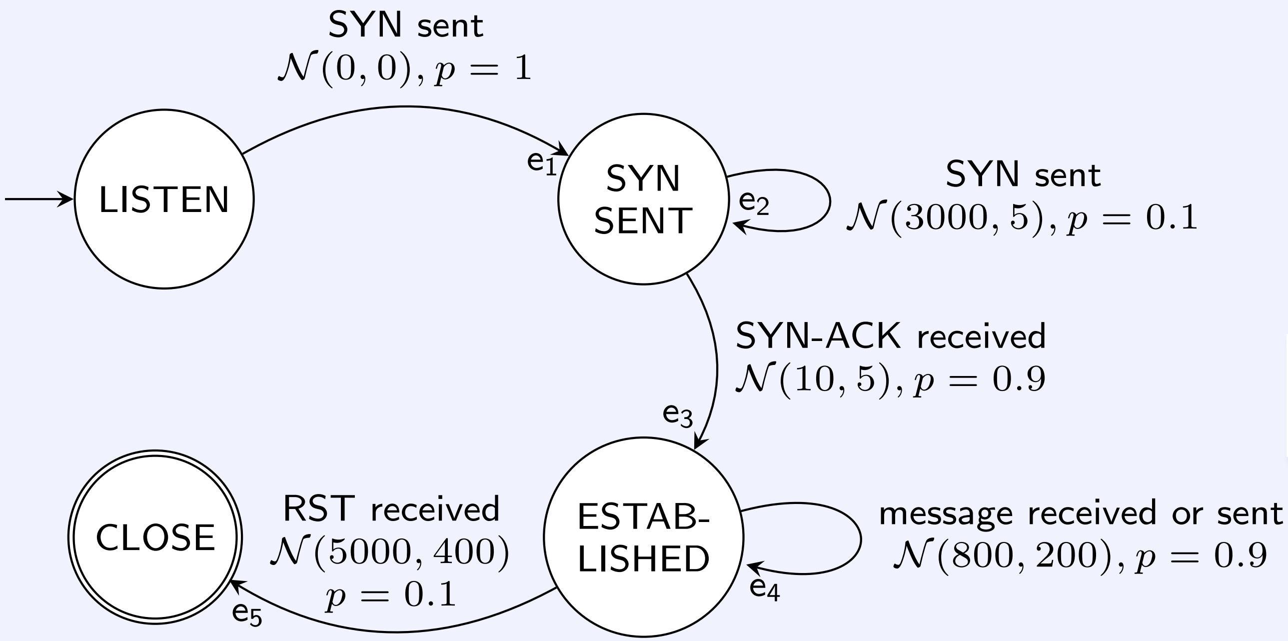
$\mathcal{N}(\mu, \sigma)$

Probabilistic Real-Time Automaton



transition probability p

Probabilistic Real-Time Automaton



timed word

(SYN sent, 0),(SYN sent, 2000),(SYN-ACK received, 10),(RST received, 6500)

$$w = \{(a,d), ...\}$$

Noise model

(SYN sent, 0),(SYN sent, 2000),(SYN-ACK received, 10),(RST received, 6500)

deletion

(SYN sent, 0),(SYN sent, 2000),(RST received, 6500)

Noise model

(SYN sent, 0),(SYN sent, 2000),(SYN-ACK received, 10),(RST received, 6500)

$\downarrow$ insertion

(SYN sent, 0),(SYN sent, 2000),(SYN-ACK received, 10),(SYN sent, 50),(RST received, 6500)

Noise model

(SYN sent, 0),(SYN sent, 2000),(SYN-ACK received, 10),(RST received, 6500)

transposition

(SYN sent, 0),(SYN-ACK received, 10),(SYN sent, 2000),(RST received, 6500)

Noise model

(SYN sent, 0),(SYN sent, 2000),(SYN-ACK received, 10),(RST received, 6500)

symbol repetition

(SYN sent, 0),(SYN sent, 2000),(SYN-ACK received, 10),(SYN-ACK received, 7),(RST received, 6500)

Question

How to learn timed automata from noisy timed words?

Occam's razor

Why won't my car start?

A group of hackers remotely accessed your car's onboard computer overnight, disabling the ignition system as part of a sophisticated cyber-attack targeting random individuals to create chaos.

Your car battery is dead.

Occam’s razor

The simplest model that fits the data is usually the correct one

MDL principle

Two-part description length

$$L(\mathcal{D}, \mathcal{A}) = L(\mathcal{D}|\mathcal{A}) + L(\mathcal{A})$$

MDL principle

Two-part description length

$$L(\mathcal{D}, \mathcal{A}) = L(\mathcal{D}|\mathcal{A}) + L(\mathcal{A})$$

global length

(# of bits needed)

MDL principle

Two-part description length

$$L(\mathcal{D}, \mathcal{A}) = L(\mathcal{D}|\mathcal{A}) + L(\mathcal{A})$$

global length
(# of bits needed)

length of describing
the automaton

MDL principle

Two-part description length

$$L(\mathcal{D}, \mathcal{A}) = L(\mathcal{D}|\mathcal{A}) + L(\mathcal{A})$$

global length
(# of bits needed)

length of describing the
data encoded with
the automaton

length of describing
the automaton

MDL principle

Two-part description length

$$L(\mathcal{D}, \mathcal{A}) = L(\mathcal{D}|\mathcal{A}) + L(\mathcal{A})$$

global length
(# of bits needed)

length of describing the
data encoded with
the automaton

length of describing
the automaton

Minimum Description Length principle

$$\mathcal{A}^* = \underset{\mathcal{A} \in \boldsymbol{\mathcal{A}}}{\operatorname{argmin}}\ L(\mathcal{D}|\mathcal{A}) + L(\mathcal{A})$$

Automaton encoding

$$L(\mathcal{A}) =$$

Automaton encoding

- Locations

$$L(\mathcal{A}) = L_{\mathbb{N}}(|\mathcal{Q}|)$$

Automaton encoding

- Locations
- Alphabet

$$L(\mathcal{A}) = L_{\mathbb{N}}(|\mathcal{Q}|) + L_{\mathbb{N}}(|\Sigma|)$$

Automaton encoding

- Locations
- Alphabet
- Initial and accepting locations

$$L(\mathcal{A}) = L_{\mathbb{N}}(|\mathcal{Q}|) + L_{\mathbb{N}}(|\Sigma|) + 2\log_2(|\mathcal{Q}|) +$$

Automaton encoding

- Locations
- Alphabet
- Initial and accepting locations
- For each transition:

$$L(\mathcal{A}) = L_{\mathbb{N}}(|\mathcal{Q}|) + L_{\mathbb{N}}(|\Sigma|) + 2\log_2(|\mathcal{Q}|) +$$

$$\sum_{e \in \mathcal{E}} \Big(\qquad \Big)$$

Automaton encoding

- Locations
- Alphabet
- Initial and accepting locations
- For each transition:
  - Source and destination locations

$$L(\mathcal{A}) = L_{\mathbb{N}}(|\mathcal{Q}|) + L_{\mathbb{N}}(|\Sigma|) + 2\log_2(|\mathcal{Q}|) +$$

$$\sum_{e \in \mathcal{E}} \Big(2\log_2(|\mathcal{Q}|) \qquad \Big)$$

Automaton encoding

- Locations
- Alphabet
- Initial and accepting locations
- For each transition:
  - Source and destination locations
  - Symbol

$$L(\mathcal{A}) = L_{\mathbb{N}}(|\mathcal{Q}|) + L_{\mathbb{N}}(|\Sigma|) + 2\log_2(|\mathcal{Q}|) +$$

$$\sum_{e \in \mathcal{E}} \Big(2\log_2(|\mathcal{Q}|) + \log_2(|\Sigma|) + \Big)$$

Automaton encoding

- Locations
- Alphabet
- Initial and accepting locations
- For each transition:
  - Source and destination locations
  - Symbol
  - Guards’ normal distributions parameters

$$
\begin{aligned}
L(\mathcal{A}) =& L_{\mathbb{N}}(|\mathcal{Q}|) + L_{\mathbb{N}}(|\Sigma|) + 2\log_2(|\mathcal{Q}|) + \\
& \sum_{e \in \mathcal{E}} \Big(2\log_2(|\mathcal{Q}|) + \log_2(|\Sigma|) + \\
& \qquad L_{\mathbb{N}}(\lfloor \mu_e \rfloor) + L_{\mathbb{N}}(\lfloor \sigma_e^2 \rfloor) \Big)
\end{aligned}
$$

Data encoding

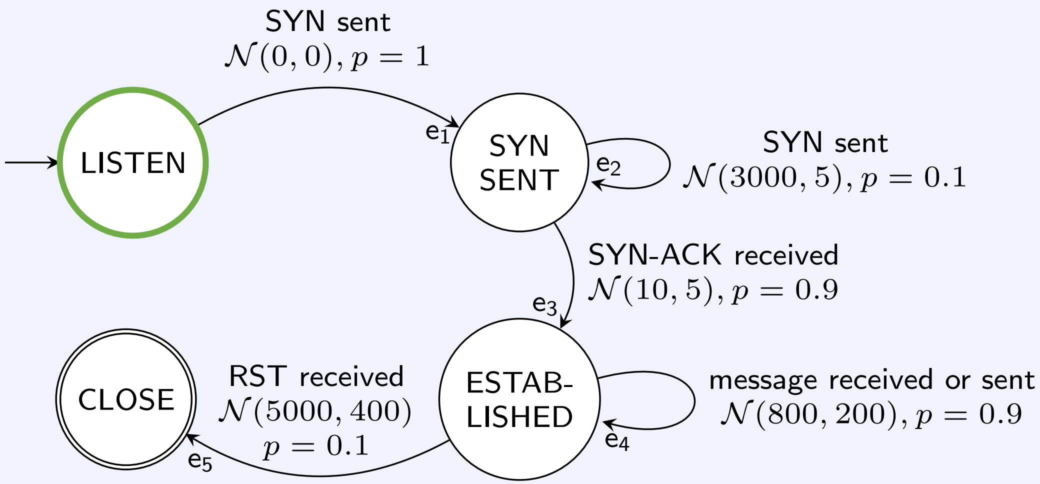
1. Correct the non-accepted words to remove the noise
2. Encode the corrected data

Data correction

Noise type	Correction operation
deletion aa**b**cad → aacad	add
insertion aabcad → aabc**b**ad	skip
transposition aabcad → aa**cb**ad	transpose
symbol repetition aabcad → aabca**a**d	deduplicate
–	follow

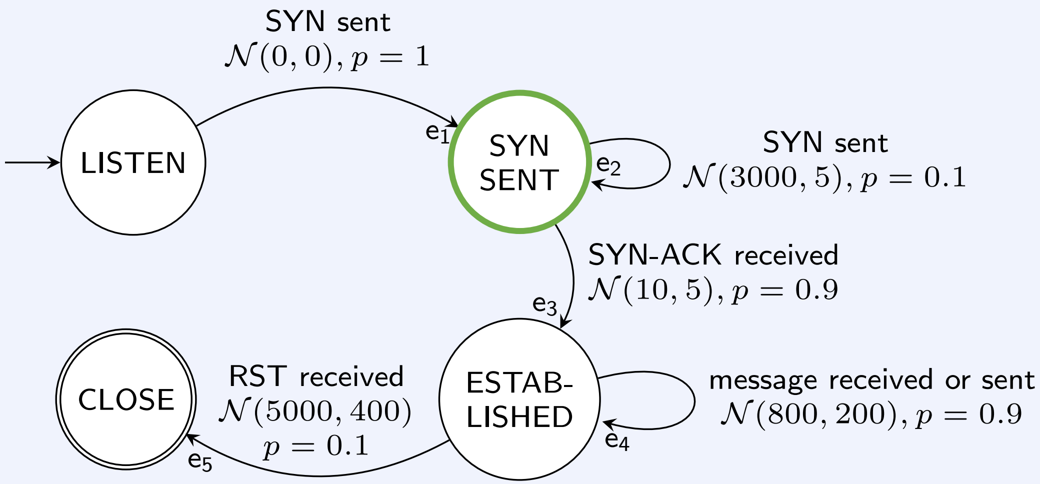
Data correction

word	edit operation sequence {operation, transition to use, delay}	corrected word
(SYN sent, 0), (ACK sent, 500), (SYN sent, 2000), (RST received, 6500)		



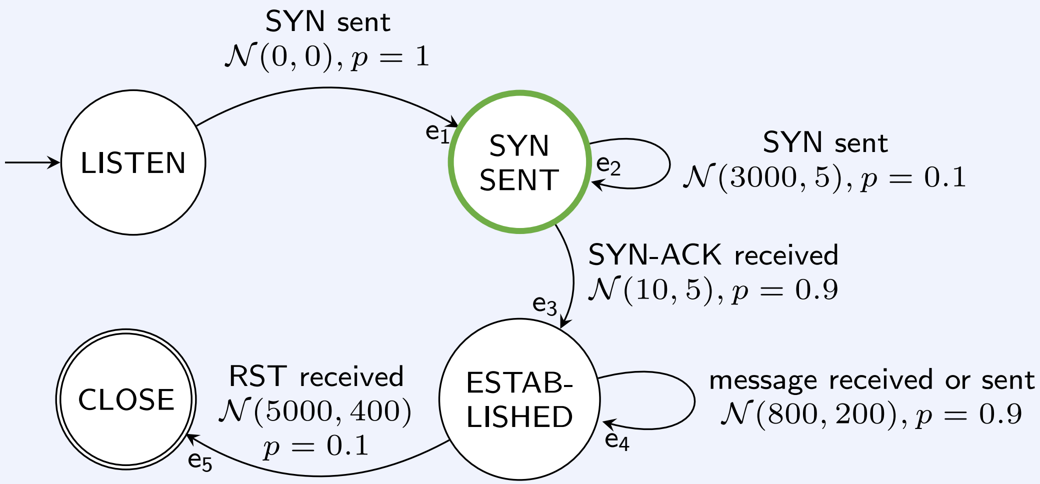
Data correction

word	edit operation sequence {operation, transition to use, delay}	corrected word
(SYN sent, 0), (ACK sent, 500), (SYN sent, 2000), (RST received, 6500)	(follow, e1 , 0),	(SYN sent, 0),



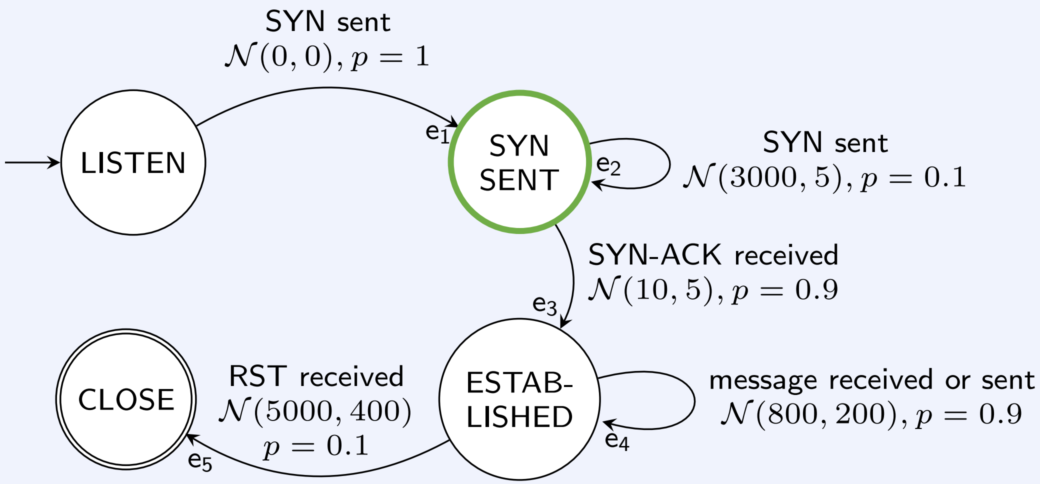
Data correction

word	edit operation sequence {operation, transition to use, delay}	corrected word
(SYN sent, 0), (ACK sent, 500), (SYN sent, 2000), (RST received, 6500)	(follow, e1 , 0), (skip, ε, 500),	(SYN sent, 0),



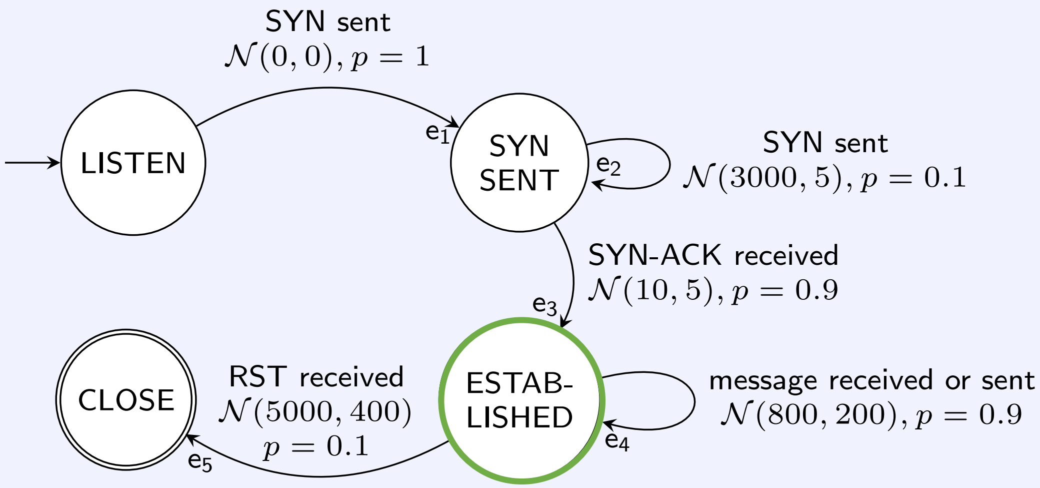
Data correction

word	edit operation sequence {operation, transition to use, delay}	corrected word
(SYN sent, 0),	(follow, e1 , 0),	(SYN sent, 0),
(ACK sent, 500),	(skip, ε, 500),	
(SYN sent, 2000),	(follow, e2 , 2000),	(SYN sent, 2000),
(RST received, 6500)		



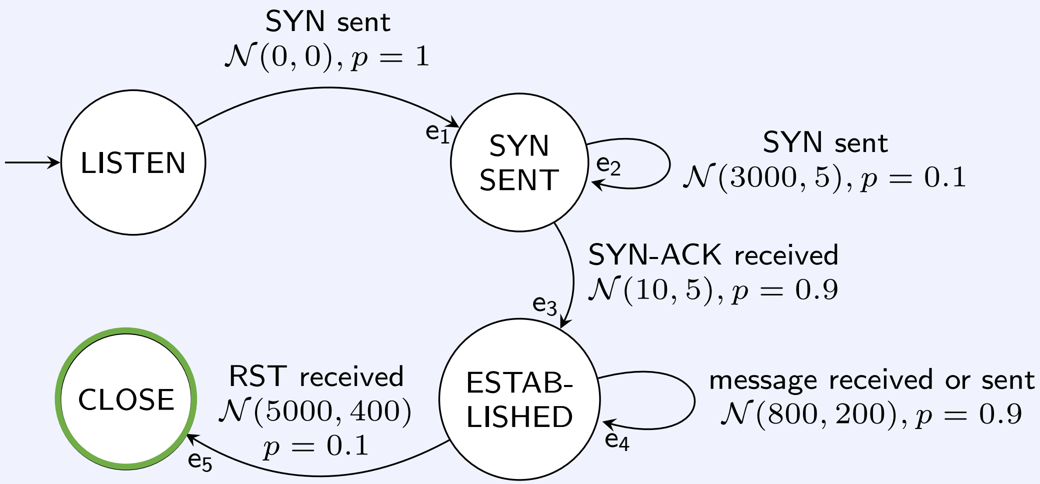
Data correction

word	edit operation sequence {operation, transition to use, delay}	corrected word
(SYN sent, 0),	(follow, e1 , 0),	(SYN sent, 0),
(ACK sent, 500),	(skip, ε, 500),	
(SYN sent, 2000),	(follow, e2 , 2000),	(SYN sent, 2000),
	(add, e3 , 10),	(SYN-ACK received, 10),
(RST received, 6500)		



Data correction

word	edit operation sequence {operation, transition to use, delay}	corrected word
(SYN sent, 0),	(follow, e1 , 0),	(SYN sent, 0),
(ACK sent, 500),	(skip, ε, 500),	
(SYN sent, 2000),	(follow, e2 , 2000),	(SYN sent, 2000),
	(add, e3 , 10),	(SYN-ACK received, 10),
(RST received, 6500)	(follow, e5 , 6500)	(RST received, 6500)



Data correction

(SYN sent, 0),(ACK sent, 500),(SYN sent, 2000),(RST received, 6500)

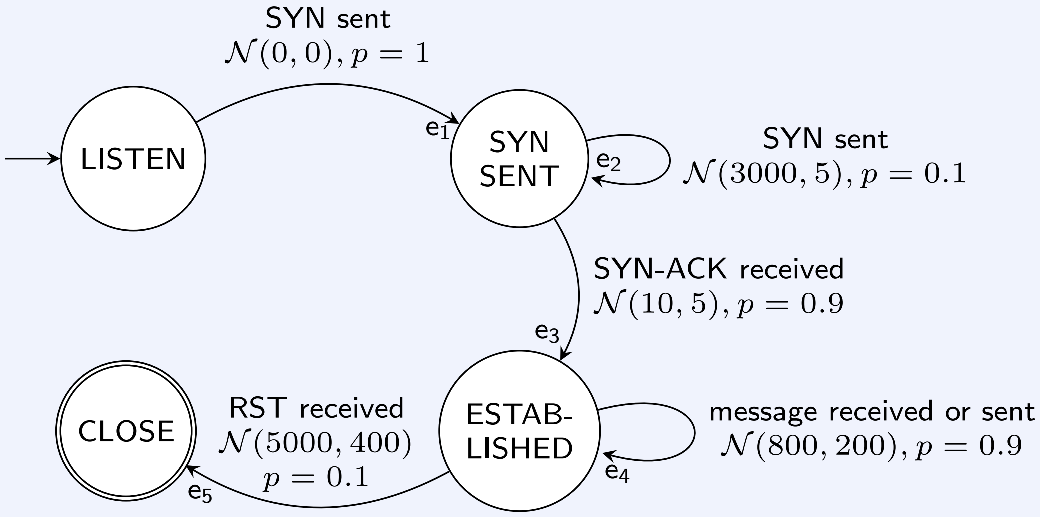
correction

(SYN sent, 0),(SYN-ACK received, 10),(RST received, 6500)

received, 10),(RST received, 6500)

...

Many correction solutions possible...



Data correction

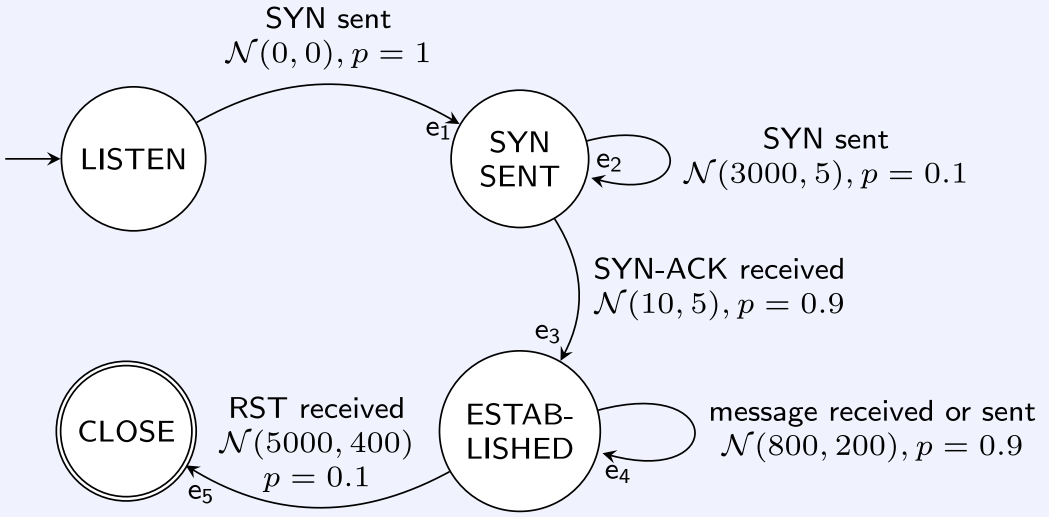
(SYN sent, 0),(ACK sent, 500),(SYN sent, 2000),(RST received, 6500)

correction

(SYN sent, 0),(SYN-ACK received, 10),(RST received, 6500)

received, 10),(RST received, 6500)

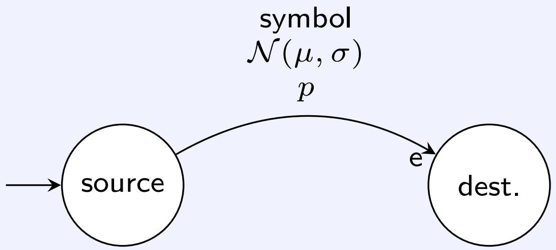
...



Many correction solutions possible…

We choose the one with the minimal description length

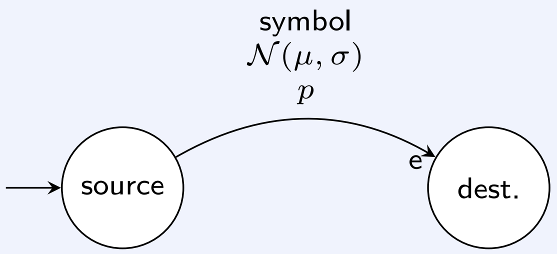
Data encoding



Cost of a followed pair (symbol, delay)

corrected with (operation, transition, delay) depends on

Data encoding



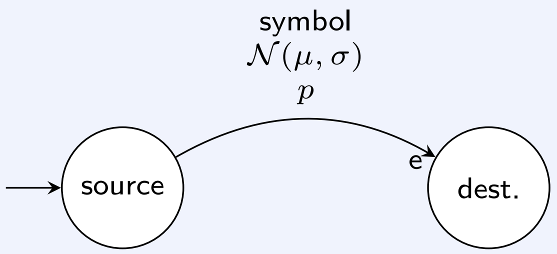
Cost of a followed pair (symbol, delay)

corrected with (operation, transition, delay) depends on

- The probability of the edit operation (follow),

$$-\log_2 p(o)$$

Data encoding



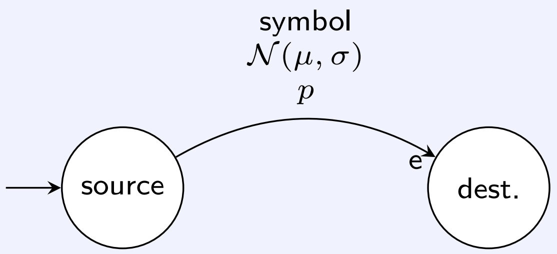
Cost of a followed pair (symbol, delay)

corrected with (operation, transition, delay) depends on

- The probability of the edit operation (follow),
- The probability of the transition given the current state,

$$-\log_2 p(o) - \log_2 p(e|q_s(e))$$

Data encoding



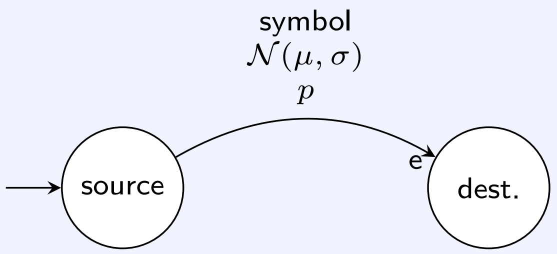
Cost of a followed pair (symbol, delay)

corrected with (operation, transition, delay) depends on

- The probability of the edit operation (follow),
- The probability of the transition given the current state,
- The probability of the delay given the transition guard's parameters.

$$-\log_2 p(o) - \log_2 p(e|q_s(e)) - \log_2 p(d|e)$$

Data encoding



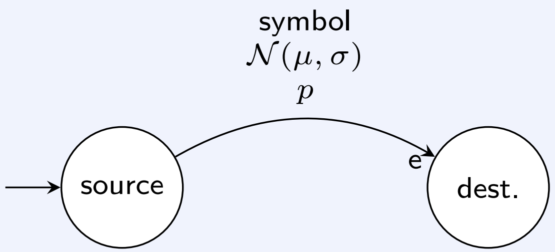
Cost of a transposed pair (symbol, delay)

corrected with (operation, transition, delay) depends on

- The probability of the edit operation (transpose),
- The probability of the transition given the current state,
- The probability of the delay given the transition guard's parameters.

$$-\log_2 p(o) - \log_2 p(e|q_s(e)) - \log_2 p(d|e)$$

Data encoding



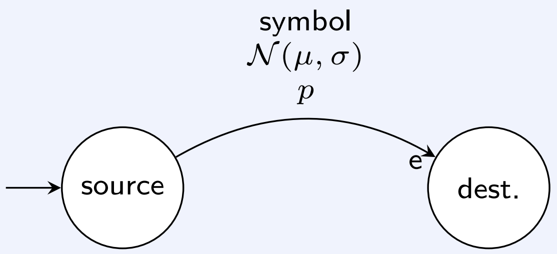
Cost of an added pair (symbol, delay)

corrected with (operation, transition, delay) depends on

- The probability of the edit operation (add),
- The probability of the transition given the current state,
- ~~The probability of the delay given the transition guard's parameters.~~

$$-\log_2 p(o) - \log_2 p(e|q_s(e))$$

Data encoding



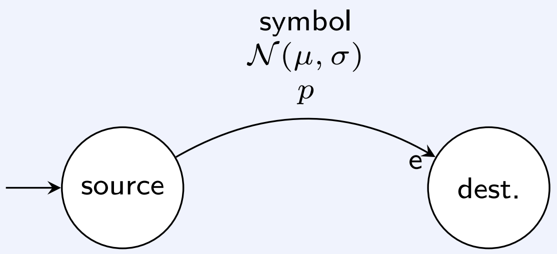
Cost of a deduplicated pair (symbol, delay)

corrected with (operation, transition, delay) depends on

- The probability of the edit operation (deduplicate),
- ~~The probability of the transition given the current state,~~
- The probability of the delay given the transition guard's parameters.

$$-\log_2 p(o) - \log_2 p(d|e)$$

Data encoding



Cost of a skipped pair (symbol, delay) corrected with (operation, ε, delay) depends on

- The probability of the edit operation (skip),
- ~~The probability of the transition given the current state,~~
- ~~The probability of the delay given the transition guard's parameters.~~
- The cost of explicitly encoding the delay and the symbol.

$$-\log_2 p(o) + L_{\mathbb{N}}(d) + \log_2 |\Sigma|$$

Question

How to find the automaton with the minimal MDL cost?

TADAM: MDL-based automata learning

Initialize an automaton $\hat{\mathcal{A}}$ with the data $\mathcal{D}$

Generate candidate automata by transforming $\hat{\mathcal{A}}$

For each candidate automaton $\mathcal{A}$

Correct $\mathcal{D}$ given $\mathcal{A}$

Compute the cost $L(\mathcal{A}, \mathcal{D})$

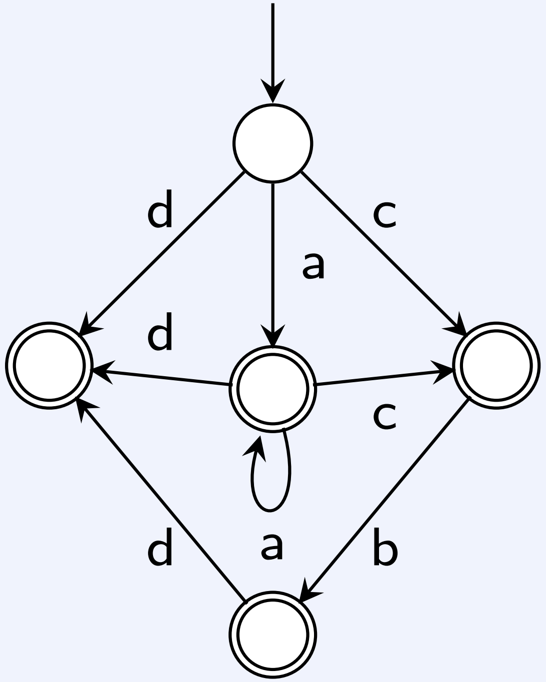
Select the automaton with minimal cost as $\hat{\mathcal{A}}$

Return $\hat{\mathcal{A}}$ when the cost doesn't descrease anymore

Initialization

Markov initialization

a c b
a d
c b a c
a c a
d a a
c b d



guards and probabilities omitted

Candidate automata generation

Automaton transformation operations:

- Location merge
- Location split
- Subpart deletion

One candidate per possible transformation and position in the automaton

Location merge

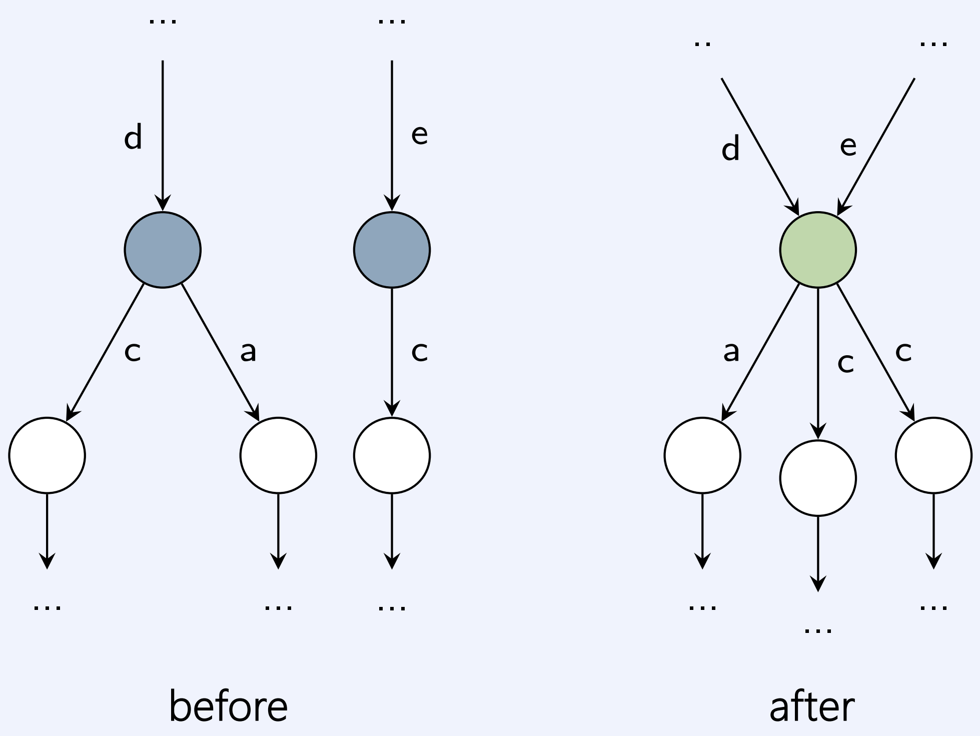
Goal:

Reducing the size of the automaton and generalize the model

(reduces the model cost)

Side effect:

Increases the data cost



guards and probabilities omitted

Location split

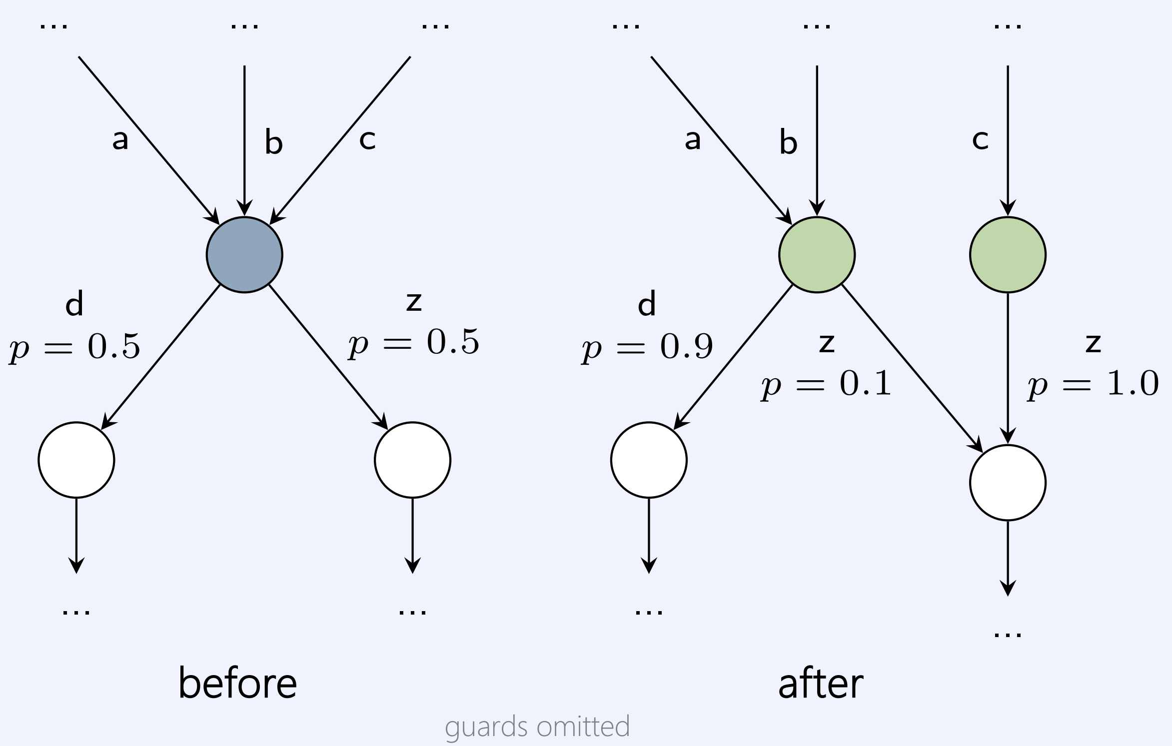
Goal:

Reducing the entropy of the "next triggered transition" at a given location

(reduces the data cost)

Side effect:

Increases the model cost



Subpart deletion

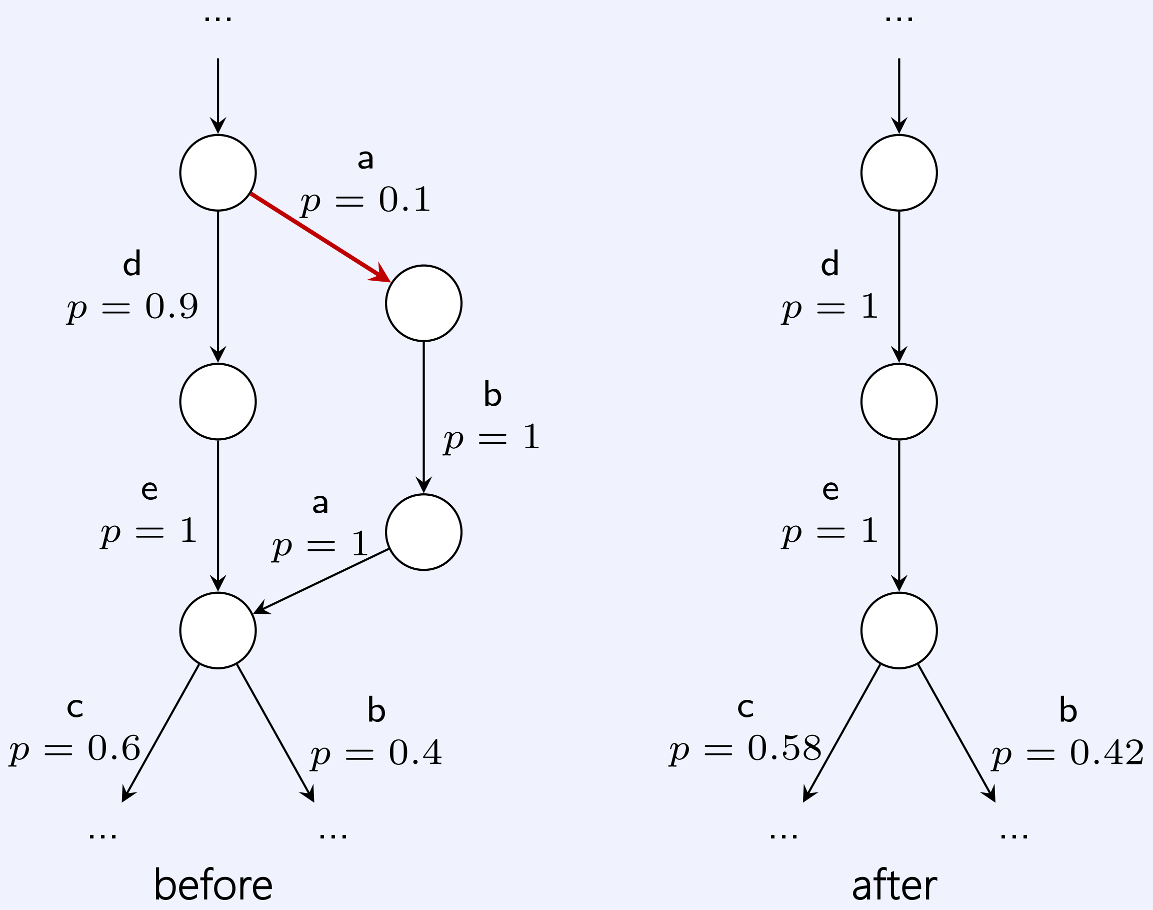
Goal:

Reducing the size of the automaton

(reduces the model cost)

Side effect:

Increases the data cost



guards omitted

Question

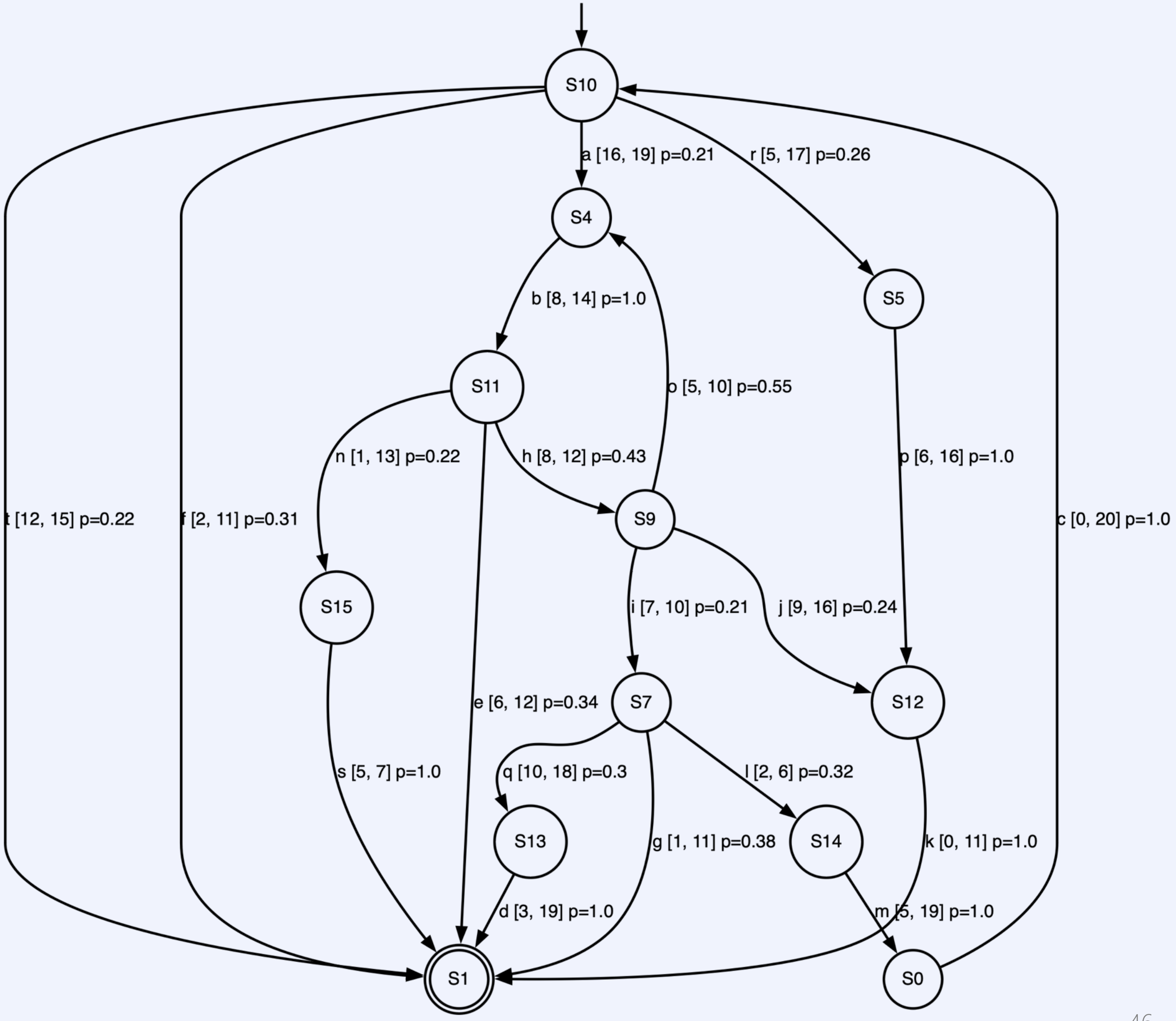
Does it work?

State of the Art

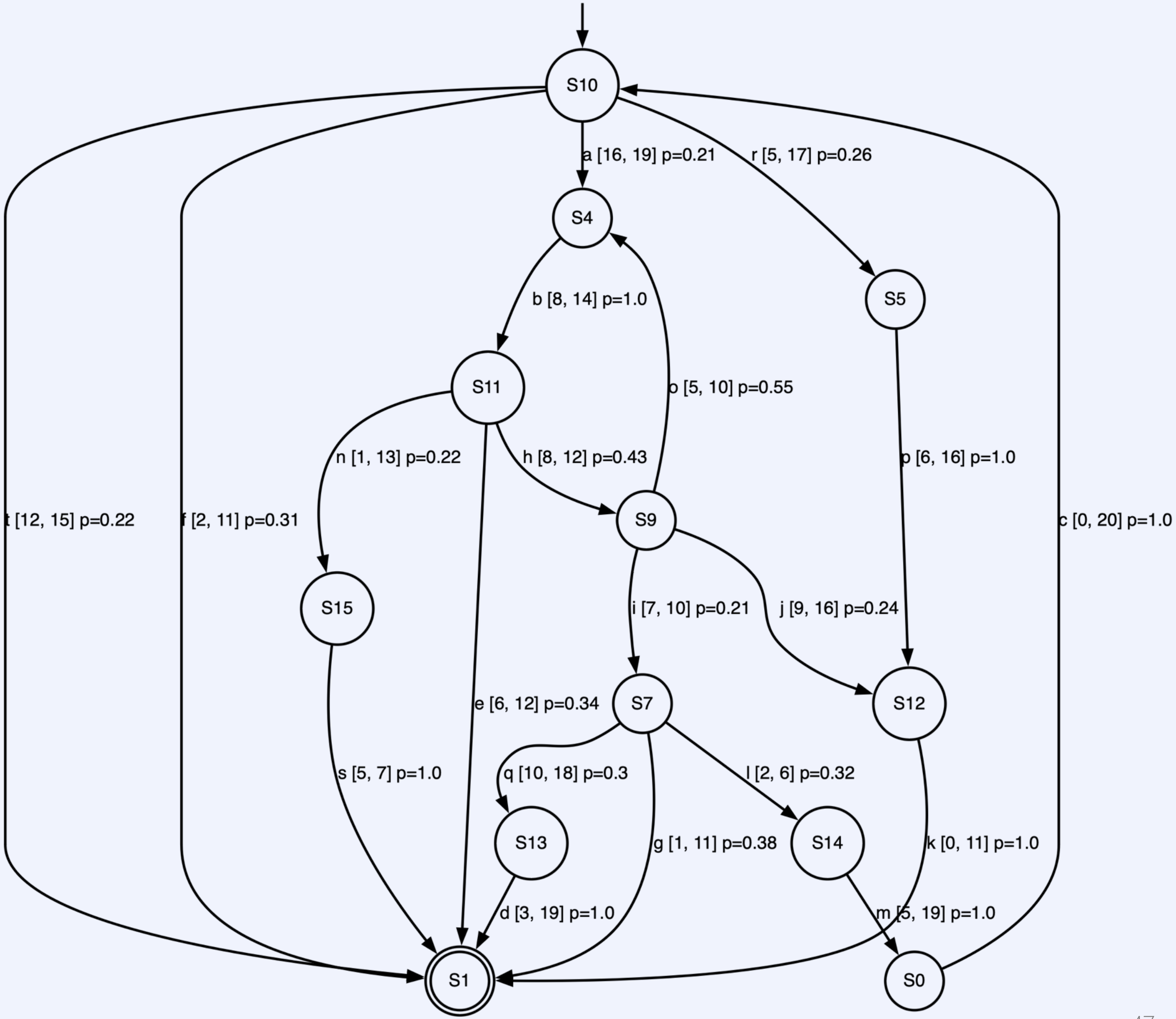
Algorithm	Strategy	Main limitation
TAG (Cornanguer et al., 2022)	Factorization on common sub-parts and location splits	
Timed k-Tail (Pastore et al., 2017)	Factorization on common sub-parts	No noise robustness strategy → requires clean data
RTI+ (Verwer et al., 2010)	Location merge based on likelihood test	

Experiments

Target automaton



Experiments



Target automaton

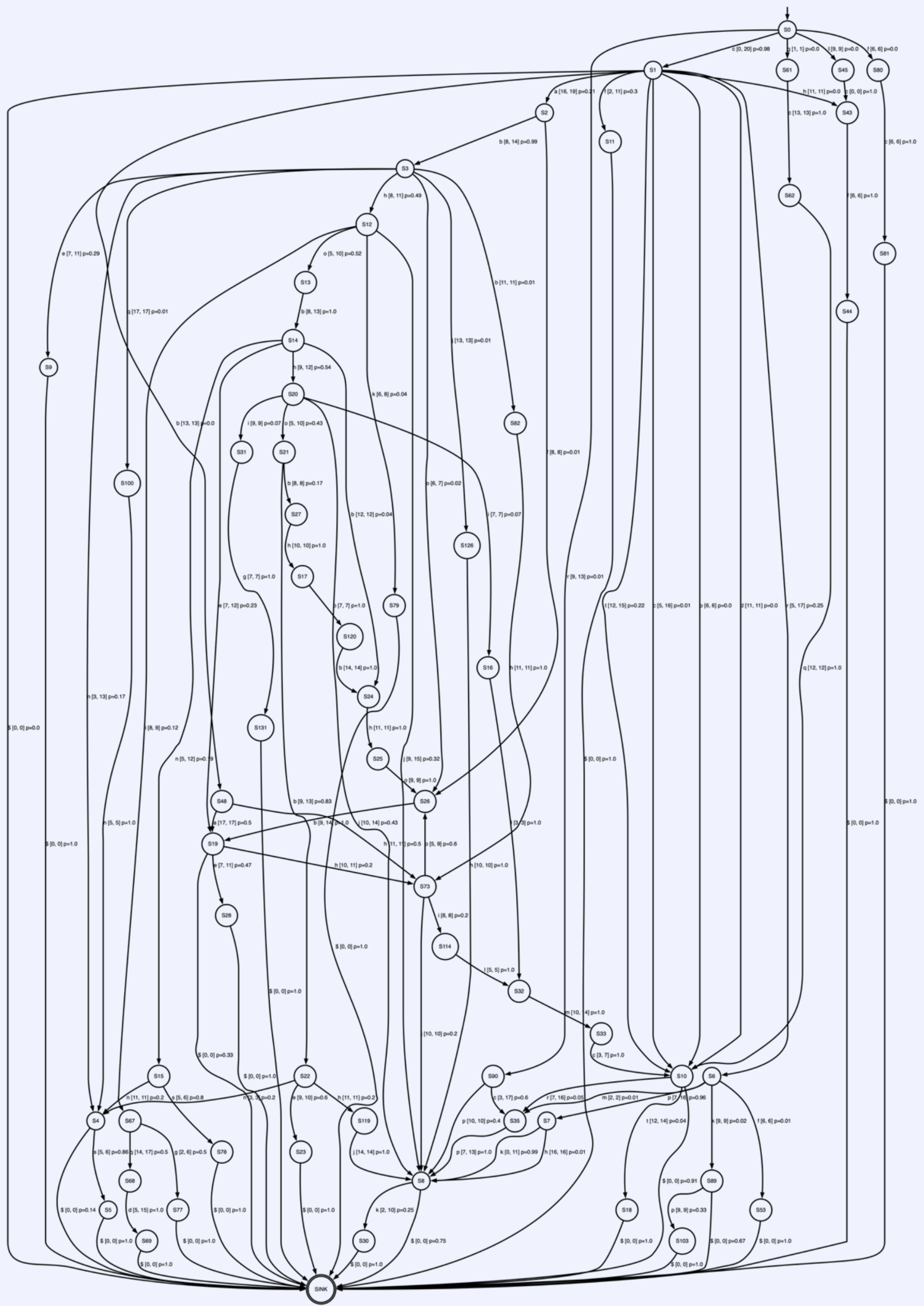
Data:

- 500 timed words
- 2,5% of noisy events

Experiments

TAG

learned automaton

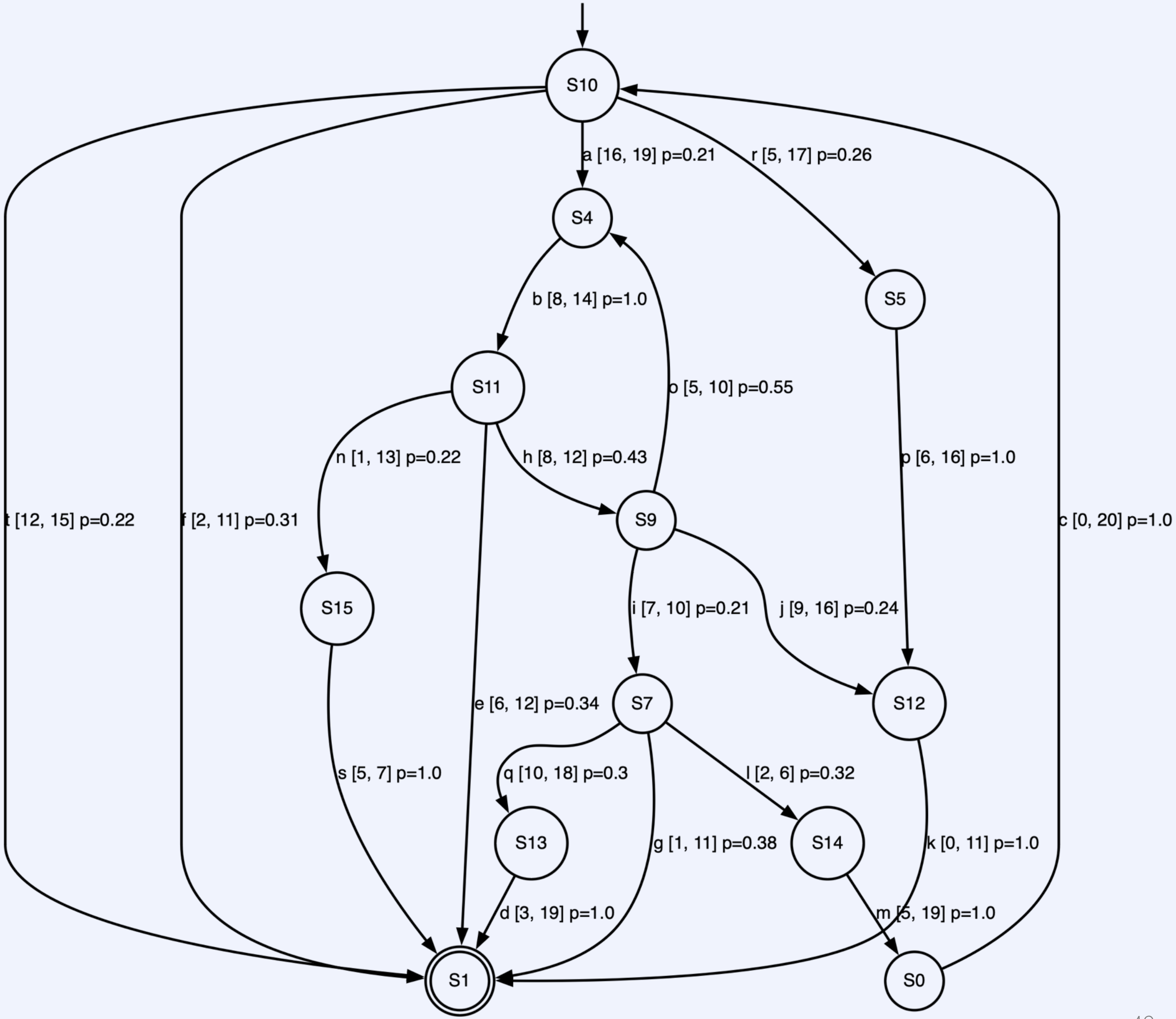


Data:

- 500 timed words
- 2,5% of noisy events

Experiments

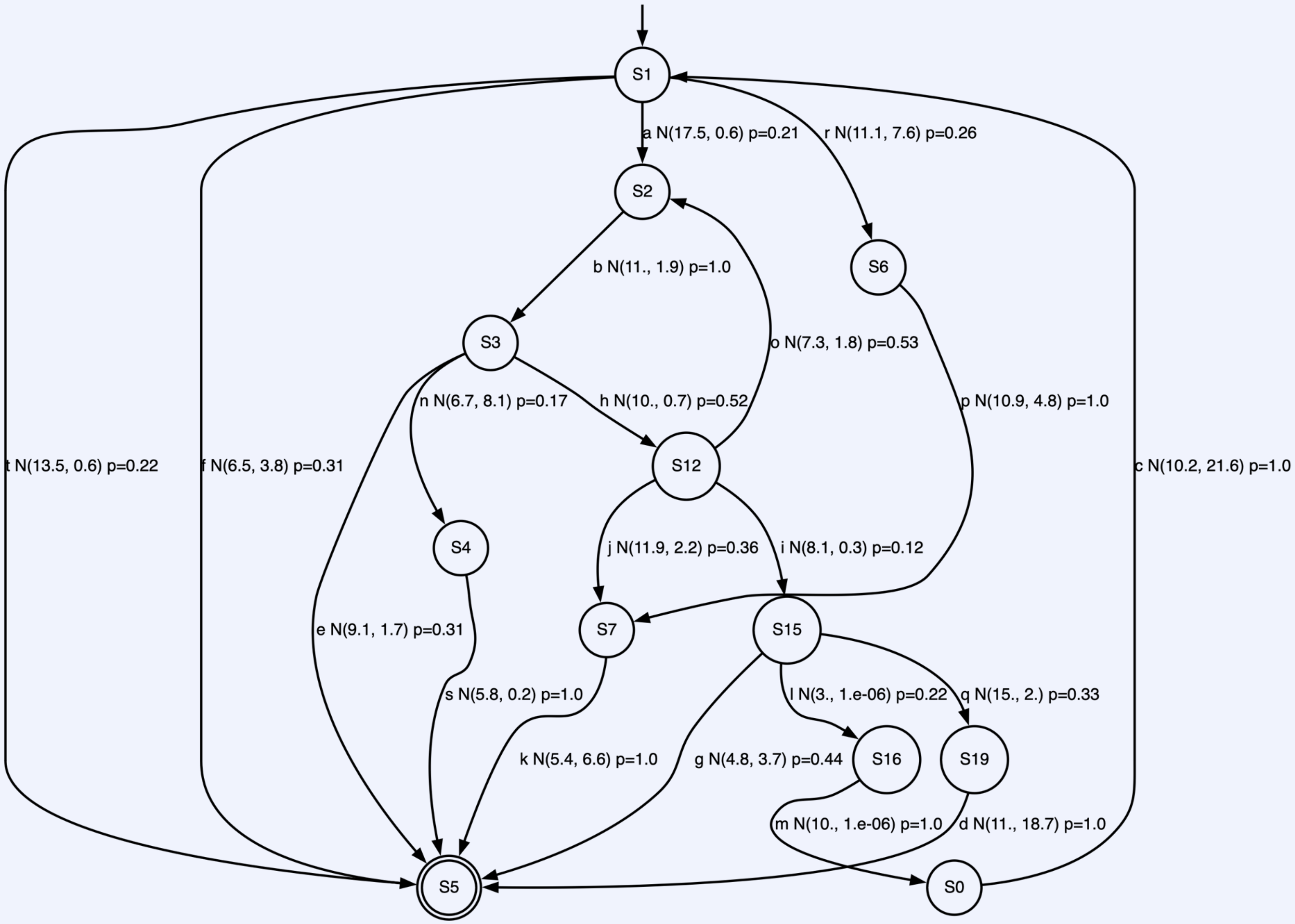
Target automaton



Experiments

TADAM

learned automaton



Data:

- 500 timed words
- 2,5% of noisy events

Experiments

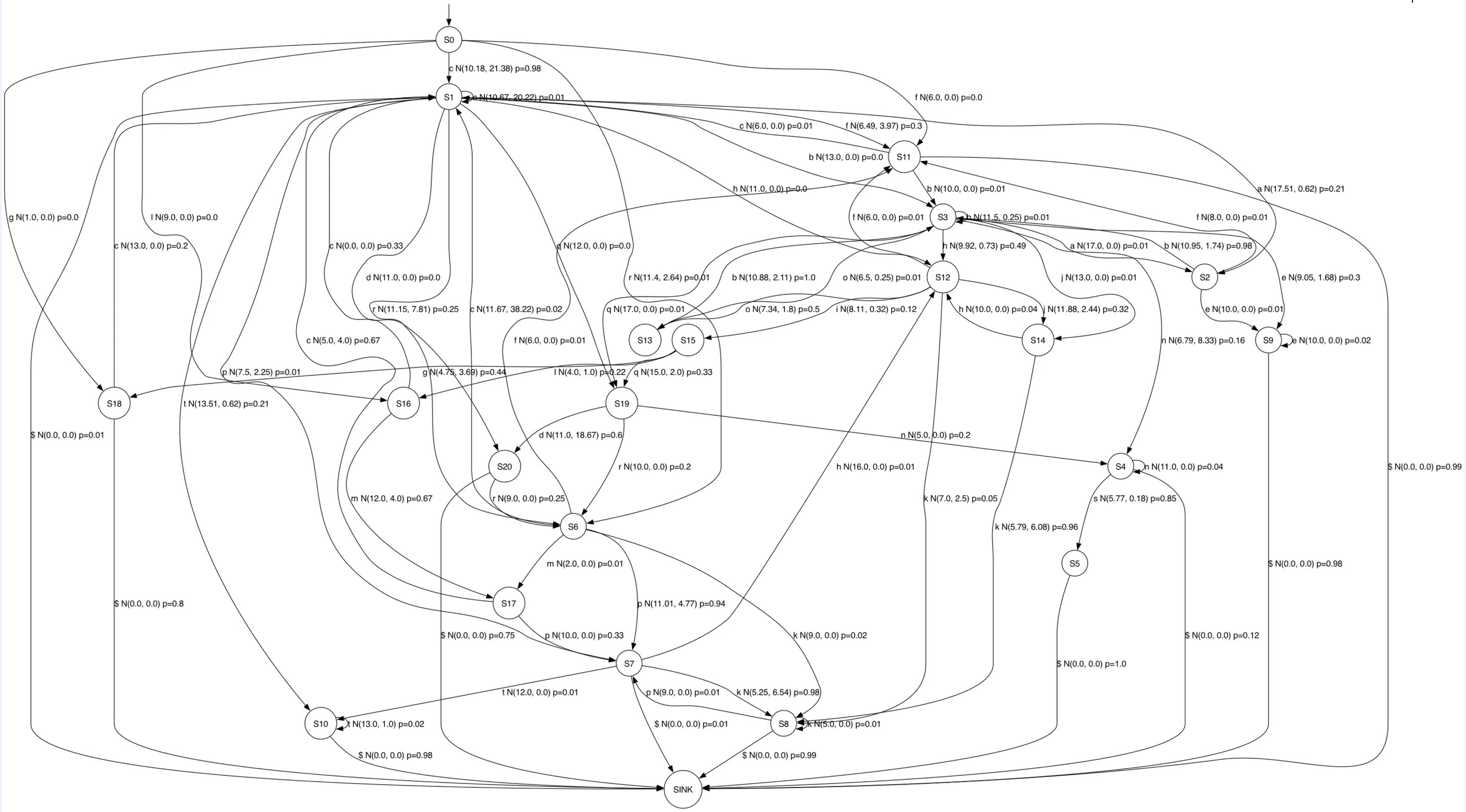
TADAM

learned automaton

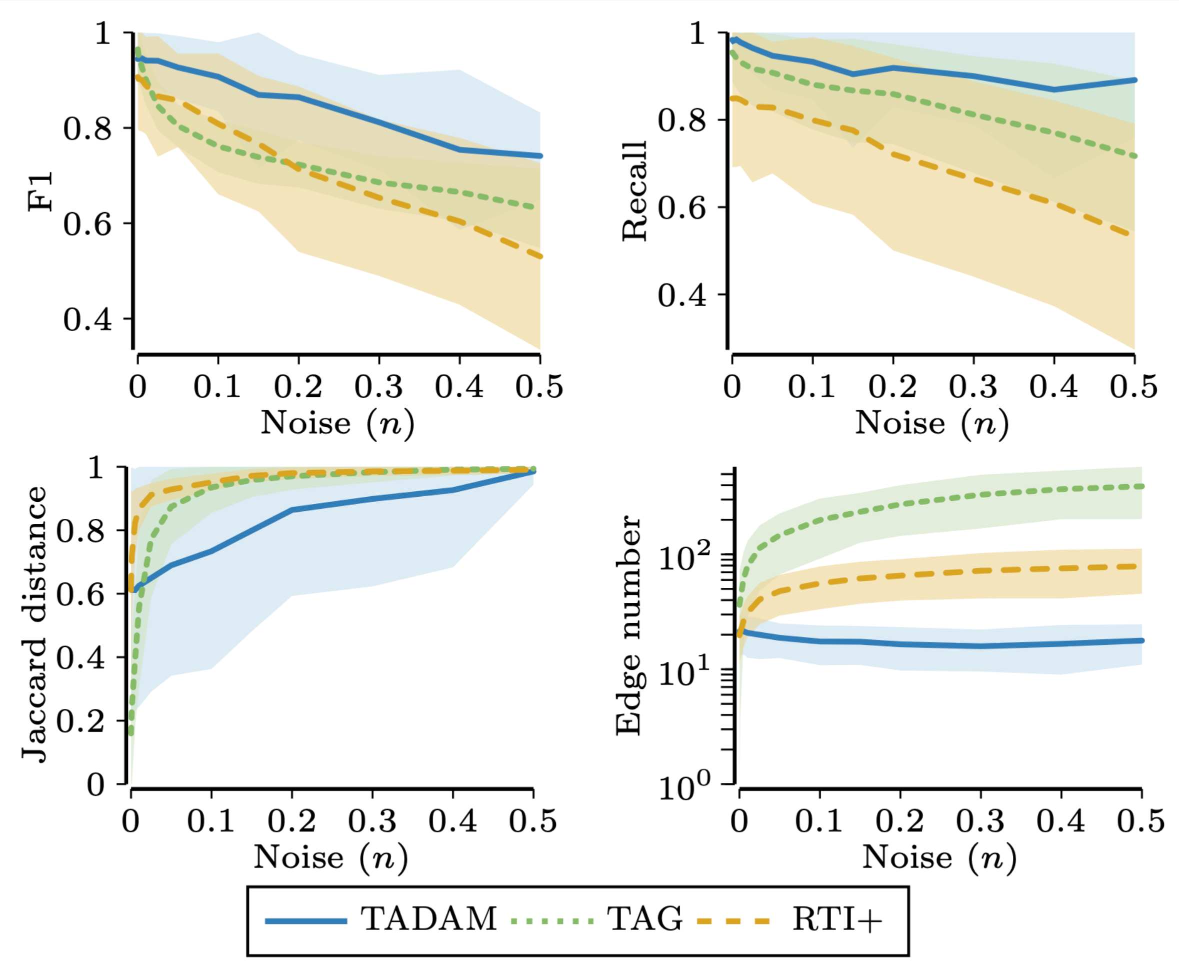
Data:

- 500 timed words
- 2,5% of noisy events

non-actual speed



Experiments



Noise robustness
on synthetic data

Experiments

	Learner	AU-ROC	TPR	FPR	F1
TA learners	TADAM	**0.982**	0.998	**0.025**	**0.705**
	TAG	0.891	**1**	0.142	0.298
	RTI+	0.790	**1**	0.292	0.171
	Hidden Markov Model	0.608	0.640	0.085	0.288

Anomaly detection performances

on HDFS dataset[1]

1. https://github.com/ait-aecid/anomaly-detection-log-datasets

Experiments

	Learner	AU-ROC	TPR	FPR	F1	
TA learners	TADAM	**0.982**	0.998	**0.025**	**0.705**	very high detection rate and less false alarms
	TAG	0.891	**1**	0.142	0.298	
	RTI+	0.790	**1**	0.292	0.171	
	Hidden Markov Model	0.608	0.640	0.085	0.288	

Anomaly detection performances

on HDFS dataset[1]

1. https://github.com/ait-aecid/anomaly-detection-log-datasets

Experiments

Learner	AU-ROC	TPR	FPR	F1
TADAM	**0.982**	0.998	**0.025**	**0.705**
TAG	0.891	**1**	0.142	0.298
RTI+	0.790	**1**	0.292	0.171
Hidden Markov Model	0.608	0.640	0.085	0.288

TA learners

very high detection rate and less false alarms

overfit on the training data and don't generalize well

Anomaly detection performances

on HDFS dataset[1]

1. https://github.com/ait-aecid/anomaly-detection-log-datasets

Experiments

	Learner	AU-ROC	TPR	FPR	F1	
TA learners	TADAM	**0.982**	0.998	**0.025**	**0.705**	very high detection rate and less false alarms
	TAG	0.891	**1**	0.142	0.298	overfit on the training data and don't generalize well
	RTI+	0.790	**1**	0.292	0.171	
	Hidden Markov Model	0.608	0.640	0.085	0.288	not expressive enough

Anomaly detection performances

on HDFS dataset[1]

1. https://github.com/ait-aecid/anomaly-detection-log-datasets

Conclusions

Contributions:

- A compression-based (MDL) score to avoid overfitting
- An explicit modelization of the noise

Experiments show that TADAM

- is far more robust to noise
- learns smaller models
- has better performances on real-world classification and anomaly detection tasks

Thank you!

Contributions:

- A compression-based (MDL) score to avoid overfitting
- An explicit modelization of the noise

Fos-R/TADAM

`pip install tadam-learner`

Experiments show that TADAM

- is far more robust to noise
- learns smaller models
- has better performances on real-world classification and anomaly detection tasks