

# A Distributed Online Learning Approach for Large-scale Pattern Prediction (WP3)

Michael Mock \*    Ehab Qadah \*

\*Fraunhofer IAIS, Germany

Sunday 14<sup>th</sup> January, 2018

# Outline

datAcron Architecture

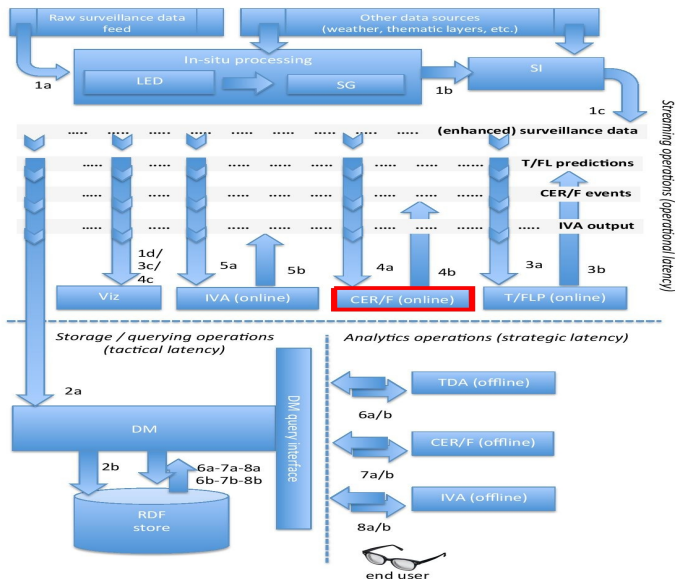
Proposed Approach

Event Forecasting with Pattern Markov Chains

Communication-Efficient Distributed Online Prediction by  
Dynamic Model Synchronization

Evaluation

# datAcron Architecture



# Problem Setting

- ▶ Given a set of  $k$  real-time streams of events  
 $S = \{s_1, s_2, \dots, s_k\}$ .
- ▶ Each stream  $s_i = \langle e_1, e_2, \dots, e_t, \dots \rangle$  is an evolving time-ordered sequence of events.
- ▶ Each event is defined as a tuple of attributes  
 $e_t = (\text{type}, \tau, \text{id}, a_1, a_2, \dots, a_n)$  where  $\text{type} \in \Sigma$  (i.e., event types).
- ▶ A user-defined pattern (i.e., complex event of interest)  $P$  expressed as sequence of event types.
- ▶ Goal: the main objective is to predict the pattern  $P$  completion with certain probability in the future over each stream  $s_i$  given the current time event  $e_t$ .

# Maritime Surveillance

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

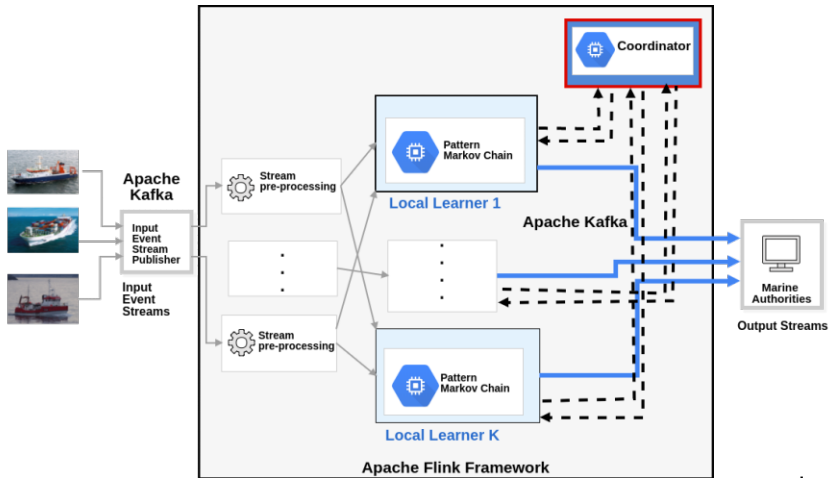
```
{  
  "timestamp":1443651492000,  
  "id":"228133000",  
  "annotation":"change_in_heading",  
  "latitude":48.117775,  
  "longitude":-4.4205885,  
  "distance":323.406,  
  "heading":264.27  
  "speed":18.48,  
}
```

- ▶ Example patterns such as

$P_1 = \text{change\_heading} \cdot \text{gap\_start} \cdot \text{gap\_end} \cdot \text{change\_heading}$   
or  $P_2 = \text{Sailing}$

# A Distributed Online Learning Approach for Large-scale Pattern Prediction <sup>1</sup>

Distributed Architecture (Qadah, E., Mock, M., Alevizos, E. & Fuchs, G. (n.d.))



<sup>1</sup>Source Code: <https://goo.gl/xwX1Mk>

# Event Forecasting with Pattern Markov Chains

(Alevizos et al. 2017)

- ▶ The system consumes a single input stream  $s_i$  of events.
- ▶ The event stream  $s_i$  is assumed to be generated by  $m$ -order Markov source.
- ▶ The complex event (i.e., pattern)  $P$  is defined in the form of regular expressions over a finite set of event types  $\Sigma$ .
- ▶ A probabilistic model provides online forecasting reports when the event pattern  $P$  is expected to be completed in future.

# Event Forecasting with Pattern Markov Chains

How does it work?

- ▶ The deterministic finite automata (*DFA*) is used to construct a Markov chain, which is called a Pattern Markov Chain (*PMC*).
- ▶ The states of *DFA* is directly mapped to states of transition probability matrix  $\mathbf{M} \mid Q \mid \times \mid Q \mid$  of the *PMC*.

▶

$$\mathbf{M} = \begin{Bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix} \begin{pmatrix} p_{0,0} & . & . & . & p_{0,4} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & . & p_{4,4} \end{pmatrix}$$

- ▶ The maximum-likelihood estimator is used to compute the transition probabilities (i.e., learning)  $p_{i,j}$  of the matrix  $\mathbf{M}$

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



# Communication-Efficient Distributed Online Prediction by Dynamic Model Synchronization

(Kamp et al. 2014)

- ▶ 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) (Dekel et al. 2012).
- ▶ A dynamic synchronization scheme based on monitoring the local models variance from a global reference model ( $\|f_i - r\|^2 \leq \Delta$ ).

# Communication-Efficient Distributed Online Prediction by Dynamic Model Synchronization

---

**Algorithm 1:** Communication-efficient Distributed Online Learning Protocol (Kamp et al. 2014)

---

**Predictor** node  $n_i$ : at observing event  $e_j$   
update the prediction model parameters  $f_i$  and provide a prediction service ;  
**if**  $j \bmod b = 0$  **and**  $\|f_i - f_r\|^2 > \Delta$  **then**  
send  $f_i$  to the Coordinator (violation) ;

**Coordinator:**

receive local models with violation  $B = \{f_i\}_{i=1}^m$  ;  
**while**  $|B| \neq k$  **and**  $\frac{1}{|B|} \sum_{f_i \in B} \|f_i - \hat{f}\|^2 > \Delta$  **do**  
| add other nodes have not reported violation for  
| their models  $B \leftarrow \{f_l : f_l \notin B \text{ and } l \in [k]\}$  ;  
| receive models from nodes add to  $B$ ;  
  
compute a new global model  $\hat{f}$  ;  
send  $\hat{f}$  to all the predictors in  $B$  and set  $f_1 \dots f_m = \hat{f}$ ;  
**if**  $|B| = k$  **then**  
| set a new reference model  $f_r \leftarrow \hat{f}$  ;

---

# A Distributed Online Learning Approach for Large-scale Pattern Prediction

## Synchronization Operation

- We propose a *synchronization operation* for the parameters of the models ( $f_i = \mathbf{\Pi}_i : i \in [k]$ ) of the  $k$  distributed PMC predictors. The operation is based on distributing the maximum-likelihood estimation for the transition probabilities of the underlying models described by:

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

# A Distributed Online Learning Approach for Large-scale Pattern Prediction

## Synchronization Operation

- ▶ We propose a *synchronization operation* for the parameters of the models ( $f_i = \Pi_i : i \in [k]$ ) of the  $k$  distributed PMC predictors. The operation is based on distributing the maximum-likelihood estimation for the transition probabilities of the underlying models described by:

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

- ▶ We measure the divergence of local models from the reference model  $\|f_k - f_r\|^2$  by calculating the sum of square difference between the transition probabilities  $\Pi_i$  and  $\Pi_r$ :

$$\|f_k - f_r\|^2 = \sum_{i,j} (\hat{\pi}_{k,i,j} - \hat{\pi}_{r,i,j})^2$$

# Evaluation of Proposed Approach

## Performance Measures

- *Precision* =  $\frac{\# \text{ of correct predictions}}{\# \text{ of total predictions}}$  is the fraction of the produced predictions that are correct (i.e., a full match occurred within the prediction interval).

# Evaluation of Proposed Approach

## Performance Measures

- ▶ *Precision* =  $\frac{\# \text{ of correct predictions}}{\# \text{ of total predictions}}$  is the fraction of the produced predictions that are correct (i.e., a full match occurred within the prediction interval).
- ▶ *Spread* =  $end(I) - start(I)$  is the width of the prediction interval  $I$ .

# Evaluation of Proposed Approach

## Performance Measures

- ▶ *Precision* =  $\frac{\# \text{ of correct predictions}}{\# \text{ of total predictions}}$  is the fraction of the produced predictions that are correct (i.e., a full match occurred within the prediction interval).
- ▶ *Spread* =  $end(I) - start(I)$  is the width of the prediction interval  $I$ .
- ▶ *Cumulative communication*: number of messages, which are required to perform the distributed online learning modes to synchronize the prediction models.

# Empirical evaluation

## Experimental Setup

- ▶ Synopses over raw AIS messages in Brest, France: 1 October 2015 to 31 March 2016 (ais\_brest\_synopses.json).
- ▶ 4,684,444 derived critical points.
- ▶  $\approx 5000$  vessels.



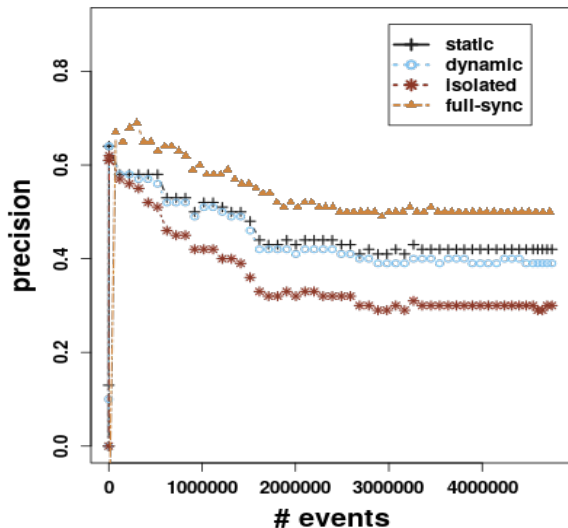
# Empirical evaluation

## Experimental Setup

- ▶ Synopses over raw AIS messages in Brest, France: 1 October 2015 to 31 March 2016 (ais\_brest\_synopses.json).
- ▶ 4,684,444 derived critical points.
- ▶  $\approx 5000$  vessels.
- ▶ Used patterns are:  $\mathcal{P}_1 = \textit{Sailing}$  &  $\mathcal{P}_2 = \textit{changeInHeading}; \textit{gapStart}; \textit{gapEnd}; \textit{changeInHeading}$ .

# Empirical evaluation

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



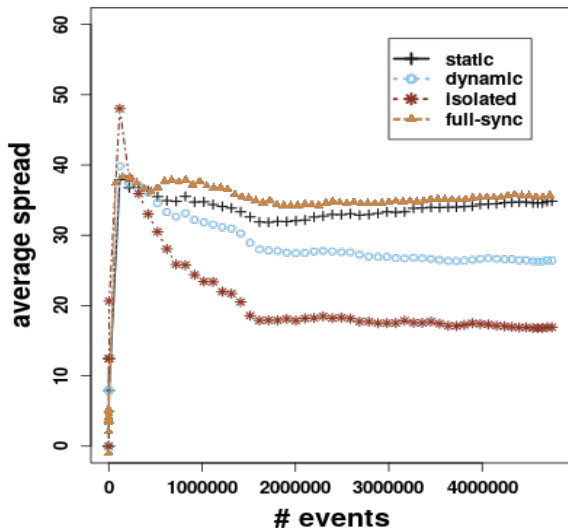
# Empirical evaluation

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

- ▶ All methods of distributed learning outperform the isolated prediction model
- ▶ The hypothetical method of full continuous synchronization has the highest precision rates.
- ▶ The static and dynamic synchronization schemes have close precision scores

# Empirical evaluation

Average spread value for  $\mathcal{P}_1$



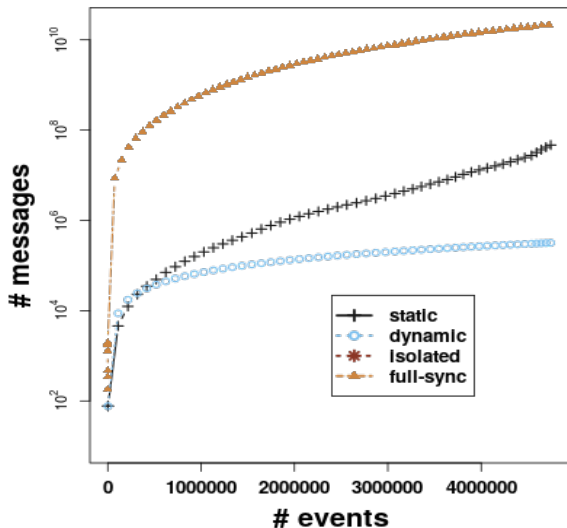
# Empirical evaluation

Average spread value for  $\mathcal{P}_1$

- ▶ The spread is higher for the distributed learning based methods comparing to the isolated approach
- ▶ The average spread is decreasing over time until convergence, which may explain the drop in the precision scores from the beginning until reaching the convergence.

# Empirical evaluation

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



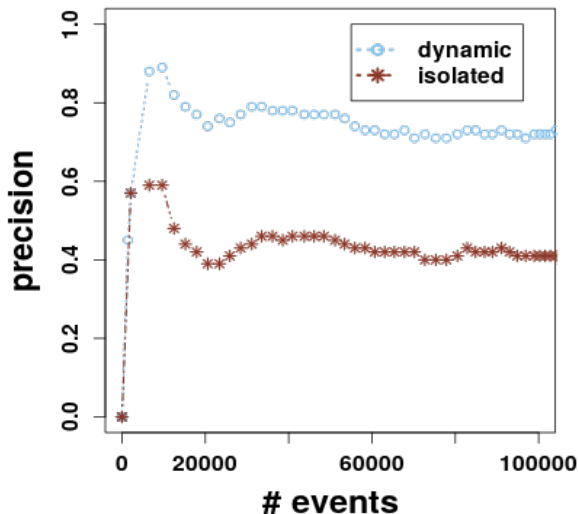
# Empirical evaluation

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

- ▶ A larger amount of communication is required for the continuous synchronization comparing to the static and dynamic approaches
- ▶ We can reduce the communication overhead by applying the dynamic synchronization protocol (a reduction by a factor of 100) comparing to the static synchronization scheme

# Empirical evaluation

Precision scores of  $\mathcal{P}_2$  for *PLEASURE CRAFT* vessels





# Empirical evaluation

Precision scores of  $\mathcal{P}_2$  for *PLEASURE CRAFT* vessels

- ▶ For the second, more complex pattern ( $\mathcal{P}_2$ ), we have found that the precision was worse for a distributed model generated over all vessels than in the model created for each vessel in isolation
- ▶ So we only enabled the synchronization of the prediction models associated with vessels that belong to the same vessel class e.g., *PLEASURE CRAFT*

# Next steps

- ▶ Interface definition with Elias for independent updates
- ▶ Evaluation on synthetic data
- ▶ Analysis of distributed learning effects
- ▶ Performance measurements on datAcron cluster
- ▶ Prediction intervals with time information

# Bibliography

- Alevizos, E., Artikis, A. & Paliouras, G. (2017), Event forecasting with pattern markov chains, *in* 'Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems', ACM, pp. 146–157.
- Dekel, O., Gilad-Bachrach, R., Shamir, O. & Xiao, L. (2012), 'Optimal distributed online prediction using mini-batches', *Journal of Machine Learning Research* **13**(Jan), 165–202.
- Kamp, M., Boley, M., Keren, D., Schuster, A. & Sharfman, I. (2014), Communication-efficient distributed online prediction by dynamic model synchronization, *in* 'Joint European Conference on Machine Learning and Knowledge Discovery in Databases', Springer, pp. 623–639.