SSTRESED: Scalable Semantic Trajectory Extraction for Simple Event Detection over Streaming Movement Data

Nikos Giatrakos
Technical University of Crete, Chania, Greece
Athena RC, Marousi, Greece

Abstract

We describe SSTRESED, a prototype focused on the real-time, online detection of simple, durative events over streaming movement data. It is the first prototype that establishes a direct connection between semantic trajectory extraction and simple event detection. SSTRESED is highly scalable by incorporating parallel processing in two separate, but connected, training and event detection pipelines implemented on state-of-the-art platforms, directly deployable in cloud environments.

2012 ACM Subject Classification Computer systems organization → Architectures

Keywords and phrases Semantic Trajectory, Event Processing, Data Streams

Digital Object Identifier 10.4230/LIPIcs.TIME.2023.15

Category Extended Abstract

Funding This work is funded by the European Union under Horizon Europe agreement No 101070430.

1 Introduction & Motivation

Detecting Simple, Derived Events (SDEs) is the first step towards Complex Event Recognition (CER) [3, 4, 5]. In time critical-applications [1, 6], such as safe robot navigation in dynamic smart factory environments, SDE detection should be performed continuously over voluminous streams of movement data arriving at high speeds. In such scenarios, extracting SDEs out of raw streams is a challenging task engaging (a) online neural network training for continuously maintaining an up-to-date model for SDE labelling purposes and (b) semantic-aware trajectory processing for identifying homogeneous movement portions, defining the SDE duration, before using the neural model for labelling it. By definition, output SDEs are simple pieces of information (Listing 2), but the volume and velocity of the original raw streams (Listing 1) in large scale smart factory applications call for scaling out (parallelizing) the computation to a number of machines to ensure real-time processing. Therefore, both (a) and (b) should be set up in state-of-the-art, relevant platforms [7, 9] to allow for direct deployment over computer clusters and/or the cloud. To tackle these challenges we develop SSTRESED, a prototype for scalable SDE detection over streaming movement data. For the first time, SSTRESED establishes a direct connection between semantic trajectory computation and SDE detection in the streaming context. This is in contrast to prior art [9, 10] which uses predetermined, application-defined time windows to a priori restrict eligible SDE durations.

2 The SSTRESED Prototype

SSTRESED (Figure 1) composes two connected pipelines distributed across worker machines running in the cloud. In the robotic scenario of Section 1, truthful, timestamped and labeled movement streams are continuously produced by robotic simulators, such as `https://github.com/rock-simulation`, as SDEs and their raw features, per robot (Listing 1).

30th International Symposium on Temporal Representation and Reasoning (TIME 2023).
Editors: Alexander Artikis, Florian Bruse, and Luke Hunsberger; Article No. 15; pp. 15:1–15:4
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

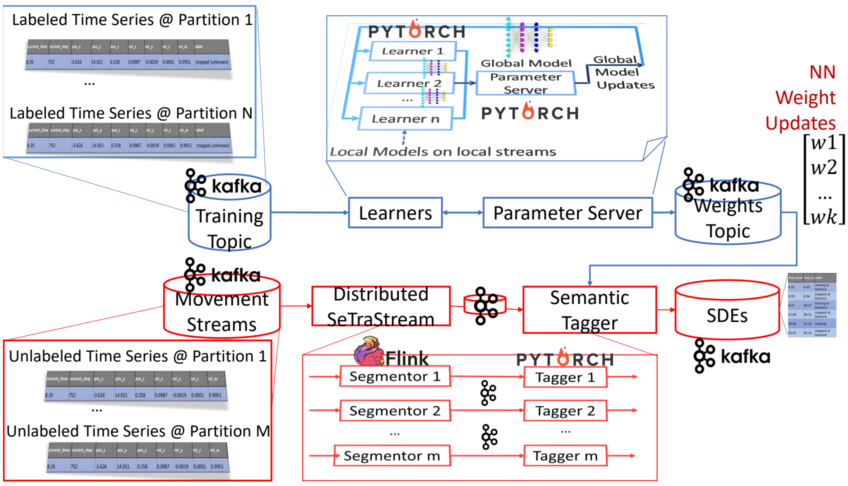

Figure 1 SSTRESED Architecture. Training (blue) and SDE Detection (red) Pipelines.

Listing 1 Example training stream for a single simulated robot. Unlabelled movement streams lack a SDE label (last column in the figure). Thousands of such streams can be ingested by SSTRESED in large scale applications.

```
time    ...   pos_x    pos_y    pos_z   ...   rot_w    SDE
8.35    ...   −3.626   14.921   0.258   ...   0.9951   stopped at Station1
30.57   ...   ...      ...      ...     ...   ...      ...
41.15   ...   −7.446   23.866   0.257   ...   0.0977   moves to Station3
41.12   ...   −7.444   23.867   0.258   ...   0.0972   rotating
```

The training pipeline (blue-colored path in Figure 1) continuously receives these robot movement time series ingested in Apache Kafka partitions of the `Training Topic`. The `Training Topic` is read by parallel PyTorch `Learners`. Each such learner, utilizes an identical neural model (specified by the application), but performs the training process on a separate set of robots. The local models learned at each `Learner i` (top of Figure 1) are synchronized into a global neural model maintained by a `Parameter Server` [2]. At a global model update, new weights of the neural network are written to a `Weights Topic` of Kafka.

The SDE detection pipeline (red-colored path in Figure 1) receives raw, unlabeled streaming movement data, partitioned in the `Movement Streams` Kafka Topic. These incoming tuples, ingested directly from the application field, have the same schema as those of the `Training Topic`, but lack a label/SDE field. Ingested `Movement Streams` of robots (or, optionally, samples of them [8, 11]) are processed by a distributed version of SeTraStream [12] developed in Apache Flink. `Distributed SeTraStream` uses each parallel `Segmentor i` to continuously identify homogeneous movement portions based on the ingested features per robot, thus semantically and temporally segmenting each trajectory. In that, the duration of a SDE is determined, which also bounds the feature tensors that should then be used for labeling the SDE. Each parallel `Segmentor i` writes the result of its processing to an intermediate Kafka topic connecting `Distributed SeTraStream` with a PyTorch `Semantic Tagger` in the red-colored path. Each parallel `Tagger i` (bottom of Figure 1) of the `Semantic Tagger`, at any given time instance, reads the up-to-date weights from the `Weights Topic` and uses the updated neural model to label SDEs. The final SSTRESED output goes to the `SDEs` Kafka topic in the form of tuples as illustrated in Listing 2 (per robot).

Listing 2 SSTRESED output SDE Stream for the movement of a single robot.

```
Time_from          Time_to          SDE
4.25                8.35              moving to Station2
8.35                8.36              stopped at Station2
...                 ...               ...
39.00               41.15             rotating
```

References

1 Alexander Artikis, Nikos Katzouris, Ivo Correia, Chris Baber, Natan Morar, Inna Skarbovsky, Fabiana Fournier, and Georgios Paliouras. A prototype for credit card fraud management: Industry paper. In *Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems, DEBS 2017, Barcelona, Spain, June 19-23, 2017*, pages 249–260. ACM, 2017.

2 Antonios Deligiannakis, Nikos Giatrakos, Yannis Kotidis, Vasilis Samoladas, and Alkis Simitsis. Extreme-scale interactive cross-platform streaming analytics - the INFORE approach. In *Proceedings of the 2nd Workshop on Search, Exploration, and Analysis in Heterogeneous Datastores (SEA-Data 2021) co-located with 47th International Conference on Very Large Data Bases (VLDB 2021), Copenhagen, Denmark, August 20, 2021*, volume 2929 of *CEUR Workshop Proceedings*, pages 7–13. CEUR-WS.org, 2021.

3 Ioannis Flouris, Nikos Giatrakos, Antonios Deligiannakis, and Minos N. Garofalakis. Network-wide complex event processing over geographically distributed data sources. *Inf. Syst.*, 88, 2020.

4 Ioannis Flouris, Nikos Giatrakos, Minos N. Garofalakis, and Antonios Deligiannakis. Issues in complex event processing systems. In *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 2*, pages 241–246. IEEE, 2015.

5 Nikos Giatrakos, Elias Alevizos, Alexander Artikis, Antonios Deligiannakis, and Minos N. Garofalakis. Complex event recognition in the big data era: a survey. *VLDB J.*, 29(1):313–352, 2020.

6 Nikos Giatrakos, Antonios Deligiannakis, Konstantina Bereta, Marios Vodas, Dimitris Zissis, Elias Alevizos, Charilaos Akasiadis, and Alexander Artikis. Processing big data in motion: Core components and system architectures with applications to the maritime domain. In Edward Curry, Sören Auer, Arne J. Berre, Andreas Metzger, María S. Pérez, and Sonja Zillner, editors, *Technologies and Applications for Big Data Value*, pages 497–518. Springer, 2022. doi:10.1007/978-3-030-78307-5_22.

7 Nikos Giatrakos, Eleni Kougioumtzi, Antonios Kontaxakis, Antonios Deligiannakis, and Yannis Kotidis. Easyflinkcep: Big event data analytics for everyone. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3029–3033. ACM, 2021.

8 Antonis Kontaxakis, Nikos Giatrakos, and Antonios Deligiannakis. A synopses data engine for interactive extreme-scale analytics. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2085–2088. ACM, 2020.

9 Emmanouil Ntoulias, Elias Alevizos, Alexander Artikis, Charilaos Akasiadis, and Athanasios Koumparos. Online fleet monitoring with scalable event recognition and forecasting. *GeoInformatica*, 26(4):613–644, 2022.

10 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.

11 Marios Vodas, Konstantina Bereta, Dimitris Kladis, Dimitris Zissis, Elias Alevizos, Emmanouil Ntoulias, Alexander Artikis, Antonios Deligiannakis, Antonios Kontaxakis, Nikos Giatrakos, David Arnu, Edwin Yaqub, Fabian Temme, Mate Torok, and Ralf Klinkenberg. Online distributed maritime event detection & forecasting over big vessel tracking data. In *2021*

IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, December 15-18, 2021, pages 2052–2057. IEEE, 2021.

12 Zhixian Yan, Nikos Giatrakos, Vangelis Katsikaros, Nikos Pelekis, and Yannis Theodoridis. Setrastream: Semantic-aware trajectory construction over streaming movement data. In *Advances in Spatial and Temporal Databases - 12th International Symposium, SSTD 2011, Minneapolis, MN, USA, August 24-26, 2011, Proceedings*, volume 6849 of *Lecture Notes in Computer Science*, pages 367–385. Springer, 2011.