

Online Detection of Anomalous Sub-Trajectories - A Sliding Window Approach based on Conformal Anomaly Detection and Local Outlier Factor

Rikard Laxhammar^{1,2} and Göran Falkman¹

¹ University of Skövde, Sweden

² Saab Security and Defence Solutions, Järfälla, Sweden
firstname.lastname@his.se

Abstract Automated detection of anomalous trajectories is an important problem in the surveillance domain. Various algorithms based on learning of normal trajectory patterns have been proposed for this problem. Yet, these algorithms suffer from one or more of the following limitations: First, they are essentially designed for offline anomaly detection in databases. Second, they are insensitive to local sub-trajectory anomalies. Third, they involve tuning of many parameters and may suffer from high false alarm rates. The main contribution of this paper is the proposal and discussion of the Sliding Window Local Outlier Conformal Anomaly Detector (SWLO-CAD), which is an algorithm for online detection of local sub-trajectory anomalies. It is an instance of the previously proposed Conformal anomaly detector and, hence, operates online with well-calibrated false alarm rate. Moreover, SWLO-CAD is based on Local outlier factor, which is a previously proposed outlier measure that is sensitive to local anomalies. Thus, SWLO-CAD has a unique set of properties that address the issues above.

1 Introduction

Anomalous behaviour may indicate objects or events of interest in a wide variety of domains. One such domain is surveillance, where there is a clear trend towards more and more advanced sensor systems producing huge amounts of *trajectory data* from moving objects, such as people, vehicles, vessels and aircraft. In the maritime domain, for example, anomalous vessel behaviour, such as unexpected stops, deviations from standard routes, speeding, traffic direction violations etc., may indicate threats and dangers related to smuggling, sea drunkenness, collisions, grounding, hijacking, piracy etc. Timely detection of these relatively infrequent events, which is critical for enabling pro-active measures, requires constant analysis of all trajectories. This is typically a great challenge to human analysts due to information overload, fatigue and inattention.

This paper is concerned with algorithms for automated detection of anomalous trajectories. A large number of algorithms based on learning of normal trajectory patterns have been proposed for this problem [11]. However, these

algorithms typically suffer from one or more of the following limitations: First, they are essentially designed for offline anomaly detection in databases, i.e. they are not appropriate for *online anomaly detection* where trajectories are processed sequentially in the order they are observed. Second, they are insensitive to *local sub-trajectory anomalies*. Third, they are *parameter-laden* [7] and their performance may suffer from *high false alarm rates* [1].

Motivated by the above limitations, we propose and discuss a new algorithm for online detection of anomalous sub-trajectories, known as the Sliding Window Local Outlier Conformal Anomaly Detector (SWLO-CAD). This algorithm is an instance of the previously proposed Conformal Anomaly Detector (CAD) [9], which is a general and parameter-light algorithm for online anomaly detection with well-calibrated false alarm rate. The key idea in CAD is to estimate *p-values* for new data based on a *Non-Conformity Measure* (NCM) as defined in the *Conformal prediction* (CP) framework [4]. The NCM is the only design parameter of CAD. In this paper, we propose a novel NCM for trajectory data, known as the Sliding Window Local Outlier Trajectory Non-conformity Measure (SWLO-TNM). This NCM is based on Local Outlier Factor (LOF) [2] and is therefore sensitive to local sub-trajectory anomalies.

To summarise, the contribution of this paper concerns the proposal and discussion of SWLO-CAD for online detection of local sub-trajectory anomalies. In particular, the main contribution is the proposal of SWLO-TNM, which is a novel NCM with unique properties, designed for trajectory data. The outline of the remaining part of this paper is as follows: In Section 2, we first discuss in more detail limitations of previous algorithms for anomaly detection in trajectories. This is followed by a review of CAD and LOF, since they underpin SWLO-CAD and SWLO-TNM, respectively. In Section 3, we will first define more precisely what we mean by trajectories and what properties that should be fulfilled by a NCM for detecting local sub-trajectory anomalies. We will then present SWLO-NCM and discuss its theoretical properties and how it can be implemented. The section will be concluded with a brief discussion of limitations and future work. Finally, Section 4 will present the conclusions of the paper.

2 Background

2.1 Limitations of Previous Algorithms for Trajectory Anomaly Detection

Previous algorithms are typically designed for offline anomaly detection in trajectory databases in a one-step or two-step fashion, depending on whether learning is unsupervised or supervised, respectively. In the supervised case, a one-class model is first learnt from a batch of training trajectories labelled normal. This normalcy model is then used repeatedly for classifying new trajectories in a test set. In the unsupervised case, all data is assumed to be unlabelled and learning and anomaly detection are blended, i.e. the training set and the test set are equivalent. But regardless of whether learning is supervised or unsupervised, a fairly large training (test) set that is representative of all possible normal behaviour

may be required in order to accurately detect new anomalous trajectories. In some applications, such a data set might not be available from the outset. Yet, it may still be desirable or necessary to be able to classify each new trajectory based on the training set accumulated so far. Moreover, “in many domains normal behaviour keeps evolving and a current notion of normal behaviour might not be sufficiently representative in the future” [3]. The need for algorithms capable of efficient online anomaly detection in trajectories, i.e. when trajectories are analysed in sequence and model parameters updated incrementally, was highlighted by Piciarelli and Foresti [12].

Another limitation of previously proposed algorithms is their insensitivity to local anomalous sub-trajectories. To illustrate this point, consider Figure 1 which highlights two trajectories (Tr_1 and Tr_2) that are anomalous at different scales. The first trajectory (Tr_1) corresponds to a global anomaly that is anomalous in

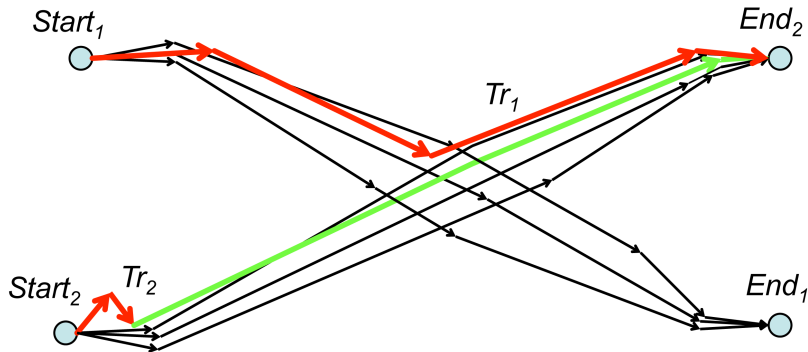


Figure 1. Illustration of a global trajectory anomaly (Tr_1), which is all red, vs. a local sub-trajectory anomaly (Tr_2), where red and green correspond to the anomalous and normal part of the trajectory, respectively.

its entirety, since trajectories originating from $Start_1$ are expected to go to End_1 rather than End_2 . Such anomalies are typically detectable by previously proposed algorithms for trajectory anomaly detection. The second trajectory (Tr_2) corresponds to a local anomaly: while the overall route of Tr_2 is consistent with other trajectories, the initial sub-trajectory (red) deviates significantly from the other local sub-trajectories in the area. Yet, most trajectory-based algorithms would probably miss this anomaly, since the distance between the anomalous sub-trajectory and its nearby sub-trajectories is less than the distance between neighbouring sub-trajectories in other less dense areas, such as the central crossing area in Figure 1. The underlying models and similarity measures of previous algorithms, including our previously proposed algorithm for sequential anomaly detection in incomplete trajectories [9], essentially consider trajectories *as a whole* and do not consider local variations in density. Therefore, local anomalous sub-trajectories of the type in Figure 1 are “smoothed out”. In fact, this limit-

Algorithm 1 The Conformal Anomaly Detector (CAD)

Input: NCM A , anomaly threshold ϵ , old examples z_1, \dots, z_{n-1} and new example z_n .

Output: Boolean variable *Anomaly*.

```
1:  $D = \{z_1, \dots, z_n\}$ 
2: for  $i \leftarrow 1$  to  $n$  do
3:    $\alpha_i \leftarrow A(D \setminus z_i, z_i)$ 
4: end for
5:  $\tau \leftarrow U(0, 1)$ 
6:  $p_n \leftarrow \frac{|\{i: \alpha_i > \alpha_n\}| + \tau |\{i: \alpha_i = \alpha_n\}|}{n}$ 
7: if  $p_n < \epsilon$  then
8:   Anomaly  $\leftarrow$  true
9: else
10:  Anomaly  $\leftarrow$  false
11: end if
```

ation was previously identified by Lee et al. [10] who argued that “a trajectory may have a long and complicated path” and that “even though some portions of a trajectory show an unusual behaviour, these differences might be averaged out over the whole trajectory”. Motivated by this problem, they proposed an algorithm for detecting anomalous sub-trajectories, which considers the local data density [10]. Yet, similar to most other algorithms, their algorithm is designed for offline detection of anomalies in a trajectory database; they do not discuss the possibility for online anomaly detection.

2.2 Conformal Anomaly Detection

The Conformal Anomaly Detector (CAD) (Algorithm 1) is a parameter-light algorithm for online anomaly detection with well-calibrate false alarm rate [9]. It is based on estimating p -values for new data based on a NCM as defined in the CP framework [4]: Assume that we are observing a stream of examples corresponding to observed behaviour in some domain of interest. For each new example, z_n , CAD estimates its *smoothed* p -value, $p_n \in (0, 1)$, relative the training set of previously observed examples, $\{z_1, \dots, z_{n-1}\}$. This is done by first calculating the *non-conformity score*, $\alpha_i \in \mathbb{R}$, for each example, $z_i : i = 1, \dots, n$, based on the specified NCM, $A(\{z_1, \dots, z_n\} \setminus z_i, z_i) \rightarrow \alpha_i$ (lines 2–4 of Algorithm 1). Next, p_n is estimated as the sum of 1) the ratio of examples that have a nonconformity score *larger* than α_n and 2) the ratio of examples that have a nonconformity score *equal* to α_n multiplied by a random number, τ , from the unit interval (lines 5–6). If p_n is below the specified anomaly threshold, $\epsilon \in (0, 1)$, z_n is classified as a *conformal anomaly* at significance level ϵ .

The definition of a conformal anomaly is consistent with the statistical definition of an *outlier* given by Hawkins [5]. That is, a conformal anomaly corresponds to an example that deviates so much from training data as to arouse suspicion that it was generated by a different stochastic process. The value p_n can be interpreted as the probability of erroneously rejecting the null hypothesis

that z_n was independently generated from the same distribution as the previous examples [13]. In other words, p_n corresponds to the probability of classifying z_n as anomalous when it is in fact not; this is known as a *false alarm*. In case of online anomaly detection with an accumulating training set of independent and identically distributed examples corresponding to normalcy, the *false alarm rate*, i.e., the frequency of normal examples erroneously classified as anomalous, is expected to be close to ϵ over time. This property of CAD is referred to as *well-calibrated false alarm rate* [9]. The parameter ϵ regulates the sensitivity to true anomalies and should be set depending on the rate of false alarms that is acceptable in the current application. A higher value of ϵ increases probability of detecting true anomalies but also the frequency of false alarms.

Ideally, the true label for each new example would be revealed to CAD after classification, i.e. online supervised anomaly detection. In applications such as surveillance, this would imply that a human has to validate the classification of each new example; obviously, this is not feasible since the purpose of the anomaly detector is to reduce human workload. On the other hand, a fully unsupervised approach, where either all examples or only those examples classified as normal are added to the training set, will likely result in decreased classification performance due to a corrupt or biased training set. A compromise between supervised and unsupervised learning is to adopt a *semi-supervised* learning strategy where only a subset of the examples are labelled. For example, the true label may be provided by a human only for those examples classified as anomalous by CAD, and all new examples *except* those that are detected and confirmed to be anomalous may be added to the training set [9].

Analogously to supervised CP [4], the choice of NCM is of central importance since it determines the character of the detectable anomalies. A poorly chosen NCM may result in that subtle, yet important, anomalies may not be detected. In our previous work, we investigated different NCMs based on the k -nearest neighbour algorithm [8,9]. Advantages of nearest neighbour methods include that they are relatively easy to implement, involve few parameters and are well-suited for online learning since they do not require extensive model update as new training data is added.

2.3 Local Outlier Factor

Local Outlier Factor (LOF) [2] is a powerful density-based outlier measure that has shown good results in wide range of anomaly detection applications [3]. The main advantage of LOF is that it does not suffer from the local density problem [6] as illustrated in Figure 2: Assume that C_1 and C_2 are two clusters with low and high densities, respectively, and that x_1 and x_2 are two anomalous examples located in low and high density areas, respectively. Clearly, x_1 is easily detected by, e.g., a standard nearest-neighbour method, since it has the largest nearest-neighbour distance among *all* examples. However, this is not true for x_2 , since its nearest-neighbour distance is approximately equal or less than most of the normal examples belonging to C_1 . LOF addresses this problem by considering the neighbourhood density of x_2 in relation to the neighbourhood densities of

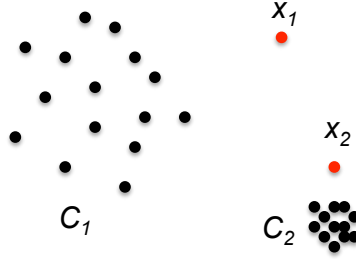


Figure 2. Illustration of the local density problem where C_1 and C_2 correspond to two clusters with different densities, and x_1 and x_2 correspond to anomalous data points in low and high density areas, respectively.

the k -nearest neighbours to x_2 . Following Janssens and Postma [6], we describe the three steps involved when calculating LOF for each example $x_i \in D$ of a dataset D :

1. The first step involves determining the *neighbourhood* of each example: Assume that *neighbourhood border distance* of x_i is defined as:

$$d_{border}(x_i, k) = d(x_i, \text{NN}(x_i, k)), \quad (1)$$

where d is a distance function and $\text{NN}(x_i, k)$ corresponds to the k -nearest neighbour to x_i . Then, the neighbourhood of x_i is constructed as:

$$\mathcal{N}(x_i, k) = \{x_j \in D \setminus \{x_i\} : d(x_i, x_j) \leq d_{border}(x_i, k)\}. \quad (2)$$

2. The second step involves estimating the *neighbourhood density* of each example based on the concept *reachability distance*: Assume that the reachability distance from x_i to x_j is given by:

$$d_{reach}(x_i, x_j, k) = \max\{d_{border}(x_j, k), d(x_j, x_i)\}. \quad (3)$$

Then, the neighbourhood density of x_i is defined as:

$$\rho(x_i, k) = \frac{|\mathcal{N}(x_i, k)|}{\sum_{x_j \in \mathcal{N}(x_i, k)} d_{reach}(x_i, x_j, k)}. \quad (4)$$

The use of d_{reach} instead of d in Eq. 4 has a smoothing effect on the neighbourhood density, which is illustrated in Figure 3: the larger the value of k , the larger is the smoothing effect.

3. In the third and final step, the local outlier factor for each example is calculated by comparing its neighbourhood density with the neighbourhood densities of its neighbouring examples:

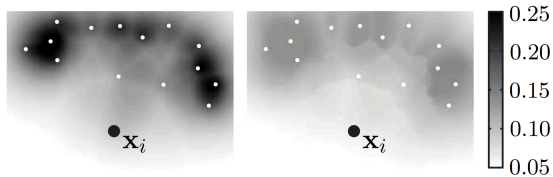


Figure 3. The use of the reachability distance d_{reach} has a smoothing effect. Density plots without (left) and with (right) reachability distance where $k = 3$. The shades of grey represent the density as indicated by the bar on the right. Reproduced from [6] with permission.

$$\text{LOF}(x_i, k) = \frac{\sum_{x_j \in \mathcal{N}(x_i, k)} \rho(x_j, k)}{\rho(x_i, k) |\mathcal{N}(x_i, k)|}. \quad (5)$$

Generally, an example which lies deep inside a cluster will have a local outlier factor close to 1, while an example that lies far from a cluster will have relatively high local outlier factor [2].

3 Sliding Window Local Outlier Conformal Anomaly Detector

This section describes the details of SWLO-CAD, which is proposed for online detection of anomalous sub-trajectories. We start by defining more precisely what trajectories are and discuss the requirements of a NCM aimed for detecting anomalous sub-trajectories in the CAD framework. Next, SWLO-TNM is described, which in combination with CAD constitutes SWLO-CAD. The section is concluded with discussion regarding theoretical properties of SWLO-TNM, how it can be implemented and future work.

3.1 Trajectory Representation

Generally, a trajectory of an object moving in the plane can be considered as a continuous curve in 3D, where the third dimension corresponds to time. In this paper, however, we will consider a simplified model where a trajectory is represented as a continuous curve in 2D:

Definition 1. *A trajectory is a continuous parameterised curve in 2D:*

$$\text{Tr}(t) \rightarrow \langle x, y \rangle : t \in [0, L],$$

where x and y correspond to the spatial location in the plane and L corresponds to the arc length of the curve.

Note that while this representation does not capture the *speed* at which the object moves, it does impose an ordering of different locations along the curve

and, thus, captures the *direction* of travel. Similar representations have previously been proposed by other authors and have shown to be useful for detecting trajectory anomalies [11,9]. Following Definition 1, let us formalise the concept of a sub-trajectory:

Definition 2. *A sub-trajectory of $Tr(t)$ is defined as:*

$$Tr^*(t, s, w) \rightarrow \langle x, y \rangle : t \in [s, s + w], 0 \leq s \leq L - w,$$

where s and w correspond to the start point and the arc length, respectively, of the sub-trajectory and L is the length of the complete trajectory $Tr(t)$.

For the remainder of this paper we will, for simplicity, drop the parameters t, s, w from the notation whenever they are not explicitly needed.

3.2 A Sliding Window Non-conformity Measure Based on Local Outlier Factor

Now, adopting the framework of CAD where each example z corresponds to a trajectory, we are interested in a NCM, $A(Tr, \mathbf{Tr})$, that calculates the nonconformity score for a possibly incomplete trajectory Tr relative a set of trajectories $\mathbf{Tr} \in \{Tr_1, \dots, Tr_n\}$. In addition to accurately discriminating anomalous trajectories from normal trajectories, we argue that the NCM should have the following properties:

1. It should be sensitive to local features of the trajectory at different scales. That is, it should be able to discriminate anomalous sub-trajectories that may be smoothed out if the trajectory is considered as a whole (*cf.* [10]).
2. It should consider the local density and variance of the neighbourhood of sub-trajectories. For example, it should account for that the distance between sub-trajectories in some areas, e.g. harbours, is relatively small compared to other less trafficked areas.
3. As discussed in our previous work [9], the NCM should support *sequential anomaly detection* in order to reduce detection delay. In particular, it should support calculation of an intermittent nonconformity score that monotonically increases as the trajectory extends, i.e. $A(Tr^*(t, s, w), \mathbf{Tr}) \leq A(Tr^*(t, s, w'), \mathbf{Tr})$ if $w \leq w'$. This property ensures that the probability of false alarm for CAD will still be equal to ϵ during sequential anomaly detection in incomplete trajectories.

We argue that SWLOF-TNM addresses all of the above properties. But before we present SWLOF-TNM, let us introduce some key concepts:

Definition 3. *Let $subdist \langle Tr_i^*(s, w), Tr_j^*(s', w) \rangle \in \mathbb{R}^+$ be a distance measure for comparing two sub-trajectories, $Tr_i^*(s, w)$ and $Tr_j^*(s', w)$, having the same arc length w .*

One such distance measure is the *maximum of the squared Euclidean distance* between each pair of corresponding points along the two sub-trajectories:

$$\begin{aligned} & \text{subdist}_{\max\text{SED}} \langle Tr_i^*(s, w), Tr_j^*(s', w) \rangle = \\ & \max_{\tau \in [0, w]} \left\{ \langle x_i(s + \tau) - x_j(s' + \tau) \rangle^2 + \langle y_i(s + \tau) - y_j(s' + \tau) \rangle^2 \right\}, \end{aligned} \quad (6)$$

where $x(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$ are the continuous parameterisations of the x and y components of the corresponding trajectory.

Definition 4. *Let:*

$$\begin{aligned} & \text{align}(Tr_i^*(s, w), Tr_j) \rightarrow Tr_j^*(s', w) : \\ & \arg \min_{s' \in [0, L-w]} \left\{ \text{subdist} \langle Tr_i^*(s, w), Tr_j^*(s', w) \rangle \right\}, \end{aligned} \quad (7)$$

be an alignment function that returns the sub-trajectory $Tr_j^*(s', w)$ from Tr_j that minimises the distance to $Tr_i^*(s, w)$ according to *subdist*.

Now, let us go on and re-define the LOF-concepts from Section 2.3 where each example x now corresponds to a sub-trajectory:

- $\text{NN}(Tr_i^*, k) \in \{\text{align}(Tr_i^*, Tr_j) : Tr_j \in \mathbf{Tr}\}$ as the k -nearest neighbour to Tr_i^* among the aligned sub-trajectories, one for each trajectory in \mathbf{Tr} .
- $d_{\text{border}}(Tr_i^*, k) = \text{subdist} \langle Tr_i^*, \text{NN}(Tr_i^*, k) \rangle$ as the neighbourhood border distance of Tr_i^* .
- $\mathcal{N}(Tr_i^*, k) \subseteq \{\text{align}(Tr_i^*, Tr_j) : Tr_j \in \mathbf{Tr}\}$ as the corresponding k -neighbourhood of Tr_i^* , i.e. the set of aligned sub-trajectories that are at distance at most $d_{\text{border}}(Tr_i^*, k)$ from Tr_i^* .
- $d_{\text{reach}}(Tr_i^*, Tr_j^*, k) = \max \{d_{\text{border}}(Tr_i^*, k), \text{subdist} \langle Tr_j^*, Tr_i^* \rangle\}$ as the reachability distance from Tr_j^* to Tr_i^* .
- $\rho(Tr_i^*, k) = |\mathcal{N}(Tr_i^*, k)| / \sum_{Tr_j^* \in \mathcal{N}(Tr_i^*, k)} d_{\text{reach}}(Tr_i^*, Tr_j^*, k)$ as the neighbourhood density of Tr_i^* .

Based on LOF-concepts above, we define the Sub-Trajectory Local Outlier Factor (ST-LOF) as:

$$\text{ST-LOF}(Tr_i^*, k) = \frac{\sum_{Tr_j^* \in \mathcal{N}(Tr_i^*, k)} \frac{\rho(Tr_j^*, k)}{\rho(Tr_i^*, k)}}{|\mathcal{N}(Tr_i^*, k)|}. \quad (8)$$

Finally, we are ready to introduce the Sliding Window Local Outlier Trajectory Nonconformity Measure (SWLO-TNM):

$$\text{SWLO-TNM} \{Tr_i, \mathbf{Tr}, w, k\} = \max_{s \in [0, L_i - w]} \{\text{ST-LOF}(Tr_i^*(s, w), k)\}, \quad (9)$$

where L_i corresponds to the length of Tr_i .

3.3 Discussion

SWLO-TNM considers all sub-trajectories of length w and will assign a relatively high non-conformity score to Tr_i whenever any of its sub-trajectories has a sufficiently high outlier score, i.e. regardless of whether the remaining part of the trajectory is perfectly normal. The parameter w corresponds to the scale/resolution of the anomalous sub-trajectories and, depending on the application, it may be appropriate to run multiple anomaly detectors in parallel using different values of w . Hence, SWLO-TNM meets requirement 1 in Section 3.2. Furthermore, by considering the local density of each sub-trajectory using ST-LOF, SWLO-TNM also meets requirement 2. Finally, requirement 3 is also met: SWLO-TNM is based on the maximum of the outlier scores of all sub-trajectories and, hence, it monotonically increases as the trajectory extends. In principle, by plugging SWLO-NCM into CAD we get SWLO-CAD, which is a parameter-light algorithm for online detection of local sub-trajectory anomalies with well-calibrated false alarm rate.

In practice, a family of parameterised curves need to be chosen that is able to approximate the true trajectories fairly well, while still allowing for efficient calculation of SWLO-TNM. Assuming that we are monitoring vessel trajectories, it makes sense to adopt a *polyline* representation since vessel traffic typically follow straight line segments, such as sea lanes. That is, each trajectory would be represented by m vertices where $t_1 = 0$ and $t_m = L$ correspond the first and last point of the trajectory, respectively, and where t_2, \dots, t_{m-1} correspond to the intermediate vertices, i.e. the end points of the line segments. The polyline representation is practical from a computational perspective and allows for efficient compression using a line simplification algorithm. Yet, implementing an algorithm for exact calculation of SWLO-TNM seems difficult, if not impossible. More specifically, finding the optimal values for s in Eq. 9 and each s' in Eq. 7 is not trivial. Hence, it may be appropriate to discretize the range of τ , s' and s and perform a linear parameter search during the minimisation and maximisation in Eq. 6, 7 and 9. Moreover, an efficient indexing of sub-trajectories may be appropriate in order to suppress computational complexity.

Because of limited space, this paper does not cover any empirical investigations of SWLO-CAD. However, we have implemented SWLO-CAD and conducted preliminary experiments in the maritime domain. These experiments show promising results and an obvious direction for future work is to investigate classification performance on labelled data sets with anomalous sub-trajectories. Moreover, we observe that SWLO-TNM could potentially may applied as NCM for supervised classification of trajectories in the standard CP framework.

4 Conclusion

Previous algorithms for anomaly detection in trajectories typically suffer from one or more of the following issues: First, they are essentially designed for off-line anomaly detection in databases. Second, they are insensitive to local sub-trajectory anomalies. Third, they are more or less parameter-laden and may have

high false alarm rates. In this paper, we have proposed and discussed SWLO-CAD for online detection of local sub-trajectory anomalies. SWLO-CAD has a unique set of properties that follow from the fact that it is based on CAD and LOF. In particular, SWLO-CAD is parameter-light, operates online, has well-calibrated false alarm rate and is sensitive to local anomalies. Hence, it addresses the limitations above. We have also discussed issues related to the implementation of SWLO-CAD in practice. Future work includes empirical investigations on trajectory data sets that include anomalous sub-trajectories.

Acknowledgments

The authors would like to thank Klas Wallenius and Egils Sviestins at Saab Security and Defence Solutions for valuable feedback regarding the manuscript of this paper. Moreover, the authors would like to acknowledge Mattias Björkman at Saab for interesting discussions regarding the topic of this paper.

References

1. S. Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. on Inform. and Syst. Security*, 3(3), Aug 2000.
2. M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proc ACM SIGMOD Int. Conf. Management of Data*, 2000.
3. Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):1–58, 2009.
4. Alexander Gammerman and Vladimir Vovk. Hedging predictions in machine learning. *Computer Journal*, 50(2):151–163, 2007.
5. D. Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.
6. J. Janssens and E. Postma. One-class classification with lof and loci: An empirical comparison. In *Proc. 18th Belgian-Dutch Conf. on Mach. Learn.*, 2009.
7. Eamonn Keogh, Stefano Lonardi, Chotirat Ratanamahatana, Li Wei, Sang-Hee Lee, and John Handley. Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14:99–129, 2007.
8. R. Laxhammar and G. Falkman. Conformal prediction for distribution-independent anomaly detection in streaming vessel data. In *Proc. 1st Int. Worksh. Novel Data Stream Pattern Mining Techniques*. ACM, 2010.
9. R. Laxhammar and G. Falkman. Sequential conformal anomaly detection in trajectories based on hausdorff distance. In *Proc. 14th Int. Conf. Information Fusion*, 2011.
10. J-G Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *Proc. 24th Int. Conf. Data Engineering*, 2008.
11. B.T. Morris and M.M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Trans. Circuits Syst. for Video Technol.*, 18:1114–1127, August 2008.
12. C. Piciarelli and G.L. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, 27, 2006.
13. Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *J. Mach. Learn. Res.*, 9:371–421, 2008.