

# A Comparative Evaluation of Anomaly Detection Algorithms for Maritime Video Surveillance

Bryan Auslander<sup>1</sup>, Kalyan Moy Gupta<sup>1</sup>, and David W. Aha<sup>2</sup>

<sup>1</sup>Knexus Research Corporation; Springfield, VA 22153

<sup>2</sup>Navy Center for Applied Research in Artificial Intelligence;  
Naval Research Laboratory (Code 5514); Washington, DC 20375  
*firstname.lastname@knexusresearch.com* | *david.aha@nrl.navy.mil*

## 1 ABSTRACT

A variety of anomaly detection algorithms have been applied to surveillance tasks for detecting threats with some success. However, it is not clear which anomaly detection algorithms should be used for domains such as ground-based maritime video surveillance. For example, recently introduced algorithms that use local density techniques have performed well for some tasks, but they have not been applied to ground-based maritime video surveillance. Also, the reasons for the performance differences of anomaly detection algorithms on problems of varying difficulty are not well understood. We address these two issues by comparing families of global and local anomaly detection algorithms on tracks extracted from ground-based maritime surveillance videos. Obtaining maritime anomaly data can be difficult or even impractical. Therefore, we use a generative approach to vary and control the difficulty of anomaly detection tasks and to focus on borderline and difficult situations in our empirical comparison studies. We report that global algorithms outperform local algorithms when tracks have large and unstructured variations, while they perform equally well when the tracks have only minor variations.

## 2 INTRODUCTION

Anomaly detection is the task of finding patterns in data that do not conform to expected behavior [1]. It is an important problem in applications such as maritime video surveillance, which is typically performed from one of two perspectives: 1) wide area coverage, where vessels are tracked across large geographical areas (e.g., such as when tracking international shipping vessels using AIS), and 2) ground-based coverage, where vessels and their activities are tracked over comparatively small distances (e.g., 1000-5000 yards). We focus on ground-based video surveillance of maritime locations such as ports, harbors, and rivers [2]. Maritime assets are vulnerable to attacks from a variety of near-shore threats such as small boats. Currently, such threats are predominantly countered by watchstanders and manual video surveillance processes. We are developing automated techniques for monitoring and recognizing activities in video, and for detecting threatening behavior [3]. We model this as an anomaly detection task, which requires selecting and applying anomaly detection algorithms to the problem of maritime threat detection.

Two broad categories of unsupervised anomaly detection algorithms exist. First, *global* methods consider information contained in an entire data set to identify anomalies. Most clustering algorithms fall into this category. Second, *local* methods operate on only the information in the neighborhood of the query. Recent work on applications of anomaly detection to complex structured data such as credit card transactions have shown that local, non-clustering anomaly detection algorithms can outperform global algorithms on some tasks [4][5][6]. However, detailed results have not been published on applications of local anomaly detection

algorithms to ground-based maritime video surveillance tasks, they have not been compared against global methods on these tasks, yet the structural complexities of maritime traffic data, which are not well understood, may benefit from using local methods.

We test this conjecture by comparing the performances of local and global anomaly detection algorithms on ground-based maritime video data. In our investigation, we focus on borderline/boundary cases of anomalous behavior. This is challenging because anomalies may rarely occur in the maritime domain. Obtaining real data of anomalous behavior can be impractical, and even more so for borderline cases. We address this by generating synthetic anomalous instances using a parametric statistical approach from real, non-anomalous data, which allows us to control the frequency and degree of abnormal behaviors present in the generated data. We evaluate two global and two local algorithms on three categories of ground-based maritime video traffic data with varying complexities and noise. We found that, for difficult cases, density-based global anomaly algorithms outperform local algorithms on a data set characterized by tracks with high variance or noise (e.g., tracks from sailboat traffic), while local anomaly detection algorithms perform as well or better than the global algorithms when the tracks have low variance (e.g., tracks from recreational boats and ferries).

Our paper is organized as follows. In Section 2, we review anomaly detection research in the context of maritime threat analysis. Section 3 describes the anomaly detection algorithms we comparatively evaluate, while Section 4 describes our evaluation method and presents the results of our analysis. We discuss our findings in Section 5, and conclude the paper with directions for future research in Section 6.

### 3 BACKGROUND AND RELATED WORK

Our goal is to provide watchstanders with tools for automated surveillance using ground-based videos of littoral locations such as ports, harbors, and rivers. Given this, we are developing methods that can automatically assess the threat status of detected surface vessels given their tracks and raise alerts when warranted. We can model this as an anomaly detection task, in which a set of normalcy models, anomaly models, or some combination could be acquired and then applied to predict whether to respond to an observed track with an alert.

Anomaly detection tasks, models, and algorithms for surveillance tasks can differ along many dimensions. We characterize the relevant literature along five dimensions: (1) the model acquisition method (manually elicited vs. learned), (2) the application focus (maritime vs. non-maritime), (3) the type of coverage (ground-based vs. wide area), (4) whether they integrate contextual domain knowledge, and (5) their model category (i.e., global vs. local).

Some models for anomaly detection can be created manually, such as by eliciting anomaly models in the form of rules from experts[7], but this may be impractical if experts are not available, they cannot easily provide these models, or the elicitation cost may be high. In this paper, we focus on algorithms that *automatically* learn anomaly detection models for maritime vessels, where the tracks are derived from ground-based optical video, and no domain-specific knowledge is employed. In this context, we will compare the performance of global with local methods.

Many algorithms for automated anomaly detection exist [1], and have been discussed in the KDD, machine learning, pattern recognition, and related literatures. Some of them have been applied to maritime surveillance tasks. For example, Kraiman et al.'s [8] Automated Anomaly Detection Processor (AADP) uses self-organizing maps (SOMs) to cluster tracks, and couples Gaussian mixture models (GMMs) with Bayesian techniques to perform decision making. Rhodes et al. [9] instead use Fuzzy ARTMAP to continuously learn

normalcy models for designated maritime regions. Johansson et al. [10] instead propose a Bayesian approach, while Dahlbom and Niklasson [11] find that trajectory clustering has practical limitations, and propose a method that instead models normal trajectories with splines. Laxhammar [12] describes a greedy expectation-maximization (EM) algorithm for learning the parameter settings of multivariate GMM models, while Laxhammar et al. [13] report a study in which its performance does not significantly differ from that of an adaptive kernel density estimator. Although these approaches were all knowledge-poor, some recent algorithms leverage substantial domain knowledge. For example, Bostwick et al. [14] describe a probabilistic case-based reasoning (CBR) approach for identifying anomalous tracks that accesses and reasons about supplemental information such as weather patterns, piracy events, and vessel ownership changes. Willems et al.'s [15] system instead derives piecewise linear segmentation models from tracks and uses an event model that integrates sensor data with information derived from the Internet. Their model can be queried via a visual analytics tool to search for abnormal spatial-temporal tracks. In summary, while these automated anomaly detection algorithms vary substantially (e.g., in the sensors they use to derive tracks, the track representations they operate on, and the contextual knowledge they leverage), they all employ a wide area coverage perspective, which differs from our focus on ground-based coverage. That is, we derive tracks from optical land-based video cameras, and do not assume, for example, that the small vessels being monitored are outfitted with AIS.

With few exceptions, most of the algorithms mentioned above employ statistical techniques that fit distributions to the observed track data, learn normalcy models, and use an outlier detection method to identify anomalies. Also, most employ global algorithms, and cluster tracks over a large geographical area. Unfortunately, global algorithms that use standard clustering algorithms cluster outliers, and can perform poorly when their distributional assumptions are violated or when the data exhibit complexities such as large variances in distribution densities. To ameliorate these problems, some of the approaches mentioned above (e.g., [12]) employ a hybrid global/local method that divides the geographical area into grid cells and applies a global method in each cell. However, this fix has several limitations. For example, it ignores contextual information and does not reflect a natural geographical partition [13].

We conjecture that maritime surveillance applications exhibit complex structures where *local* anomaly detection algorithms may perform comparatively well. Among those mentioned above, only Bostwick et al. [14] learn local models, but their method does not use standard density-based methods for outlier detection (e.g., LOF [3], LOCI [6]), and we are not aware of any comparisons of local and global anomaly detection algorithms in the literature on maritime surveillance. In the following sections, we describe a set of global and local anomaly detection methods and detail their empirical comparison on a ground-based maritime surveillance task.

## 4 ANOMALY DETECTION ALGORITHMS

We expect that different kinds of maritime traffic are characterized by different levels of complexity and noise. For example, small sailboats may have complex chaotic behavior patterns that differ markedly from those of mid-sized ferries. Thus, an anomaly detection algorithm's performance could vary depending on the type of maritime traffic. We investigate this conjecture by comparing two global and two local anomaly detection algorithms. The global algorithms we consider are an EM version of k-means clustering and the k-NN Localized p-value Estimator (KNN-LPE). K-means is a popular distance-based clustering algorithm while KNN-LPE performs global density-based anomaly detection. The local algorithms we consider are a variant

of the Local Outlier Factor (LOF) and k-NN Normalized Average Density (NAD), which are both density-based anomaly detection algorithms. We describe these algorithms in the following subsections.

#### 4.1 Expectation Maximization k-Means Clustering

Clustering is the assignment of a set of observations into subsets called *clusters* that, typically, minimizes the distance between observations within a cluster and maximizes the distance between observations that belong to different clusters. Clustering can be used to detect anomalies based on one or more of the following assumptions[1]:

1. Normal data instances lie close to the closest cluster centroid while anomalies are further away.
2. Normal data instances belong to large and dense clusters, while anomalies belong to small or sparse clusters.
3. Normal data instances belong to a cluster, while anomalies do not belong to any cluster. This assumption is somewhat at odds with the traditional clustering techniques that require all nodes to belong to some cluster.

In this paper, we test the k-means clustering algorithm [16] for anomaly detection and adopt assumptions 1 and 2. It starts with a random selection of  $k$  observations as the cluster centers. Next, it iterates over an assignment step and an update step to find the optimum clusters. In the assignment step, it assigns an observation to its closest cluster center. In the update step, the cluster centers are recomputed based on their member observations. The algorithm terminates when the assignments do not change during two consecutive iterations. The standard k-means algorithm requires a manual specification of the number of clusters  $k$  as input. However, an optimal  $k$  and its associated clusters can be automatically obtained using an EM approach, which performs probabilistic assignments of observations to clusters instead of deterministic assignments and maintains multivariate Gaussian distributions instead of means as cluster centers.

The EM k-means clustering algorithm for anomaly detection has two phases: training and decision making. The training phase includes the following steps:

1. *Expectation maximization clustering*: Observations are used to generate the clusters. The distance of an observation to a centroid along one feature is computed using log normal probability density. The overall distance of an observation to a cluster center is the sum of the log normal densities of its  $n$  features.

$$diff(x_i) = x - mean$$

$$logNormalDensity(x_i) = \left( \frac{diff(x_i)^2}{2\sigma^2} \right) - C - \log(\sigma)$$

$$logProbability(x) = \left( \sum_{i=0}^n logNormalDensity(x_i) \right)$$

where  $\sigma$  is the standard deviation of the cluster,  $C$  is a constant,  $x$  is a test instance, and  $x_i$  is a feature of  $x$ . *The optimum number of clusters* is the one that maximizes the log probabilities of observations in a validation set.

2. *Sparse cluster identification*: Sparse clusters are used to identify anomalous instances during the test phase. We identify sparse clusters as those that have a low prior and a standard deviation that is significantly higher than the standard deviations of other clusters. The identification rule is as follows:

IF  $((\sigma_{cl} > \mu_{\sigma_{cl}} + A \cdot \sigma_{\sigma_{cl}}) \wedge (prior < \alpha))$   
 THEN label  $cl$  as a sparse cluster

where  $\sigma_{cl}$  is the deviation in a cluster,  $\mu_{\sigma_{cl}}$  is the mean of cluster deviations,  $\sigma_{\sigma_{cl}}$  is the deviation of cluster deviations, and  $A$  and  $\alpha$  are constants.

In the decision making phase, we label an instance as anomalous or normal by computing the distance of the instance to the clusters and identifying the closest cluster. The instance is anomalous if it is either associated with a sparse cluster or the distance to its closest cluster is greater than a specified threshold ( $\tau$ ).

#### 4.2 k-NN Localized p-value Estimator

Conventional k-nearest neighbor (k-NN) approaches use local neighborhood information to detect anomalies [17]. They label an instance as anomalous when the distance to its  $k^{\text{th}}$  nearest neighbor exceeds a specified threshold. In contrast, a k-NN Localized p-value Estimator (LPE) [5] is a global anomaly detection algorithm that uses the entire training set to compute a score, which is an estimate of the probability that it is anomalous. This score ( $S_{te}$ ), or *p-value estimate*, for a test instance  $te$  is computed as follows:

$$S_{te} = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} 1(d_{te} \leq d_i)$$

where  $N_{tr}$  is the number of training instances, the node density  $d_i$  of an instance  $i$  is defined as the distance between  $i$  and its  $k^{\text{th}}$  nearest neighbor in the training set, and  $1()$  is the indicator function.

A test instance is labeled as anomalous if its score  $S_{te}$  is greater than a specified threshold  $\alpha$ , which represents the probability of an alarm. This threshold can be adjusted for a desired false alarm rate. This algorithm's runtime complexity for a single test instance is  $O(N_{tr}^2)$  as it must identify the  $k$  nearest neighbors of each training instance. When applying this to test instances, we mitigate this computational cost by caching the training node densities.

#### 4.3 Local Outlier Factor

The Local Outlier Factor (LOF) is a measure of the degree to which an instance is an outlier or an anomaly [4]. It is a local anomaly detector since it relies on only a restricted neighborhood of the test instance. It is particularly suitable for outlier analysis in large multi-dimensional data sets. The restricted neighborhood is defined by the input parameter  $k$  (the notation *MinPts* was instead used in the original paper).

Computing LOF of an instance  $x$  includes the following elements:

- $k$ -distance( $x$ ): is the density estimate of an instance  $x$ , defined as:

$$k\text{-distance}(x) = d(x, x_k)$$

where  $d(x, x_k)$  is the distance between  $x$  and its  $k^{\text{th}}$  nearest neighbor.

- $N_k(x)$ : is the  $k$ -neighborhood of  $x$ , which includes all instances whose distances from  $x$  are not greater than  $d(x, x_k)$ . This allows instances to be equally far away, which may occur when distances are measured in discrete units rather than real numbers. Since we compute distances as real numbers, we define the  $k$ -neighborhood of  $x$  to be its set of  $k$ -nearest neighbors.
- $lrd(x)$ : is the local reachability density of an instance  $x$ :

$$lrd_k(x) = 1 / \left\{ \frac{\sum_{o \in N_k(x)} reach-dist_k(x,o)}{|N_k(x)|} \right\}$$

where  $reach-dist_k(p,o)$  is the reachability distance for an instance  $x$  with respect to another instance  $o$  ( $o \in N_k(x)$ ), defined as:

$$reach-dist_k(x,o) = \max \{k\text{-distance}(o), d(x,o)\}$$

The reachability distance tempers the effect of high-density neighborhoods on the LOF computation. It forces the density of instances  $p$  that are close to  $o$  to the same value (i.e.,  $N_k(o)$ ) and those instances that are far away from  $o$  to retain the larger densities  $d(x,o)$ .

- LOF( $x$ ): is the local outlier factor of an instance  $p$ , defined as:

$$LOF_k(x) = \frac{\sum_{o \in N_k(x)} \frac{lrd_k(o)}{lrd_k(x)}}{|N_k(x)|}$$

A LOF value of 1 indicates an instance has the same density relative to its neighbors, while a value less than 1 indicates an inlier, and a value significantly greater than 1 indicates that it is an outlier/anomaly.

We classify an instance as anomalous if it is greater than a specified threshold ( $\tau$ ), whose value may be set based on a desired false positive rate.

Our preliminary trials showed that LOF, as defined, does not perform well on our task. Consequently, we created LOF Normalized (LOFN), a version of LOF that linearly scales the feature values of all instances to  $[0,1]$ . We perform this normalization as follows:

$$f_i^{\text{Normalized}} = (f_i - f_{\min}) / (f_{\max} - f_{\min})$$

where  $f_i$  is the non-normalized value of feature  $f$  in instance  $i$ ,  $f_{\min}$  is the minimum value of  $f$  in the training set, and  $f_{\max}$  is the maximum value of  $f$  in the training set. LOFN's runtime is also  $O(N^2)$ .

#### 4.4 k-NN normalized average density

We introduce k-NN Normalized Average Density (NAD), a variant of LOFN that simplifies its density calculation and reduces its run time. Instead of computing local reachability distances, we estimate a node's density as the distance between itself and its  $k^{\text{th}}$  nearest neighbor. In particular, we replace the  $lrd(x)$  computation in LOFN with

$$lrd_k^{\text{Simple}}(x) = 1 / \left( \frac{1}{k} \sum_{i=1}^k dist(x_i, x_k) \right)$$

The k-NN NAD score is computed in the same way as in LOFN (item 4, Section 3.3) and is compared to a threshold ( $\tau$ ) to determine whether it is an anomaly. The runtime of this algorithm is also  $O(N_{tr}^2)$ , as it must compute the nearest neighbor of each training instance.

## 5 EVALUATION

### 5.1 Objective

Our goal is to compare the performance of local and global anomaly detection algorithms on a ground-based maritime anomaly detection task. An important consideration in our investigation is the degree of prediction difficulty. We expect the algorithms to differ only on borderline cases and not when the anomaly cases are clearly distinct from normal instances.

### 5.2 Method

**Data.** We created a data set of maritime tracks from a combination of observed surveillance data and synthetically generated data.

*Potomac River Surveillance Tracks.* We collected two weeks of video data of maritime traffic on the Potomac River using a fixed camera mounted on a building at the Naval Research Laboratory in Washington, DC. Our camera recorded black and white digital images. We used background subtraction for object detection and tracking techniques [2]. These automatically extracted tracks were annotated for objects and activities by subject matter experts (SME) using a maritime ontology extracted from United States Coast Guard navigation guidelines [18].

**Table 1:** Potomac maritime traffic data summary

Vessel Category	Number of Occurrences in Tracks
Sailboat	949
Recreational Vessel	375
Passenger	365
Utility Vessels	106
Row Boat	43
Cargo Vessel	24
Fishing Vessel	4

**Table 2:** Standard deviations across features of three categories of surface vessels

Feature	Sailboat	Recreational Vessel	Simulated Vessel
$x$	311.7	364.9	256.3
$y$	<b>25.7</b>	19.3	9.0
$v_x$	2.0	10.0	11.4
$v_y$	3.3	4.8	0.1

Table 1 displays the set of vessels detected in this video data and the number of tracks in which they were observed. We selected tracks from the two most frequent vessel categories (i.e., Sailboat and Recreational Vessel) to evaluate the anomaly detection algorithms. We also generated synthetic tracks for a Simulated Vessel category with fewer positional and velocity variations using a uniform distribution. Our motivation for creating this simulated data was that we expected large algorithmic performance differences when feature

value variations are large. We conjectured that certain vessel categories such as sailboats have larger value deviations on certain features.

Table 2 shows the four features we used to represent tracks in our study and their variability for these three vessel categories. Features  $x$  and  $y$  are position values along the  $x$  and  $y$  image axes, while  $v_x$  and  $v_y$  are their velocities. As highlighted, sailboats have notably higher deviations along the  $y$  axis compared to the other categories.

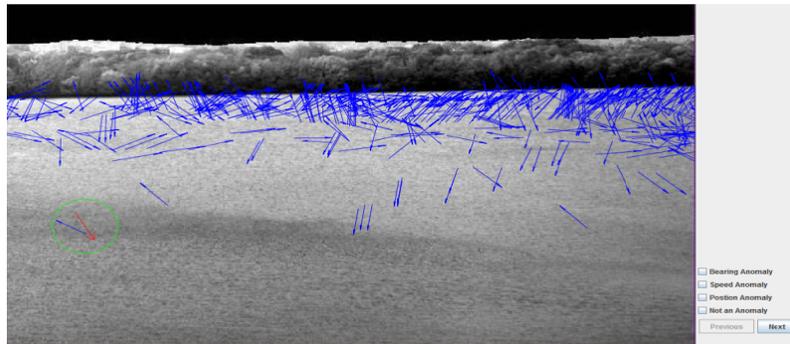
Since the surveillance data rarely contain anomalies, we synthesized anomalous tracks from the observed tracks as follows. We partitioned the observed tracks into two parts:

1. *Non-anomaly surveillance data set*: We used this set to select non-anomalous data for the test set.
2. *Data generation set*: We used this set to estimate data generation parameters and to generate synthetic anomalous tracks. We then annotated these tracks for anomalies.

*Synthetic Track Generation*. We generated the feature values for synthetic tracks by using the mean and standard deviation of the feature value in the data generation set as follows:

where  $\sigma$  is the standard deviation,  $\mu$  is the mean,  $\text{randGaussian}()$  provides a value sampled from a univariate Gaussian distribution with a mean of 0 and standard deviation of 1, and  $\Delta$  is the percentage change to the mean. We used this approach to independently generate the values for each track's four features. We generated 100 synthetic tracks, where  $\Delta$  was varied from 20% to 100% in increments of 20.

*Track Annotation*: The synthesized tracks potentially contain anomalies that must be annotated for ground truth. We annotated tracks using a software tool we developed called the Anomaly Annotator (see Figure 1), which also displays the training data to SMEs as a baseline. The blue arrows represent the velocities and positions of observed maritime objects in the training data. The length of each arrow denotes its magnitude and the pointer denotes its bearing. The candidate to be annotated is displayed using a red arrow. The SME can mark the candidate as a bearing anomaly, speed anomaly, position anomaly, or as non-anomalous. For this paper, we collapsed the three types of anomalies into one type. Two SMEs provided their inputs on each synthetic track to allow for more robust anomaly annotations. We annotated synthetic tracks as anomalous only if both SMEs marked them as anomalous.



**Figure 1:** A screenshot of the Anomaly Annotator being used by an SME to annotate synthetic tracks (shown in red) generated from observed Potomac River tracks

*Test Set Generation.* For each vessel category, we sampled 64 instances of non-anomalous data from the surveillance set and combined them with random samples of 24 anomalies from the synthesized tracks for each perturbation level.

**Algorithm Implementations and Optimal Parameter Settings.** We implemented the four algorithms described in Section 3:

1. *Expectation Maximization k-Means Clustering* (EM-KMC): We used the Weka implementation for this algorithm.<sup>1</sup> We also implemented the sparse cluster identification rule we described in Section 3.1.
2. *k-NN Localized p-value Estimator* (KNN-LPE): We implemented this using the Nearest Neighbor Classification Library available in the Knexus Classification Workbench (KCLAW). We used the Euclidean metric to measure the distance between instances.
3. *Local Outlier Factor Normalized* (LOFN): This was implemented as described in Section 3.3.
4. *k-NN Normalized Average Density* (KNN-NAD): This was implemented as described in Section 3.4.

#### *Algorithm Parameter Settings*

The global and local density-based anomaly detection algorithms input the parameter  $k$ , which is the number of nearest neighbors. We identified the value of  $k$ , between 3 and 30, that optimized each algorithm’s AUC performance values on each category using a validation set. The validation set was generated in the same way as the training sets using data set aside for validation. These values are displayed in Table 3.

**Table 3:** Optimal  $k$  values for each density-based anomaly detection algorithm per category

Vessel Categories	Local		Global
	LOFN	KNN-NAD	KNN-LPE
Sailboat	18	5	3
Recreational Vessel	17	23	3
Simulated Vessel	9	9	8

**Performance Metrics.** We use the Area Under the Receiver Operating Characteristic (ROC) curve (AUC) to assess the performance of these algorithms. ROC curves are often used to compare the performance of binary classifiers [19]. A ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR).

$$\text{TPR} = \frac{\text{\#anomalies correctly classified}}{\text{Total number of anomalies}}$$

$$\text{FPR} = \frac{\text{\#non-anomalies classified as anomalies}}{\text{Total number of non-anomalies}}$$

For each anomaly detection algorithm, we generate a ROC curve by plotting the TPR (y-axis) against the FPR (x-axis) while varying the threshold  $\tau$  until the TPR reaches 100%. We compute the AUC using the Gini coefficient [20]. A larger value of AUC indicates a superior performance. A straight line from (0,0) to (1,1), called an *indifference curve*, serves as the baseline detector.

We also report the runtime of the algorithms in milliseconds.

**Test Procedure.** We presented each algorithm with the same training and test sets. For each set, we trained an anomaly detector and recorded their TPR and FPR against the test sets for the range of incremental detection thresholds needed for the ROC curve.

---

<sup>1</sup> <http://www.cs.waikato.ac.nz/ml/weka/>

**Analysis.** We ranked the algorithms’ performances according to their AUC values, and report their average rankings.

### 5.3 Results and analysis

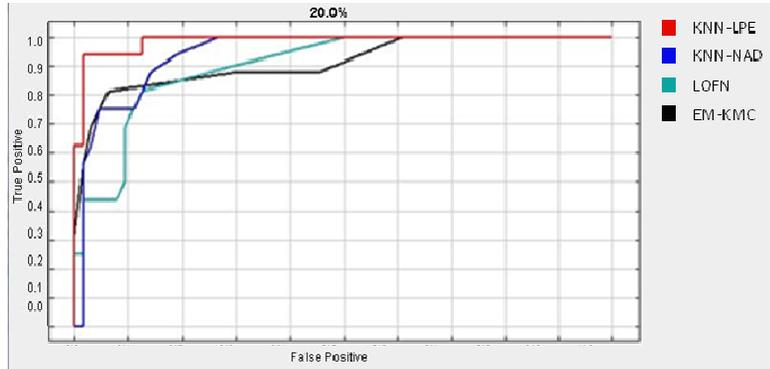
Table 4 summarizes the AUC values and runtimes obtained for the local and global anomaly detection algorithms on the three categories of maritime vessels in our study. As shown, the density-based global algorithm KNN-LPE is the dominant performer in the Sailboat category for the AUC metric, while the density-based local algorithm KNN-NAD is the second best performer for this metric. Figure 2a shows that, for a 10% false alarm rate, KNN-LPE achieves a 94% TPR, while the second highest TPR for this rate was recorded by EM-KMC (82% TPR).

For the Recreational Vessel category KNN-LPE remains the best AUC performer, although KNN-NAD is a very close second. Figure 2b shows that, for a 10% false alarm rate, KNN-LPE (TPR 94.3%) outperforms KNN-NAD (TPR 91.7%).

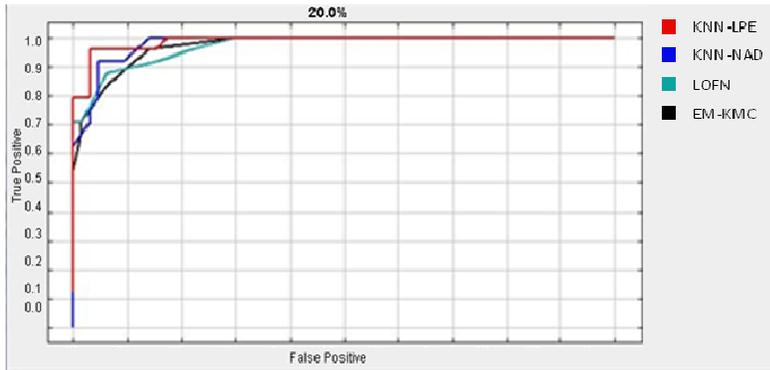
For the Simulated Vessel data, at 20% perturbation, the local density-based algorithms marginally outperform the global algorithms. For example, the KNN-NAD and LOFN have perfect AUCs of 1 and the EM-KMC and KKN-LPE have AUCs of 0.990 and 0.998, respectively. This difference is practically inconsequential (see Figure 2c).

**Table 4:** AUC values and average run times for the local and global anomaly detection algorithms on three categories of maritime data

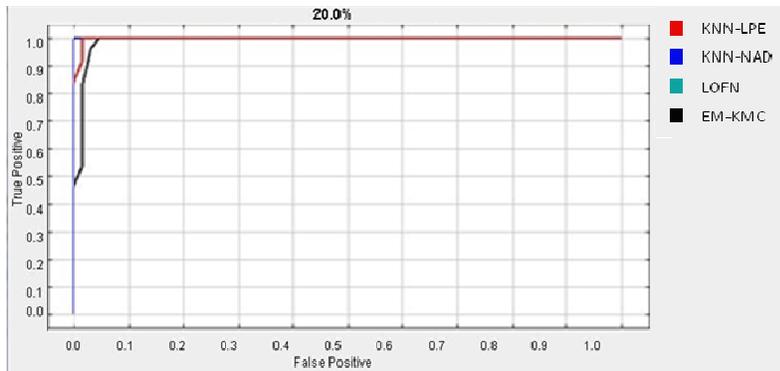
	Local		Global	
	LOFN	KNN-NAD	EM-KMC	KNN-LPE
<b>Sailboats</b>				
20%	0.901	0.944	0.911	<b>0.987</b>
40%	0.726	0.915	0.911	<b>0.991</b>
60%	0.813	0.927	0.971	<b>0.993</b>
Average Rank	4	2.33	2.66	<b>1</b>
Average Runtime (ms)	15443	322	29417	465
<b>Recreational Vessels</b>	6758			
20%	0.970	0.980	0.971	<b>0.987</b>
40%	0.995	0.995	0.999	1.000
60%	1.000	1.000	1.000	1.000
Average Rank	2.66	2	2	<b>1</b>
Average Runtime (ms)	2682	217	3728	38
<b>Simulated Data</b>				
20%	<b>1.000</b>	<b>1.000</b>	0.990	0.998
40%	1.000	1.000	1.000	1.000
60%	1.000	1.000	1.000	1.000
Average Rank	<b>1</b>	<b>1</b>	1.66	1.33
Average Runtime (ms)	1382	164	2167	80



**Figure 2a:** ROCs for Sailboat tracks



**Figure 2b:** ROCs for Recreational Vessel tracks



**Figure 2c:** ROCs for Simulated Vessel tracks

**Figure 2:** ROC curves for all the test sets at 20% perturbation.

The global algorithms perform robustly and outperform the local algorithms when the variances in speed and position are large (e.g., as with the Sailboats category). The performance gap between the global and local anomaly detection algorithms decreases with these variances. Furthermore, as expected, when anomalies are more distinct (i.e., 40% or 60% perturbations), the AUC performance differences among algorithms is

substantially reduced. As shown in Table 4, KNN-LPE, despite being a global algorithm, has the fastest run time. Thus, it is a promising algorithm to use for these conditions.

## 6 DISCUSSION

Anomaly detection using normalcy models is among the most commonly used technique for maritime threat detection. For example, Rhodes et al. [9] and Laxhammer et al. [13] describe applications of global anomaly algorithms to large vessels maritime traffic in open ocean. However, Rhodes et al [9] do not report any empirical evaluation. Furthermore, they focus on ocean going vessel traffic, which compared to small boat traffic is well behaved and less noisy because the vessels can be tracked via their Automated Identification Systems. Under such conditions we conjectured and demonstrated, at least for our data, that there are no notable performance differences across anomaly detection algorithms.

In this paper, we were predominantly concerned with small vessel traffic in coastal areas and in rivers, whose activities are erratic and noisy when tracked via ground-based video surveillance. These situations are likely to include difficult to detect or borderline anomalies. Given the availability of a wide variety of anomaly detection algorithms [1], their application to small vessel maritime traffic and can result in unpredictable performances. We expected the performance differences between anomaly detection to be substantial for such borderline cases. As expected, our comparative evaluation of global and local algorithms shows that, indeed, there are substantial performance differences between them. However, the surprising result is that the global algorithms outperform the local algorithms both on accuracy and speed. This is in contrast to the evaluations reported by Jassens et al. [6], possibly because they evaluated performance on non-maritime data where the problem characteristics are likely to differ from ours. Additionally, we systematically controlled the degree of anomalies, which enabled us to isolate the differences between the two categories of algorithms.

Our ultimate objective is real-time threat detection using ground-based surveillance. Thus, the computational speed of anomaly detection algorithms is of significant concern. Typically, the global algorithms are more computationally expensive in time than local algorithms. However, we find that the KNN-LPE, a global density based approach, has the lowest runtime on two of the three vessel categories we examined, possibly because it lends itself to caching of distance computations.

Our investigations in this paper have several limitations. First, we evaluated the algorithms on only two categories of (real) vessels. Second, although we created simulated tracks based on actual data, our simulation algorithm made independence assumptions across features. This could generate unrealistic tracks. We will address these limitations in our future work by increasing the categories of vessels and using multi-variate distributions for synthetic data generation.

## 7 CONCLUSION

Threat detection in maritime environments is of vital interest to the US Navy and the Department of Homeland Security. In particular, threat of attack on assets and maritime infrastructure from small vessels is a key concern. We addressed this by systematically evaluating performance differences among anomaly detection algorithms on small vessel maritime traffic. Our contributions in this paper are twofold. First, we show that some small boat traffic characteristics can be markedly different in nature than open ocean

maritime traffic. This difference must be considered in the selection and application of anomaly detection algorithms to the threat detection problem. Using generated data, we showed that a density based global algorithm outperforms two local algorithms in accuracy and speed for two of the three vessel categories we examined, making them attractive for ground-based real-time surveillance of small vessel traffic. Second, we showed that the performance differences between local and global algorithms are inconsequential for more simpler anomaly detection problems.

There are several avenues of future research that we intend to pursue. We will consider the temporal aspects of anomalous scenarios and model them using approaches for sequence or activity labeling such as Hidden Markov Models [21] and Conditional Random Fields [22]. Most existing work on threat detection focuses on analyzing the behavior of single vessels. However, threats from coordinated activities are also a serious concern. We intend to address coordinated threats using probabilistic relational models such as Markov Logic Networks [23].

### ACKNOWLEDGEMENTS

This research was supported by the Office of Naval Research.

### REFERENCES

- [1] Chandola, V., Banerjee, A., & Kumar, V. "Anomaly detection: A survey." Minneapolis, MN: University of Minnesota, Department of Computer Science & Engineering Technical Report, 07-017 (2007).
- [2] Gupta, K.M., Aha, D.W., & Moore, P. "Case-based collective inference for maritime object classification." *Proceedings of the Eighth International Conference on Case-Based Reasoning*. Seattle, WA: Springer 443-449 (2009).
- [3] Gupta, K.M., Aha, D.W., & Hartley, R. "Adaptive maritime video surveillance." *Proceedings of the Society of Photographic Instrumentation Engineers Conference*. Orlando, FL: SPIE, (2009).
- [4] Breunig, M.M., Kriegel, H.-P., Ng, R.T., & Sander, J. "LOF: Identifying density-based local outliers." In *Proceedings of the ACM SIGMOD Conference on Management of Data*. Dallas, TX: ACM Press, (2000).
- [5] Zhao, M., & Saligrama, V. "Outlier detection via localized p-value estimation." *Proceedings of the Forty-Seventh Annual Allerton Conference on Communication, Control, and Computing*. Monticello, IL: IEEE Press, 1482-1489 (2009).
- [6] Janssens, J.H.M., & Postma, E.O. "One-class classification with LOF and LOCI: An empirical comparison." *Proceedings of the Eighteenth Annual Belgian-Dutch Conference on Machine Learning*. Tilburg, The Netherlands: ACL, 56-64 (2009).
- [7] Nilsson, M., van Laere, J., Ziemke, T., & Edlund, J. "Extracting rules from expert operators to support situation awareness in maritime surveillance." *Proceedings of the Eleventh International Conference on Information Fusion*. Cologne, Germany: IEEE, 1-8 (2008).
- [8] Kraiman, J.B., Arouh, S.L., & Webb, M.L. "Automated anomaly detection processor." In *Enabling Technologies for Simulation Science VI*. Orlando, FL: SPIE, (2002).
- [9] Rhodes, B.J., Bomberger, N.A., Seibert, M., & Waxman, A.M. "Maritime situation monitoring and awareness using learning mechanisms." In *Proceedings of the IEEE Military Communications Conference*. Atlantic City, NJ: IEEE Press, (2005).
- [10] Johansson, F., & Falkman, G. "Detection of vessel anomalies – a Bayesian network approach." In *Proceedings of the Third International Conference on Intelligent Sensors, Sensor Networks, and Information Processing*. Melbourne, Australia: IEEE Press, (2007).
- [11] Dahlbom, A., & Niklasson, L. "Trajectory clustering for coastal surveillance." In *Proceedings of the Tenth International Conference on Information Fusion*. Quebec City (Quebec), Canada: IEEE Press, (2007).
- [12] Laxhammar, R. "Anomaly detection for sea surveillance." In *Proceedings of the Eleventh International Conference on Information Fusion*. Cologne, Germany: IEEE Press, (2008).

- [13] Laxhammar, R., Falkman, G., & Sviestins, E. "Anomaly detection in sea traffic - a comparison of gaussian mixture model and kernel density estimator." *Proceedings of the Twelfth International Conference on Information Fusion*. Seattle, WA: IEEE Press, 756-763 (2009).
- [14] Bostwick, D., Goldstein, J., Stephenson, T., Stromsten, S., Tierno, J., Torrelli, M., & White, J. "PARSEC, an application of probabilistic case based reasoning to maritime surveillance." In *Proceedings of the IEEE Conference on Technologies for Homeland Security*. Boston, MA: IEEE Press, (2009).
- [15] Willems, N., van Hage, W.R., de Vries, G., Janssens, J., & Malaisé, V. "An integrated source for visual analysis of a multi-source moving objects knowledge base." *International Journal of Geographical Information Science*, **24**, 1553-1568 (2010).
- [16] MacQueen, J.B. "Some methods for classification and analysis of multivariate observations." *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, CA: University of California Press, 281-297 (1967).
- [17] Zhang, K., Hutter, M., & Jin, H. "A new local distance-based outlier detection approach for scattered real-world data." *Proceedings of the Thirteenth Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. Bangkok, Thailand: Springer, 813-822 (2009).
- [18] NavRules., [Navigation rules, International – Inland] (Commandant Instruction M16672.2D). Washington, DC: U.S. Department of Transportation, United States Coast Guard, (1999).
- [19] Bradley, A.P "The use of the area under the ROC curve in the evaluation of machine learning algorithms." *Pattern Recognition*, **30**(7), 1145-1159 (1997).
- [20] Gini, "Gini Coefficient." Retrieved from [http://en.wikipedia.org/wiki/Gini\\_coefficient](http://en.wikipedia.org/wiki/Gini_coefficient), on 22 Sep, (2010).
- [21] Rabiner, L.R. "A tutorial on Hidden Markov Models and selected applications in speech recognition." *Proceedings of the IEEE*, **77**(2), 267-286 (1989).
- [22] Lafferty, J., McCallum, A., Pereira, F. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." *Proceedings of the Eighteenth International Conference on Machine Learning*. Williamstown, MA: Morgan Kaufmann, 282-289 (2001).
- [23] Richardson, M., & Domingos, P. "Markov logic networks." *Machine Learning*, **62**, 107–136 (2006).