

RHEINISCHE
FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

MASTER THESIS PRESENTATION

**Distributed Online Learning for Large-scale
Pattern Prediction over Real-time Event
Streams**

Ehab QADAH

First Examiner:

PD. Dr. Michael MOCK

Second Examiner:

Prof. Dr. Stefan WROBEL

Outline

Motivation and Problem Formulation

The Proposed Approach

Event Forecasting with Pattern Markov Chains

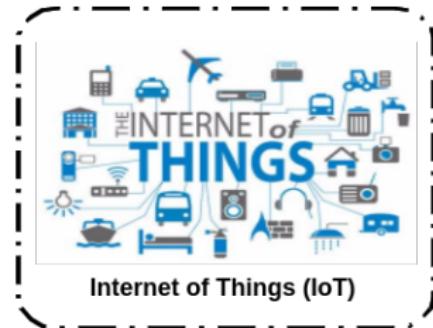
Distributed Online Learning for Pattern Prediction

Empirical Evaluation

Future Work and Conclusion

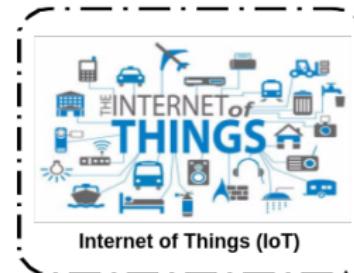
Motivation

A New Era: Big event Data streams



Motivation

A New Era: Big event Data streams



- ▶ Common Goal: recognition and prediction full matches of complex event patterns in real-time.

Problem Formulation

- ▶ Given a set of k real-time streams of events
 $S = \{s_1, s_2, \dots, s_k\}$.
- ▶ Each stream $s_i = \langle e_1, e_3, \dots, e_t, \dots \rangle$ is a time-ordered infinite sequence of events.
- ▶ Each event is defined as a tuple of attributes
 $e_i = (id, type, \tau, a_1, a_2, \dots, a_n)$, where $type \in \Sigma$ (i.e., event types), $\tau \in \mathbb{R}$, and $id \in \mathbb{N}$.
- ▶ A user-defined pattern \mathcal{P} is given in the form of a regular expression over a set of event types. Σ
 $(\mathcal{P} := E \mid \mathcal{P}_1; \mathcal{P}_2 \mid \mathcal{P}_1 \vee \mathcal{P}_2 \mid \mathcal{P}_1^*, E \in \Sigma)$.

Problem Formulation

- ▶ The main objective is to predict the pattern \mathcal{P} completion with certain probability in the future over each stream s_i given the current time event e_t .

Problem Formulation

- ▶ The main objective is to predict the pattern \mathcal{P} completion with certain probability in the future over each stream s_i given the current time event e_t .
- ▶ **(Pattern Prediction over Multiple Event Streams)**

An Illustrative Application Domain:

Maritime Surveillance

- ▶ Event tuple (i.e., critical points) derived from raw Automatic Identification System (AIS) messages of moving vessels e.g.,

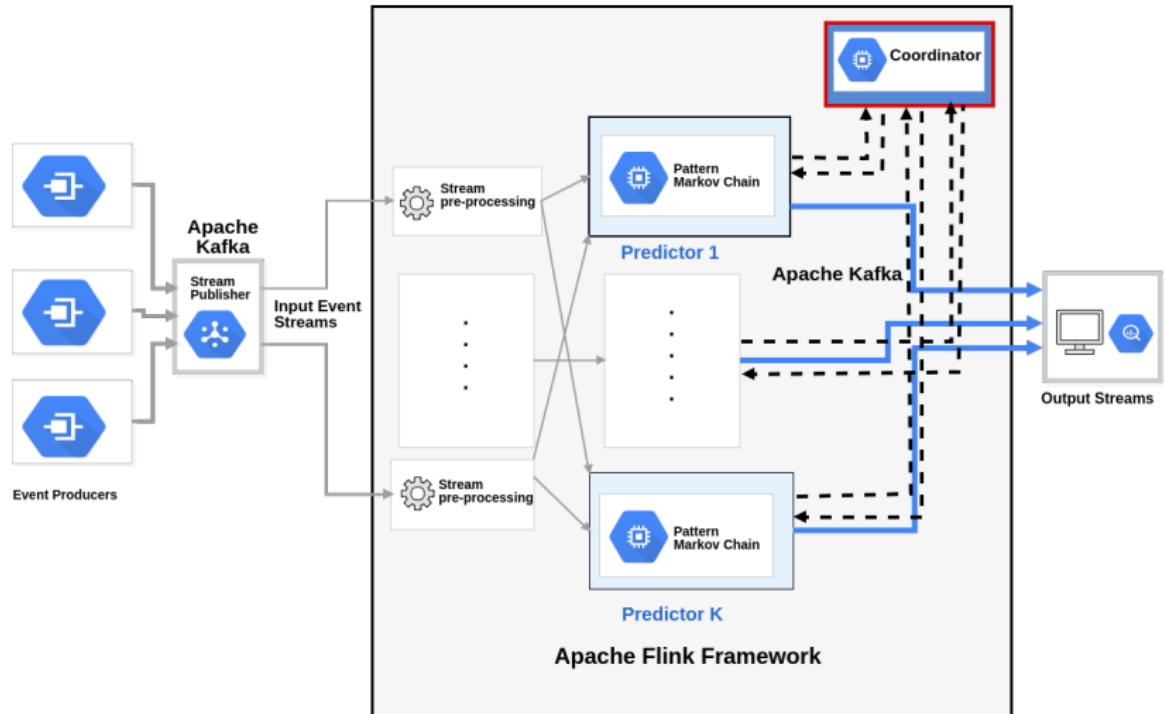
```
{  
    "timestamp":1443651492000,  
    "id": "228133000",  
    "annotation": "change_in_heading",  
    "latitude": 48.117775,  
    "longitude": -4.4205885,  
    "distance": 323.406,  
    "heading": 264.27  
    "speed": 18.48,  
}
```

- ▶ Example patterns such as:

$P_1 = Sailing$ or

$P_2 = changeInHeading; gapStart; gapEnd; changeInHeading$

System Architecture



Scalable Pattern Prediction System ²

- ▶ A scalable and distributed system that provides online pattern prediction over multiple real-time streams of events.
- ▶ The proposed system is based on a novel method that combines online probabilistic prediction models based on pattern Markov chain technique [1] with a distributed online learning protocol [5] to learn a global prediction model in a communication-efficient way.
- ▶ For large-scale processing support, the system is implemented on top of Apache Flink [2] along with Apache Kafka [4].
- ▶ Developed in the context of the datAcron project ¹.

¹<http://www.datacron-project.eu/>

²Source code: <https://goo.gl/BZ2Prk>.

Event Forecasting with Pattern Markov Chains [1]

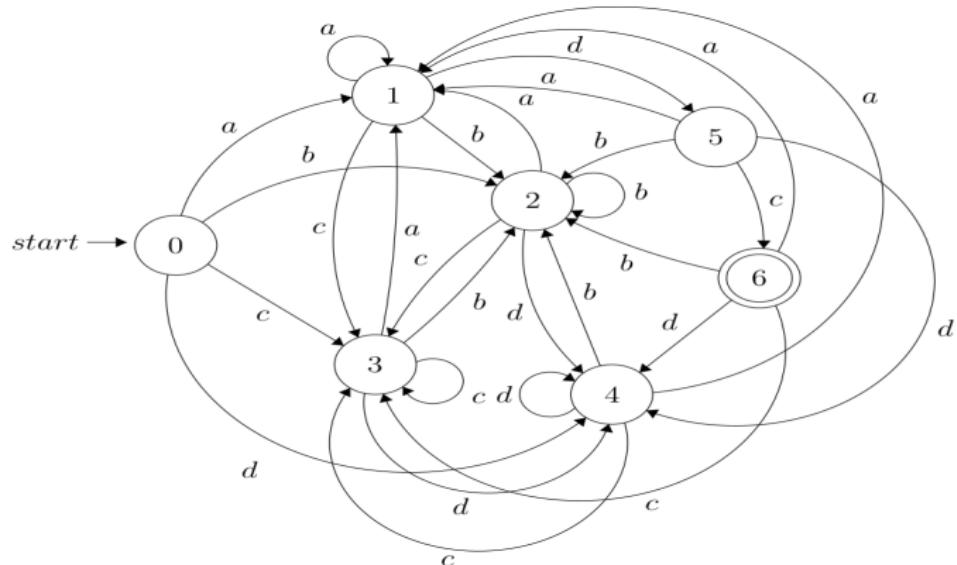
overview

- ▶ The model handles a single input stream s_i of events.
- ▶ The event stream s_i is assumed to be generated by m -order Markov source.
- ▶ The event pattern \mathcal{P} is defined in the form of regular expressions over a finite set of event types Σ .
- ▶ A probabilistic model provides online forecasting reports when the \mathcal{P} is expected to be completed in future.

Event Forecasting with Pattern Markov Chains

How does it work?

- The pattern $\mathcal{P} = a; d; c$ is converted to deterministic finite automaton (DFA) with $\Sigma = \{a, b, c, d\}$.



$$Q = \{0, 1, 2, 3, 4, 5, 6\} \text{ for } m = 1$$

Event Forecasting with Pattern Markov Chains

How does it work?

- ▶ The *DFA* is used to construct a Markov chain, which is called a Pattern Markov Chain (PMC_m^P).
- ▶ The states of *DFA* is directly mapped to states of a transition probability matrix Π $|Q| \times |Q|$ of the PMC_m^P .
- ▶

$$\Pi = \begin{Bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ 6 \end{Bmatrix} \begin{pmatrix} p_{0,0} & \cdot & \cdot & \cdot & p_{0,6} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & p_{6,6} \end{pmatrix}$$

- ▶ The maximum-likelihood estimator is used to compute the transition probabilities $p_{i,j}$ of the matrix Π

$$\hat{p}_{i,j} = \frac{n_{i,j}}{\sum_{k \in Q} n_{i,k}} = \frac{n_{i,j}}{n_i} \quad (1)$$

Event Forecasting with Pattern Markov Chains

Constructing the Pattern Prediction Model

- ▶ The probability distribution of the waiting-time (i.e., time required until the pattern is completed from state q) $P(W_{\mathcal{P}}(q) = n)$, is calculated based on the Markov chain transition matrix.

$$P(W_{\mathcal{P}}(q) = n) = \xi_i^T \mathbf{N}^{n-1} (\mathbf{I} - \mathbf{N}) \mathbf{1}$$

where

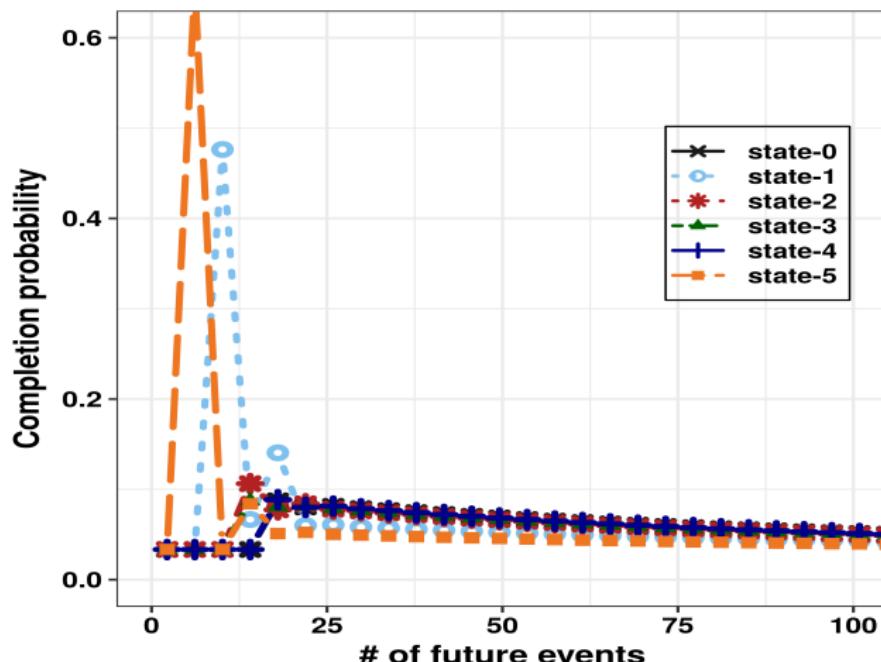
$$\boldsymbol{\Pi} = \begin{pmatrix} \mathbf{N} & \mathbf{C} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (2)$$

- ▶ The prediction intervals $I = (start, end)$ (i.e., the completion interval of the pattern \mathcal{P} from the current state q) are built using the waiting-time distribution

$$P(I) = \sum_{n \in I} W_{\mathcal{P}}(q) = n, \text{ and } P(I) \geq \theta_p \quad (end - start) \leq \theta_s$$

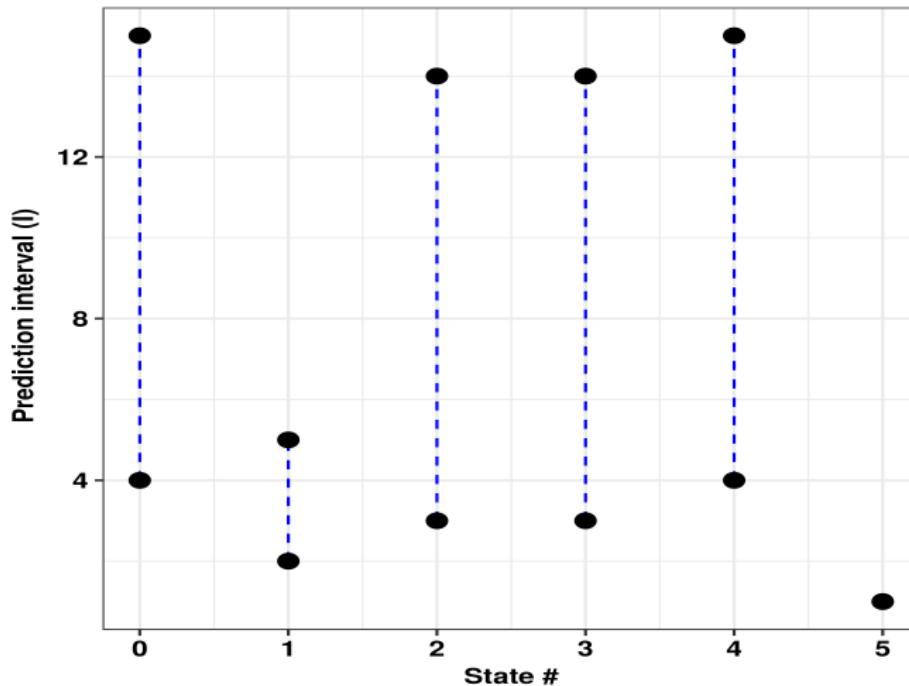
Event Forecasting with Pattern Markov Chains

Waiting-time distribution for $\mathcal{P} = a; d; c$, $\Sigma = \{a, b, c, d\}$, $m = 1$.



Event Forecasting with Pattern Markov Chains

Example of the computed prediction intervals for $\mathcal{P} = \{a; d; c\}$, $\Sigma = \{a, b, c, d\}$, $m = 1$, $\theta_p = 0.5$, and $\theta_s = 20$.



Event Forecasting with Pattern Markov Chains

What is wrong?

- ▶ The overhead learning time of the Markov transition matrix for each input stream isolated from the others, how can we accelerate it?
- ▶ The low performance of the model for an input event stream with less data.
- ▶ How can we enable the communication between the local models and leverage an aggregated global shared model in a distributed and communication efficient fashion while maintaining provable and strong guarantees in service quality?

Distributed Online Learning for Pattern Prediction

Algorithm 1: Communication-efficient Distributed Online Learning [5] .

Predictors:

```
node  $n_i$ : at observing event  $e_j$ 
    update the parameters of the local prediction model  $w_i$  and provide a
    prediction interval  $I$  ;
    if  $j \bmod b = 0$  and  $\|w_i - w_r\|^2 > \Delta$  then
        send  $w_i$  to the Coordinator (violation) ;
```

Coordinator:

```
receive parameters of local models with violation  $B = \{w_i\}_{i=1}^m$  ;
while  $|B| \neq k$  and  $\frac{1}{|B|} \sum_{w_i \in B} \|w_i - \hat{w}\|^2 > \Delta$  do
    add other nodes that have not reported violation for
    their models  $B \leftarrow \{w_l : w_l \notin B \text{ and } l \in [k]\}$  ;
    receive models from nodes in  $B$ ;
    compute a new global model  $\hat{w}$  ;
    send  $\hat{w}$  to all the predictors in  $B$  and set  $w_1 \dots w_m = \hat{w}$ ;
    if  $|B| = k$  then
        set a new reference model  $w_r \leftarrow \hat{w}$  ;
```

Communication-Efficient Distributed Online Prediction by Dynamic Model Synchronization [5]

- ▶ A protocol for distributed online prediction over multiple input data streams in a communication efficient manner.
- ▶ It allows to combine local models into a global model using a *synchronization operation*.
- ▶ The distributed learners exchange their local model with a central coordinator node periodically after observing a fixed number of data points (i.e., mini-batches) [3].
- ▶ A dynamic synchronization scheme based on monitoring the local models variance from a global reference model ($\|f_i - r\|^2 \leq \Delta$).

Distributed Online Learning for Pattern Prediction

- The *synchronization Operation* for the global transition probability matrix $\hat{\Pi}$:

$$\hat{\pi}_{i,j} = \frac{\sum_{k \in K} n_{k,i,j}}{\sum_{k \in K} \sum_{l \in L} n_{k,i,l}}$$

Where $n_{k,i,j}$ the number of transitions from state i to state j on node k .

Distributed Online Learning for Pattern Prediction

- ▶ The divergence of the local models from the reference model $\|w_i - w_r\|^2$ is calculated by the sum of square difference between the local transition probability matrix Π_i and the reference transition matrix Π_r :

$$\|w_i - w_r\|^2 = \|\Pi_i - \Pi_r\|^2 = \sum_{l,j} (\pi_{i,l,j} - \pi_{r,l,j})^2$$

Distributed Online Learning for Pattern Prediction

Probabilistic Learning Guarantee

For k distributed $PMC_m^{\mathcal{P}}$ predictors:

$$\Pr(|\hat{\pi}_{i,j} - p_{i,j}| \geq c) \leq \frac{1}{k} \Pr(|\hat{p}_{i,j} - p_{i,j}| \geq c)$$

Where $\hat{\pi}_{i,j}$ is the global estimation and $\hat{p}_{i,j}$ is an estimation by an isolated predictor.

Empirical evaluation

Evaluation Metrics

- ▶ $Precision = \frac{\# \text{ of correct predictions}}{\# \text{ of total predictions}}$. The fraction of the predictions that are correct.
- ▶ $Spread = end(I) - start(I)$. The width of the prediction interval I , which represents the number of events between the start and the end of I .
- ▶ $PS\text{-score} = \alpha * precision + (1 - \alpha) * (1 - \frac{spread}{max\ spread})$. In analogy with the $F\text{-score}$ we introduce this new score to assess the performance of the system in terms of both the *precision* and the *spread* combined.
- ▶ $Distance = start(I) - now$. The distance between the start of the prediction interval I and the time of prediction is produced (now).

Empirical evaluation

Evaluation Metrics

- ▶ *Cumulative communication.* The number of messages, which are required to perform the distributed online learning modes to synchronize the prediction models.
- ▶ *Throughput.* The number of events processed per unit time (second).

Empirical evaluation

Synchronization Schemes

Our proposed system can operate in three different modes of synchronization schemes:

- ▶ continuous full synchronization for each incoming event (hypothetical).
- ▶ static scheme based on synchronizing the prediction models periodically every b of input events in each stream.
- ▶ dynamic synchronization protocol based on making the predictors communicate their local prediction models periodically but only under condition that the divergence of the local models from a reference model exceeds a variance threshold Δ (recommended).

Empirical evaluation

Over Real-word Event Streams

- ▶ Event streams describe critical points (i.e., synopses) of moving vessels trajectories, which are derived from raw AIS messages as described in [6].
- ▶ 4,684,444 critical points in Brest, France: October 2015 to March 2016 (\approx 5000 vessels).
- ▶ $\mathcal{P}_1 = \text{Sailing}$ with $m = 2$ over $\Sigma_1 = \{\text{VerySlow}, \text{Slow}, \text{Moving}, \text{Sailing}, \text{Stopping}\}$.

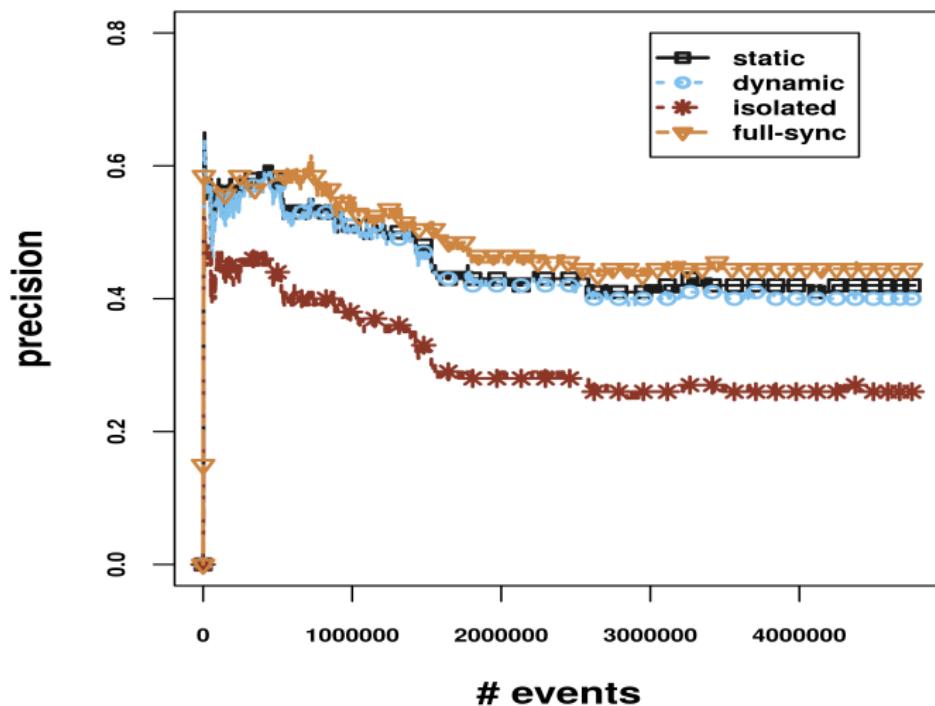
Empirical evaluation

Over Real-word Event Streams

- ▶ $\mathcal{P}_2 = \text{changeInHeading}; \text{gapStart}; \text{gapEnd}; \text{changeInHeading}$ with $m = 1$ over $\Sigma_2 = \{\text{stopStart}, \text{stopEnd}, \text{changeInSpeedStart}, \text{changeInSpeedEnd}, \text{slowMotionStart}, \text{slowMotionEnd}, \text{gapStart}, \text{gapEnd}, \text{changeInHeading}\}.$
- ▶ All experiments are performed with setting the batch size of 100 ($b = 100$), the variance threshold of 2 ($\Delta = 2$), 80% as PMC prediction threshold ($\theta_p = 80\%$), and 200 for the maximum spread ($\theta_s = 200$)

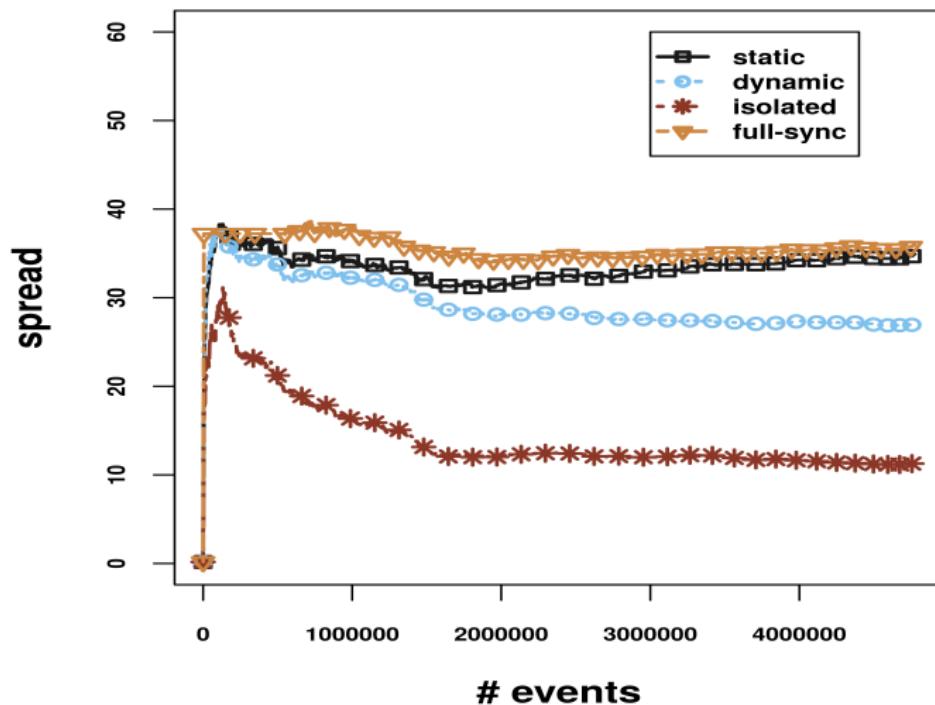
Results on Real-word Event Streams

Precision scores with respect to the number of input events over time for \mathcal{P}_1 .



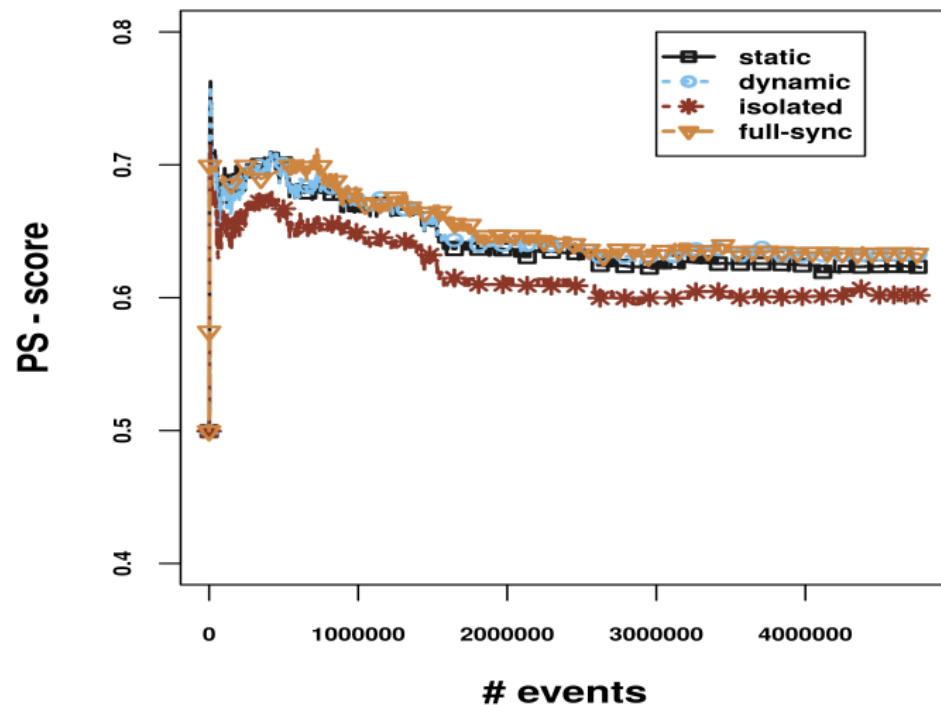
Results on Real-word Event Streams

Average spread for \mathcal{P}_1 .



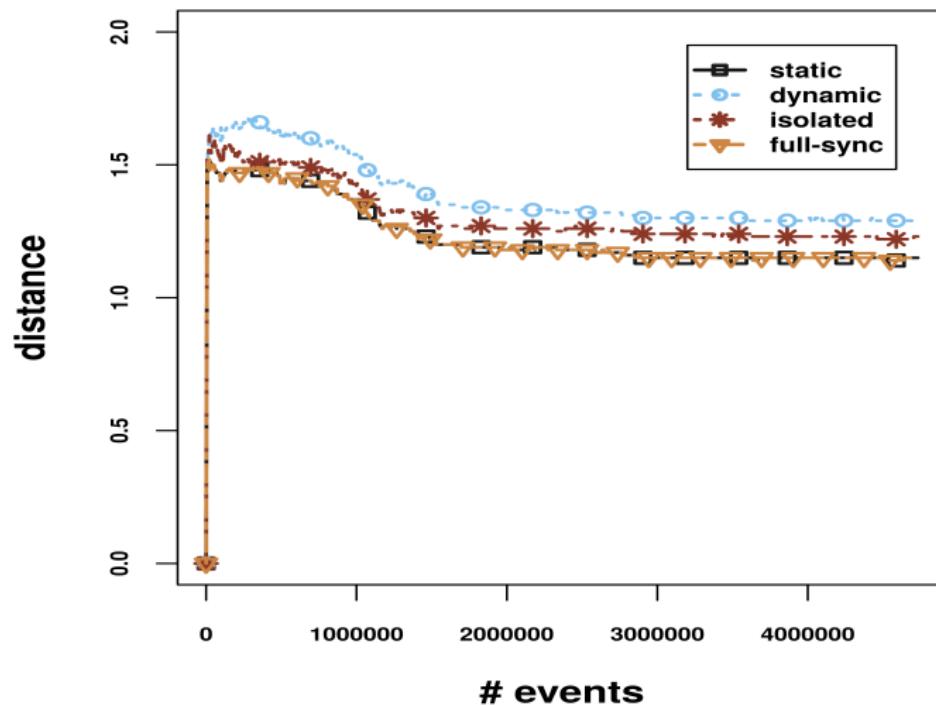
Results on Real-word Event Streams

PS -score for \mathcal{P}_1 with $\alpha = .5$.



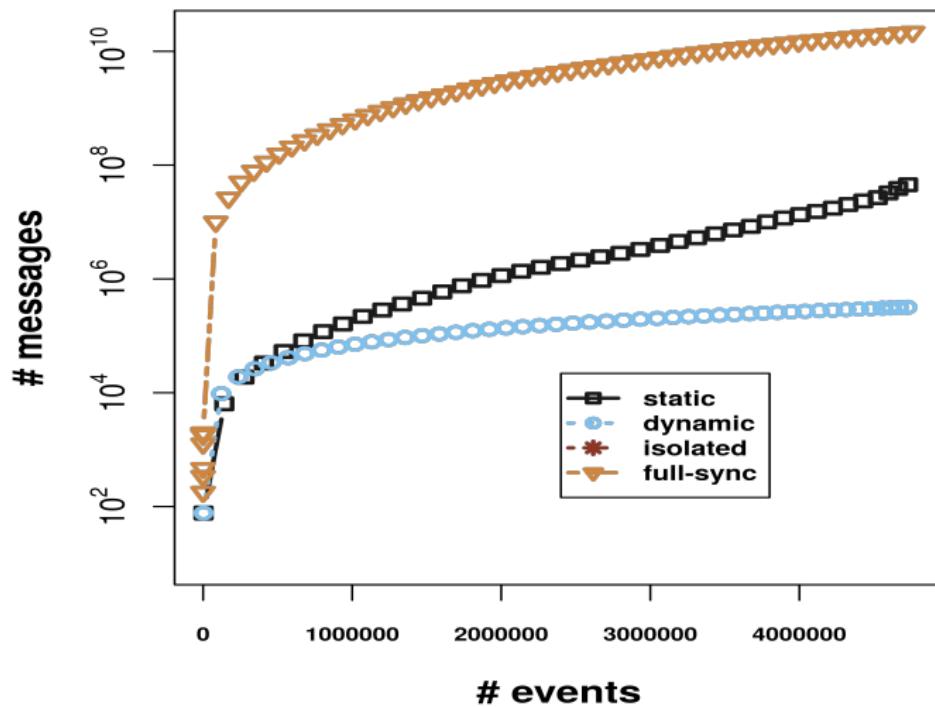
Results on Real-word Event Streams

Distance for \mathcal{P}_1 .



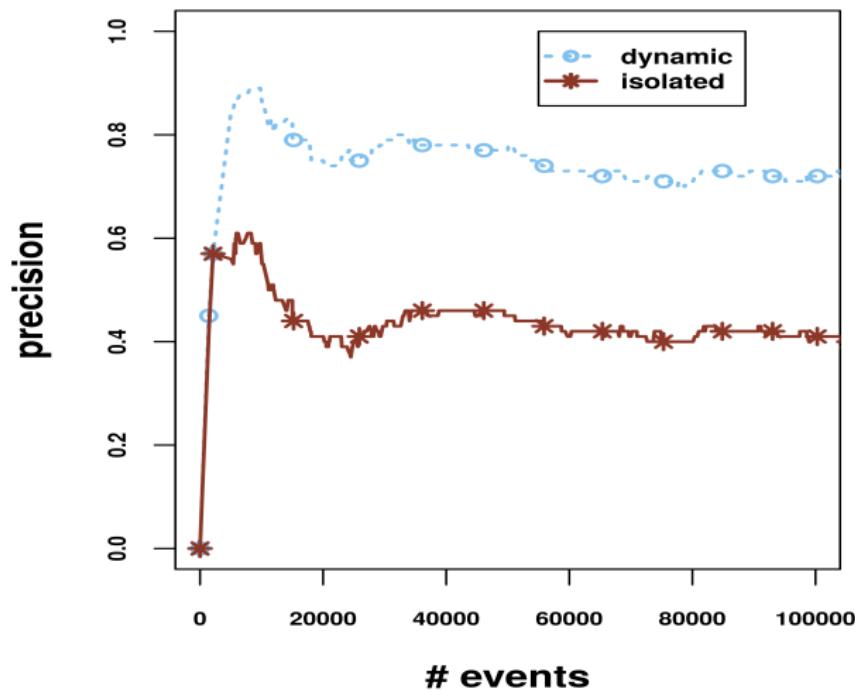
Results on Real-word Event Streams

Cumulative communication with respect to the number of input events over time for \mathcal{P}_1 .



Results on Real-word Event Streams

Precision scores of \mathcal{P}_2 for vessels of *pleasure craft* type.



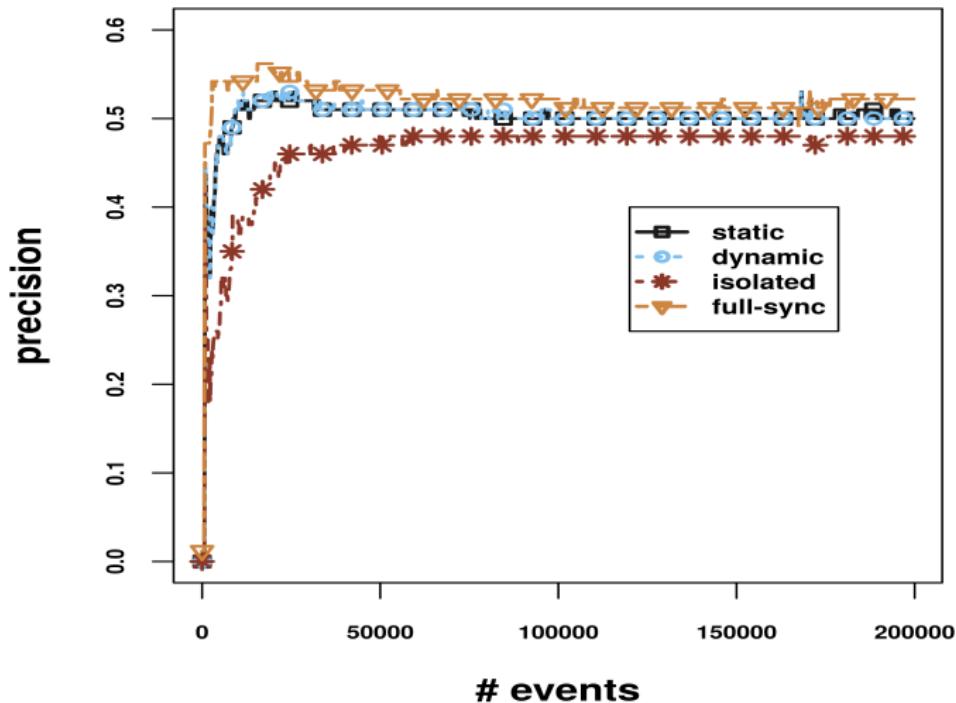
Empirical evaluation

Over Synthetic Event Streams

- ▶ We generate 20 streams of size 10,000 events from a simulated 1-order Markov process over $\Sigma = \{a, b, c, d\}$.
- ▶ The used pattern $\mathcal{P} = a; d; c$ with $m = 1$.
- ▶ We set the batch size to 20 ($b = 20$), the variance threshold to .0001 ($\Delta = .0001$), the $PMC_m^{\mathcal{P}}$ prediction threshold to 50% ($\theta_p = 50\%$), and the maximum spread to 10 ($\theta_s = 10$).

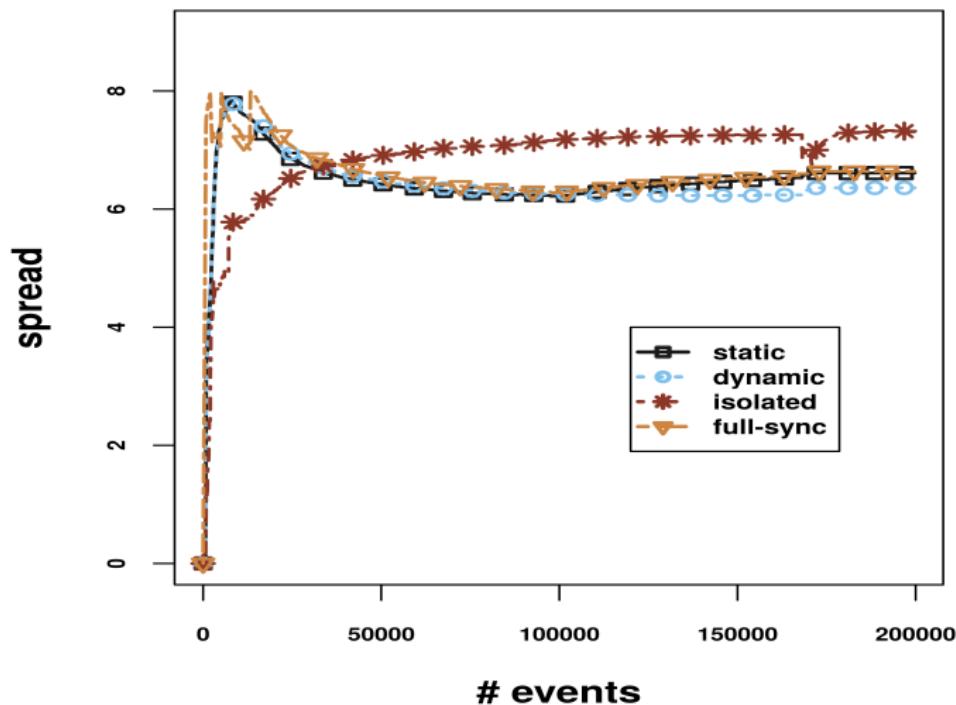
Results on Synthetic Event Streams

Precision scores with respect to the number of input events over time for $\mathcal{P} = a; d; c$.



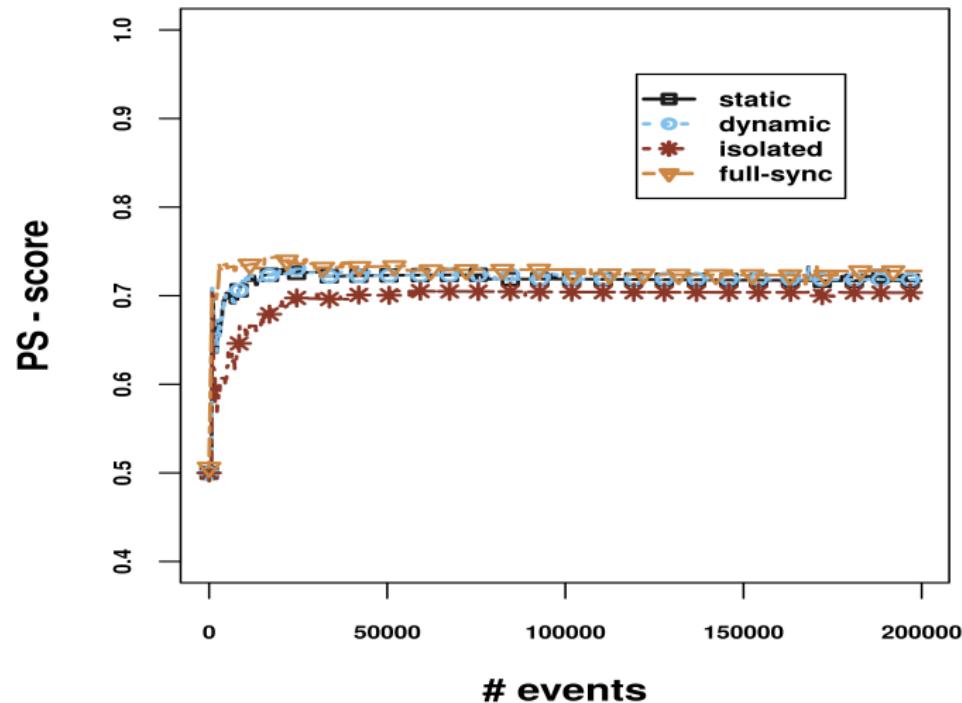
Results on Synthetic Event Streams

Average spread with respect to the number of input events over time for $\mathcal{P} = a; d; c.$



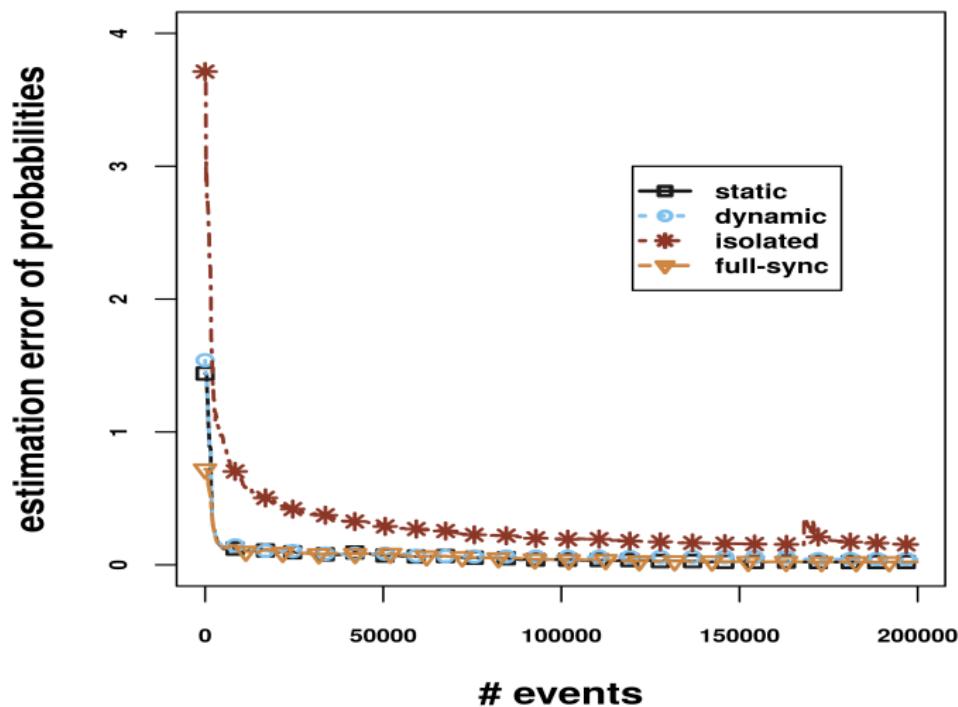
Results on Synthetic Event Streams

PS-score for $\mathcal{P} = a; d; c$ with $\alpha = .5$.



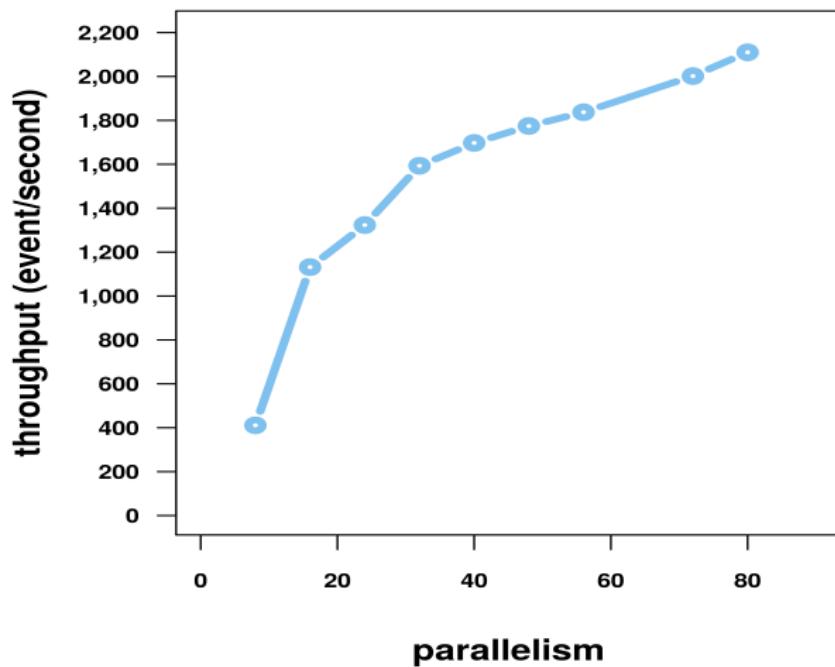
Results on Synthetic Event Streams

The error $\sum_{i,j} |\hat{p}_{i,j} - p_{i,j}|$ of estimating the transition probabilities for $\mathcal{P} = a; d; c$.



Throughput Results

Throughput of the system on YARN cluster in terms of number of events processed per second with respect to the parallelism level over \mathcal{P}_1 with batch size $b = 100$ and divergence threshold $\Delta = 2$.



Future Work

- ▶ In some applications the input event streams may belong to different distributions, we propose to dynamically divide the input event streams into similar groups (clusters).
- ▶ Introducing a weighted based synchronization operation. For example, the transition counts could be weighted by the number of the full matches of the monitored pattern in the associated event stream.

Future Work

- ▶ Provide real-time predictions intervals by also predicting the time-stamp of the future events using some machine learning techniques.
- ▶ The derived probabilistic learning guarantee relies on the static-like synchronization scheme ($\Delta = 0$). Therefore, it would be interesting to study the effects of the dynamic synchronization scheme on the learning guarantee ($\Delta > 0$).

Summary

- ▶ We have presented a system that provides a distributed pattern prediction over multiple large-scale event streams.
- ▶ The system uses the event forecasting with pattern Markov chain (PMC) [1] as the base prediction model on each event stream, and it applies the protocol for distributed online prediction [5] to exchange the information among the distributed predictors.

Summary

- ▶ The system has been implemented using Apache Flink and Apache Kafka.
- ▶ Experimental results over synthetic event streams and large real-world event streams related to trajectories of moving vessels show the effectiveness of our approach.

Publication

Parts of this thesis have been published in [7]:

Ehab Qadah, Michael Mock, Elias Alevizos, and Georg Fuchs. A Distributed Online Learning Approach for Pattern Prediction over Movement Event Streams with Apache Flink.

In *EDBT/ICDT Workshops*, 2018.

Bibliography I

- [1] Elias Alevizos, Alexander Artikis, and George Paliouras. Event forecasting with pattern markov chains. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems*, pages 146–157. ACM, 2017.
- [2] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 36(4), 2015.
- [3] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012.
- [4] The Apache Software Foundation. Apache Kafka.
<https://kafka.apache.org/>, 2012.

Bibliography II

- [5] Michael Kamp, Mario Boley, Daniel Keren, Assaf Schuster, and Izchak Sharfman. Communication-efficient distributed online prediction by dynamic model synchronization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 623–639. Springer, 2014.
- [6] Kostas Patroumpas, Elias Alevizos, Alexander Artikis, Marios Vodas, Nikos Pelekis, and Yannis Theodoridis. Online event recognition from moving vessel trajectories. *GeoInformatica*, 21(2):389–427, 2017.
- [7] Ehab Qadah, Michael Mock, Elias Alevizos, and Georg Fuchs. A Distributed Online Learning Approach for Pattern Prediction over Movement Event Streams with Apache Flink. In *EDBT/ICDT Workshops*, 2018.

QUESTIONS?

Thank you!