Anomaly Detection from Time-Changing Environmental Sensor Data Streams

by

© Abdullah-Al-Mamun

A thesis submitted to the

School of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Memorial University of Newfoundland

January 2016

St. John's Newfoundland

To my late father whom I lost last year. I cherish the past we shared but miss the future we will not have.

Abstract

This thesis stems from the project with real-time environmental monitoring company EMSAT Corporation. They were looking for methods to automatically flag spikes and other anomalies in their environmental sensor data streams. The problem presents several challenges: near real-time anomaly detection, absence of labeled data and time-changing data streams. Here, we address this problem using both a statistical parametric approach as well as a non-parametric approach like Kernel Density Estimation (KDE).

The main contribution of this thesis is extending the KDE to work more effectively for evolving data streams, particularly in presence of concept drift. To address that, we have developed a framework for integrating Adaptive Windowing (ADWIN) change detection algorithm with KDE. We have tested this approach on several real world data sets and received positive feedback from our industry collaborator. Some results appearing in this thesis have been presented at ECML PKDD 2015 Doctoral Consortium.

Acknowledgments

I would like to express my utmost gratitude to my thesis supervisor, Dr. Antonina Kolokolova, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate her vast knowledge and skills in significant areas, and her comments on reports (i.e. research proposals, research paper and this thesis), which have inspired me enough to follow her guideline throughout. Each weekly meeting with her has enlightened me about new prospects of the research. Every time I had faults within the idea, she pointed those out in such a encouraging way that drove me to strive for better solutions. Moreover, I am really happy to have the opportunity to work under the supervision of an extremely nice person like her.

I would also like to take the opportunity to thank Dr. Miklos Bartha and Dr. Todd Wareham for stimulating discussion sessions and helping me learn ample new concepts during the graduate program.

A very special thanks to my undergraduate thesis supervisor Dr. Rezwanul Huq, without whose teaching and supervision I would not have considered a research based graduate program in computer science. I doubt that I will ever be able to convey my appreciation fully, but I owe him my eternal gratitude.

I am thankful to SmartBay project for granting access to the raw data generated by their buoys. I also must acknowledge Mr. Dan Brake of EMSAT Corporation for his valuable suggestions and feedback of the experimental results. A special thanks to my friends in the Department of Computer Science, particularly Mr. Tamkin Khan,

for sharing his knowledge on programming, Mr. Abdullah Ali Faruq, for exchanging ideas, Mr. Mustafa Kamal Bhuiyan and Ms. Zahra Sajedinia for effective discussions during coursework, Mr. Mitu Kumar Debnath for working together before deadlines and venting of frustration during our graduate program. I would also like to thank Mr. Taufiqur Rahman for his help to learn Matlab and Mr. Raju Hossain for helpful discussions.

I must thank my parents for their love and support they provided me through my entire life. I also acknowledge the contribution of my elder brother and sisters to pursue this graduate program. A special thanks to Joyeeta for motivating and supporting me.

In conclusion, I recognize that this research would not have been possible without the financial assistance of NSERC Engage Grant, Memorial university of Newfoundland (MUN) Graduate Studies, the Department of Computer Science at MUN(Teaching Assistantships), and express my gratitude to those agencies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Large amounts of quickly generated data have shifted the focus in data processing from offline, multiple-access algorithms to online algorithms tailored towards processing a stream of data in real time. Data streams are temporally ordered, fast changing and massive. Wireless sensor network traffic, telecommunications, on-line transactions in the financial market or retail industry, web click streams, video surveillance, and weather or environment monitoring are just a few sources of such data streams. As these kinds of data can not be stored in a data repository, effective and efficient management and online analysis of data streams brings new challenges.

The impetus for this research work came from a joint project with EMSAT Corporation [ems], which specializes in real-time environment monitoring. EMSAT develops solutions applicable to a variety of clients' needs, with a range of types of data and applications. With the aggregation and visualization components of their software already present, they were interested in further preprocessing and knowledge discovery in these data streams, incorporating advanced real-time quality control (QC) and quality assurance (QA) techniques as well as alerting users to unusual events

in real time. This requires techniques for automatically flagging a wider range of anomalies and other events in the data. A special challenge is to perform these tasks on evolving data streams, where the statistical make-up and quality of data can vary widely over the deployment of sensors.

Here, we focus on data streams generated by in-situ environmental sensors: a sequence of time-stamped vectors containing measurements from a group of sensors of several types. Our first goal was to detect spikes, sharp local increases or decreases in the measured values due to noise or other anomalous events. Some such spikes could be detected with simple rule-based approach; other required the structure of nearby data to be taken into account. Subsequently, we extended our techniques to handle more subtle contextual anomalies, which could signal a natural event, noise or sensor malfunction.

As EMSAT works with a variety of clients, we tried to make our approach as general as possible. In particular, we were motivated by the following scenarios suggested to us by EMSAT:

- A weather buoy is recording air and water temperature, current and wind speed, etc. A storm creates a sharp change in the observed data; or a buoy is pulled under water causing atmospheric measurements to become noise; or the surface of the water freezes, making data look quite different.

- A nearby forest fire results in an increase of carbon dioxide level in the vicinity of a sensor; it would be within limits if it were winter, but is unusual in the summer. Additionally, levels of CO_2 increase gradually from year to year, rendering previous thresholds obsolete.

- Sensors installed on machinery such as an industrial drill generate different-looking data under load vs. idle; nevertheless, in either regime events which

might indicate malfunction have to be detected in real time.

As in these scenarios each group of sensors placed together was considered an independent data stream, we did not explore the spatiotemporal approach.

Originally we planned to obtain samples of raw and corresponding label of anomalous data, and use supervised and semi-supervised learning techniques for anomaly detection. However, there was no such labelling of data, which lead us to explore unsupervised techniques. Another major challenge was to tackle the issue of evolving nature of the data streams.

The main focus of the research work was detecting and flagging anomalies in raw environmental sensor data in near real-time. Although some types of anomalies can be detected with simple rule-based techniques, much of the more subtle quality control is still done manually; we were interested in automating as much of this process as possible. At this time, the anomalies are detected manually by the experts who are trained to differentiate between the visual appearance of anomalies and normal data points. However, for the real-time monitoring tasks such anomalies need to be detected by an automated process. Moreover, the streaming scenario demands online processing of data with time delay minimized.

1.2 Our Approach

The major two unsupervised methods for anomaly detection are clustering and statistical methods. For the types of data we were seeing, such as streams of environmental and meteorological sensor measurements (wind speed, temperature, ocean current, etc), statistical methods seemed most appropriate. Moreover, initial interest from EMSAT was using statistical methods as there is such industry standard methods for spike detection in similar problem domain [noa, p. 23]. Some simple algorithms are

present for peak detection in online setting using a window based framework in [Pal09]. Basu and Meckeshmeier proposed a median based framework to clean noise from flight data [BM07]. Subramaniam et al. used kernel density estimation for online outlier detection in sensor data but no explicit change detection was used [SPP+06]. Takeuchi and Yamanishi combined change detection with outlier detection, by reducing change detection to detecting outliers in modified data [TY06]. Bakker et al. proposed a learning framework which combines change detection mechanism with regression models [BPŽ+09]. But the primary goal was not outlier detection. Hill and Minsker used a supervised approach to detect anomalies in environmental data stream which required a significant amount of training data [HM10]. Hayes and Capretz used a framework for contextual anomaly detection for big sensor data using a parametric approach. It also includes an offline component [HC15].

The simplest method is to use a rule based approach to detect noise and other anomalies. Such rules must be developed with close collaboration with users and domain experts. But this approach requires significant expert involvement and it is impossible to develop rules for every real-world scenarios. The most common approach used in various similar domains is based on the assumption of normal distribution. That is, data points which reside at least three standard deviations away from the mean are detected as anomalies. The major drawback of such approach is that it assumes the underlying data distribution is fixed over time.

Considering statistical methods for anomaly detection, we first explored the parametric statistical approach using a Gaussian based model. This technique works well if the underlying distribution fits properly and the distribution is fixed over time. But in case of evolving data stream, the distribution is non-Gaussian and the underlying distribution changes over time due to concept drift or shift. In such cases, the assumptions needed for parametric approach do not apply. And indeed, parametric

approach was not showing good performance on our datasets.

To remedy that, we switched to non-parametric methods such as Kernel Density Estimation (KDE) [Sil86]. Kernel density estimation is attractive because of four reasons: no prior assumptions about the data distribution, initial data for building the model can be discarded after the model is built, it scales up well for multivariate data and is computationally inexpensive [ZMH10]. On our data, KDE has produced much better results (in terms of the number of misclassified points) than the parametric approach.

However, to improve detection of contextual anomalies, it is useful to know when the statistical properties of the data, its context, changes [SG14]. KDE may misclassify points for some time after the change. There is a number of dedicated methods for detecting such changes in evolving data stream [GŽB+14], like Adaptive Windowing (ADWIN), [BG07] one of the most well-known. ADWIN has been incorporated into several predictive and clustering methods, but our goal was to integrate it with non-parametric statistical approaches such as KDE.

The main motivation of our approach was to fine-tune the definition of contextual anomalies in evolving data streams by a well defined context. To achieve that, after initial anomaly detection, we used ADWIN to detect where the change has occurred, and, providing there is enough data between change points, retrain KDE on this more homogeneous stretch of the data stream. Then, some data points can be relabeled more accurately. We have experimented on several real world data sets and the result is validated by industrial experts.

Although change detection inevitably introduces a delay in data processing, if the data is coming fast enough, this is still a viable approach, especially provided that there is a preliminary labeling done in real time. To the best of our knowledge, there is no result present in the literature for the combination of adaptive windowing change

detection and kernel density estimation.

In our future work, we are working on expanding this idea to the setting of outlier ensembles [Agg13c]. We are exploring a variety of directions, from manipulating KDE bandwidth in a sequential model-based ensemble approach, to considering an ensemble of multiple disparate outlier detection and change detection algorithms. And in the longer term, we are proposing to consider more complex anomalies as well as investigating properties of the data which can suggest the techniques most applicable to that setting.

1.3 Thesis Organization

The remainder of the thesis is organized as follows:

- In Chapter 2, we provide an extensive overview of related work.
- In Chapter 3, we describe our approach and discuss the experimental results.
- In Chapter 4, we summarize our work and discuss future research directions.

Chapter 2

Literature Review

2.1 Data Stream Mining

Data stream mining has been an active research area for more than a decade, with sensor data as one of the major sources of data streams. Knowledge discovery from data streams is a broad topic which is covered in several books including [Agg07, Gam10], [LRU14, ch. 4], [Agg15, ch. 12]. Extensive analysis tailored specifically to sensor data can be found in [GGO+08, Agg13a].

In many application domains, a data stream includes a temporal attribute where each data point has either an implicit or an explicit timestamp. Real time sensor data, medical data, mechanical system diagnosis are example of such streams of time-series data. Traditionally it is assumed that time series data can be stored easily and established offline analysis and mining methods can be applied. But in a streaming setting, the focus is shifted towards online data mining. This requirement makes the offline algorithms infeasible, as tasks are expected to be performed in a single pass and in real-time. Here, we are focusing on detecting anomalies in such sensor data streams.

2.2 Anomaly Detection

Anomaly detection is one of the most important areas in data stream mining. This area is also referred to as outlier detection, deviant discovery, fault detection, intrusion detection, or misuse detection [GGAH14]. In some sources, there is a subtle distinction between an outlier and an anomaly, in that outlier refers to an unusual data point or noise, but anomaly is a particular type of outlier which is interesting to an analyst [Agg13b, p. 4]. However, here we will use the terms "outlier" and "anomaly" interchangeably, and only focus on classifying each point. Some well established definitions of outliers are provided in [Gru69, Haw80, BL94]. Anomaly detection is a process to effectively detect anomalies based on the domain specific definition. It is highly unlikely to find a general purpose anomaly detection technique.

Several extensive surveys for anomaly detection are present in the literature [HA04, CBK09, KKZ09]. Some surveys are more focused on a particular domain. Outlier detection methods for wireless sensor networks are covered in [ZMH10, MSME15]. In [CBK12], the topics related to discrete sequences are presented. The research issues of outlier detection for data streams are provided in [SG14].

For temporal/time-series data, a detailed overview is presented in [Fu11, EA12, GGAH14]. An overview of time-series data streams is presented in [Gam10, ch. 11]. A separate comprehensive chapter on outlier detection for time series data is presented in [Agg13b, ch. 8].

It has been identified that an anomaly detection problem has four main aspects [CBK09]. Firstly, the nature of data such as discrete vs continuous. Secondly, based on the availability of data labels, anomaly detection problem can be treated using a supervised or semi-supervised or unsupervised method. Thirdly, anomalies are divided into three types: point, contextual and collective. Recently a new type of anomaly

called contextual collective anomaly has been proposed [JZXL14]. We can identify a data point as a point anomaly if that particular data point is considered anomalous with respect to the rest of the data. In case of contextual anomalies, a data point is considered anomalous only in a particular context such as time. Collective anomalies are a collection of data points which are considered anomalous from the entire data set. Finally, output of an anomaly detection method is generated as scores or labels. In our problem domain, we want to detect mostly contextual anomalies.

On the other hand, anomaly detection methods are divided into several groups like statistical methods, nearest neighbour methods, classification methods, clustering methods, information theoretic methods and spectral decomposition methods [CBK09, ZMH10].

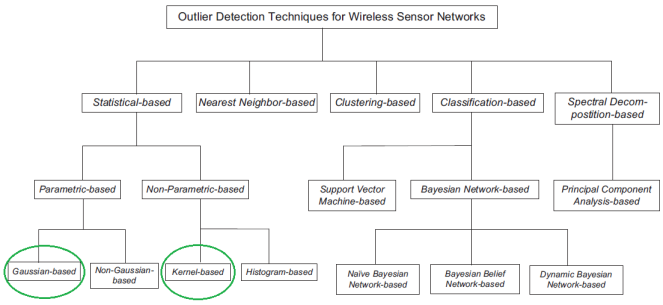


Figure 2.1: Taxonomy of outlier detection in wireless sensor network from [ZMH10], highlighting two methods studied in this thesis.

Out of many anomaly detection techniques, choice largely depends on the problem domain [CBK09, ZMH10, KKZ09]. Each method has its strength and weakness. For example, classification based methods are suitable where a large amount of labeled data is present. Statistical and clustering based methods are useful where the labeled data is unavailable. However, the performance of clustering methods largely depends on the cluster structure. Moreover, anomaly detection is a by-product of clustering. That is, such methods are not optimized for only anomaly detection [CBK09]. Auto

regressive models are useful for time-series data where anomaly detection is deviation based. However, a significant amount of training data is still required.

Statistical methods are the oldest for anomaly detection. The general principle is that normal data are generated by a fixed generative model. These techniques assume that the normal data points reside in high probability region while anomalous points are in the lower probability area of the stochastic model. These are mainly model based methods. Such methods work in two steps: firstly, to learn a generative model fitting the data set and secondly, if the probability of the data point to be generated by this model is very low, the data point is identified as an anomaly. This modelling can be done in an unsupervised way. Due to the nature of the data, the problem domain and the interest of our industry collaborator, statistical methods were most appropriate. There are several advantages of statistical methods [CBK09]: these techniques provide statistically justifiable solution if the assumption about the distribution of data is true, and such methods can work without the need of labeled training data if the distribution estimation is robust.

Recently, the research direction of outlier detection is moving towards "Outlier Ensembles" after the influential paper of the same title by Charu Aggarwal [Agg13c]. In particular, [ZCS14] extend this with a focus on unsupervised methods, whereas [MMA14] leveraged both supervised and unsupervised methods for learning outlier ensembles. In general, the outlier ensembles methods are sequential and independent. Each of them can be further categorized into either data centered or model centered [Agg13c]. There is a great scope of new work in this domain. For example, the sequential model centered ensembles area is still quite open.

2.2.1 Parametric Methods

Parametric methods assume prior knowledge about the underlying distribution of the data. Then the distribution parameters are estimated from the given data. Here, it is assumed that the normal data points are generated from a known distribution with parameter p. After that, it is checked whether a data point x is generated by the estimated distribution provided by the probability density function(pdf) $f(x, p)$. Now, data point x is identified as an anomaly if it resides at the lower probability region [HKP11].

Parametric methods can be further divided into two main subcategories: Gaussian based model and mixture of parametric distributions [CBK09, ZMH10, HKP11]. These methods assume that the underlying data distribution is Gaussian. This is determined using two parameters: mean μ and standard deviation σ. The parameters are estimated using maximum likelihood estimation. The μ and σ is calculated using the following equations:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.1}$$

$$\hat{\sigma} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})} \tag{2.2}$$

A particular data point is declared an anomaly based on its distance from the mean μ. In general, it is calculated under the assumption of normal distribution. That is 99.7% of data resides in $\mu \pm 3\sigma$ regions. However, different approaches may apply various thresholds to determine the anomalies.

Another statistical method using Gaussian model is Grubb's test [Gru69], also called maximum normal residual test [HKP11]. Firstly, for each data x in a data set,

the z score is calculated using the following equation:

$$z = \frac{|x - \hat{\mu}|}{\sigma} \tag{2.3}$$

In the next step, a data point is declared an anomaly if the following equation holds true:

$$z \geq \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N-2+t_{\alpha/(2N),N-2}}} \tag{2.4}$$

Here, N is the total number of objects in the data set, $t^2_{\alpha/(2N),N-2}$ is the threshold by a t-distribution at a significance level of $\alpha/(2N)$. Although the assumption of Gaussian distribution works well in many practical applications, assuming data is generated by a single distribution could miss the actual distribution. This is the case when the underlying data distribution is complex. Considering this problem, it is assumed that data is generated by a mixture of parametric distributions. For example, if there are two Gaussian distributions with parameters $P_1(\mu_1, \sigma_1)$ and $P_2(\mu_2, \sigma_2)$, then the probability that a data point x is generated from the mixture of two distributions is provided by:

$$Pr(x|P_1, P_2) = f_{P_1}(x) + f_{P_2}(x) \tag{2.5}$$

Here, the f_{P_1} and f_{P_2} are the *pdf* of P_1 and P_2. To learn the parameters, Expectation Maximization(EM) algorithm is used. This is similar to the mixture models for clustering. Data point x is identified as an anomaly if it does not belong to any cluster. We have not explored these mixture models for our experiments as we have not used any clustering.

2.2.2 Non-parametric Methods

A non-parametric method tries to learn the model from the given data without any prior assumption on the underlying distribution of the data. It is to be noted that non-parametric does not mean that there are no parameters. Rather, such methods make fewer assumptions about the data and consider that parameters are not fixed before learning the model. A distance measure is defined between a new test instance and the statistical model. Then some thresholds on this distance are used to determine whether the data point is an anomaly. Primarily, the non-parametric approaches are either histogram based methods or Kernel Density Estimation(KDE) [HKP11].

Histograms are specifically effective for density based summarization of univariate data. These methods are the simplest of the non-parametric approaches. The construction of a histogram is intuitive. These methods are implemented using two steps: histogram construction and anomaly detection. In the first step, a histogram is constructed using initial input data. This initial data work like a set of training data. As non-parametric methods are not completely parameter free, some user defined parameters are needed for this construction, in particular number of bins and the size of the bins. But it does not hold any prior assumption about the underlying distribution of data. In the anomaly detection step, a data point is considered an anomaly if it does not belong to any bin of the histogram. There are two major challenges of such approaches. Firstly, it is difficult to pick the optimal bin size. Secondly, it often misses the global nature of the data. We have not explored this area for our experiments due to its limitations.

Kernel Density Estimation(KDE) is used to overcome the limitations of histogram based methods. These methods use a kernel function to estimate the true density. There is similarity between kernel density estimation and histogram regarding the

construction of density profiles. But KDE can construct a smoother density profile than a histogram. A kernel is a non-negative real valued integrable function which satisfies the following conditions:

$$\int_{-\infty}^{+\infty} K(u)du = 1$$

$$K(-u) = K(u)$$

The kernel density estimation of the probability density function(pdf) is following:

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{2.6}$$

Here, $x_1, ..., x_n$ are independent and identically distributed (i.i.d.) sample of a random variable f. $K()$ is the kernel and h is the bandwidth parameter which is the main factor for the smoothing. The selection of different kernel functions is not that important for the accuracy of the density estimation [Sco15, SPP+06]: rather, the single most significant parameter of the kernel density estimation is the bandwidth parameter h. It controls the smoothness or roughness of a density estimate. The bandwidth selection bears the danger of under or over smoothing.

After estimating the probability density function using kernel density estimation, the calculated probability density function is used for anomaly detection. For any data point x, the funtion $\hat{f}_h(x)$ computes the probability whether x is generated from the stochastic process. Data point x is declared an anomaly if the computed probability $\hat{f}_h(x)$ is very low otherwise x is considered normal. There is a variety of kernels for the estimation such as Gaussian, Uniform, Triangular, Epanechnikov, etc. Kernel density estimation is primarily attractive because of four reasons: no prior assumption about the data distribution, initial data for building the model can be discarded

after the model is built, scales up well for multivariate data and computationally inexpensive [ZMH10]. To improve the robustness of the model, sequential ensembles can be used [Agg13b, p. 125].

2.2.3 Anomaly Detection in Environmental Sensor Data Stream

For spike detection in oceanographic data, statistical methods are used as industry standard [noa]. There, underlying normal distribution is assumed and data points are declared as anomalies if they are more than three standard deviations away from the mean. This process is repeated several times, removing detected outliers and recomputing mean and standard deviation after each pass.

Several simple algorithms for peak or spike detection in online setting using a window based framework are described in [Pal09]. The author presented five different algorithms for peak detection in raw time-series data, which work by comparing a data point to its neighbours. Out of this group of algorithms, the entropy based algorithm performed better than the others. However, there was no experiment on real world data set.

A median based approach has been used in [BM07]. The main motivation of this work was cleaning noisy data rather than detecting contextual anomalies. The experiments were done on flight data recorder data, in particular, altitude and roll angle data. This framework is also window based, and the data stream is not considered to be evolving.

In the context of anomaly detection for environmental monitoring data, a variety of ways to construct predictive models from a sensor data stream is presented in [HMA09, HM10]; the authors considered issues specific to the sensor data setting

such as significant amounts of missing data and possible correlation between sensor readings that can help classify a measurement as anomalous. But these are mostly supervised methods and required a significant amount of training data. The real-time anomaly detection using dynamic bayesian networks with Kalman filtering, robust Kalman filtering and Rao-Blackwellized particle filtering is presented in [HMA09]. The results shows that either robust Kalman filtering and Rao-Blackwellized particle filtering outperform the general Kalman filtering. In [HM10], four different types of predictor have been used for one step ahead prediction of the next measurement. Then the predicted value is compared with the actual data for deviation based anomaly detection. Two different modes of the method are presented: anomaly detection and anomaly detection and mitigation. The experiments were done in wind speed data, where January-May data were used for training and only the data of the month of June is used for testing.

A kernel based technique for online anomaly detection in streaming sensor data is presented in [PPKG03]. No prior assumption for known data distribution is required and kernel density estimator is used to approximate the underlying distribution of sensor data. However, the method does not maintain the model when sensor data is updated frequently. The above work is extended in [SPP$^+$06]. Here, a framework is presented to easily extend the kernel density estimation for multivariate setting. Epanechnikov kernel is used for simplicity. Moreover, it is mentioned that the accuracy of the density estimation largely depends on the selection of bandwidth parameter rather than choosing various kernels. But no separate change detector was used to tackle the issue of concept drift.

An online anomaly detection method for sensor system measurements is presented in [YSGG10]. A pairwise linear model is used for representation of data. Then the alignment and similarity measurement are done. Finally, anomaly detection is

performed using a threshold. However, there is no explicit mechanism for change detection.

Another framework on contextual anomaly detection for big sensor data has been presented recently [HC15]. The nature of the problem is closely related with our problem domain, but this work also does not consider the issue of concept drift. The experiments are done on real data sets by powersmiths sensors which generate data from the electrical, water and gas system. The framework has both offline and online components. It generates k clusters and k Gaussian classifiers for each sensor profile. The evaluation of Gaussian classifiers is done online. But the general assumption is Gaussian model based which is a parametric statistical method. This assumption might not work well in case of evolving data stream where the underlying distribution changes over time.

2.3 Change Detection

When a data stream evolves over time, changing the composition and distribution of the data, it becomes useful to note when the changes have occurred, and take this information into account. The task of detecting such change in the data is referred to as "change detection". Different types of change are shown in figure 2.2.

Environmental monitoring data is an example of time-changing sensor data stream. The algorithm that models the underlying processes should be able to identify these changes and update the decision model accordingly. The changes primarily appear as either concept drift or concept shift. The former refers to more gradual change whereas the latter refers to abrupt change. Let us consider two hypothesis: null hypothesis (H_0) and alternative hypothesis (H_1). Hypothesis H_0 states that the previously seen values and current observed values are from the same distribution. Hypothesis H_1

states that the previously seen values and current observed values are from different distributions. For example, let $D_1 = (x_1, x_2, ..., x_m)$ and $D_2 = (x_{m+1}, ..., x_n)$ with $0 < m < n$ represent two samples of instances from a Gaussian-generated stream with means μ_1 and μ_2, respectively. Now the change detection is testing the null hypothesis H_0 with $\mu_1 = \mu_2$ that the two samples are drawn from the same distribution versus the alternate hypothesis H_1 that they arrive from different distributions with $\mu_1 \neq \mu_2$.

There are several methods for change detection described in the literature. [Gam10, ch. 3] and [Agg07, ch. 5] are separate chapters that cover change detection for data streams. Detecting concept drift is more difficult than concept shift. Extensive overview for detecting concept change is provided in [SG09, GŽB+14]. In contrast with anomaly detection, for concept drift detection, two distributions are being compared, rather than comparing a given data point against a model prediction. Here, a sliding window of most recent examples is usually maintained, which is then compared against the learned hypothesis or performance indicators, or even just a previous time window. Much of the difference between the change detection algorithms is in the way the sliding windows of recent examples are maintained and in the types of statistical tests performed (except for Concept-adapting Very Fast Decision Tree (CVFDT) [HSD01]) though some algorithms like ADWIN [BG07] allow different statistical tests to be used. These statistical tests vary from a comparison of means of old and new data, to order statistics [KBDG04], sequential hypothesis testing [MvdBW07], velocity density estimation [Agg03], density test method [SWJR07], Kullback-Leibler (KL) divergence [DKVY06]. Different tests are suitable for different situations; in [DKP11], a comparison of applicability of several of the above mentioned tests is presented.

The following are a sample of algorithms for detecting concept drift. There has been publicly available implementations of some of them: in particular, the Massive Online Analysis (MOA) [moa] software environment for online learning of evolving

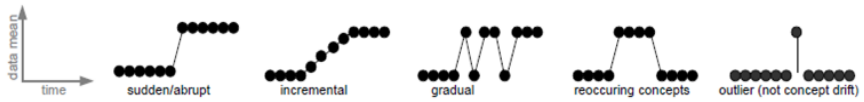


Figure 2.2: Different types of Change [GŽB+14]

data stream incorporates some change detection algorithms.

Probably the oldest algorithm for change detection [Pag54], Cumulative Sum (CUSUM) maintains a mean of (adjusted) examples seen so far: $g_0 = 0$ and $g_t = \max(0, g_{t-1} + (r_t - v))$ in its simplest form (assuming only positive change). Whenever the cumulative sum g_t exceeds a given threshold, a change is detected. A similar idea with a different cumulative variable is used in Page-Hinkley (PH) test.

The CVFDT [HSD01] algorithm is an early algorithm that proposed an incremental approach for building and maintaining a decision tree (Hoeffding tree) in the face of changes or concept drift that occur in a data stream environment. This algorithm does not need an external classifier, checking the incoming data against the decision tree it is maintaining; when that tree does not adequately describe the data, a switch to an alternative tree is made. There is a number of implementations available.

A common theme amongst change detection algorithms is maintaining a sliding window of new or relevant data. Bifet et al. [BG07] proposed an adaptive windowing scheme called ADWIN; the second version ADWIN2 is now available, as well as a version with Kalman filter [BG06]. In ADWIN, the detection of change is based on statistical methods, in particular on the use of the Hoeffding bound. An implementation of ADWIN is available at [adw]. ADWIN and k-ADWIN are incorporated into MOA [moa]. ADWIN is one of the most well known change detection algorithms for data streams. It has been successfully integrated with several learning algorithms in order to deal with concept drift [BG07]. ADWIN keeps a variable length window of recently seen items with a property. The property is that the window has maximal

length statistically consistent with the hypothesis that there is no change in the average value inside the window. Whenever two large enough subwindows exhibit distinct enough averages, the older portion of the window is dropped.

Recently, a faster algorithm has been proposed named OnePassSampler [SPK13]. This algorithm does not do the extensive within-window comparisons of ADWIN, but it uses a sequential hypothesis testing strategy. The statistical test involves computing sample means and using Bernstein bound to estimate the error. It seems to have good performance in terms of false positive/true positive rate, however its detection delay is higher. An extension of this work with reservoir sampling technique has been presented in [PSK14]. Reservoir sampling [Vit85] is a single pass method for obtaining a random sample of a given size from a unknown sized data pool.

2.4 Combined Approaches

Combined approaches take the issue of change detection into consideration while developing a method for anomaly detection. In some of the cases, a single method have been used for both change and anomaly detection.

Two algorithms for outlier detection (considering the issue of change detection) are presented in [AF07]. Between these two, the second one is memory efficient. To tackle the issue of concept drift, the methods support queries at arbitrary points-in-time.

Unified techniques for change point and outlier detection are presented in [TY06, KS09, SZLH13]. Particularly, [TY06] describes unified framework for change and anomaly detection. Here, the change detection problem is reduced to the issue of detecting outliers in the time-series. The time-series model considered is an auto-regressive (AR) model.

Online mass flow prediction from a pilot circulating fluidized bed (CFB) reactor is

presented in [BPŽ+09]. In this problem domain, changes may occur in burning rate, oxygen level and carbon dioxide emission. A further extension for handling outlier and concept drift, a framework is presented in [PBŽ+10]. The authors have proposed a regression learning framework which combines change detection mechanism with regression models. Three different external change detection mechanisms have been used and ADWIN is one of them. The framework presented in [PBŽ+10] detects outliers first and eliminates them. After that, change detection is done for better prediction. The main motivation of this work is not outlier detection rather improving the robustness of online prediction. It is mentioned that the ADWIN was very precise for change detection.

Chapter 3

Our Approach

In our joint project with EMSAT, we primarily focused on environmental monitoring sensor data streams such as air temperature, wind speed, sea surface temperature, etc. The intention was to integrate our anomaly detection algorithms with EMSAT real-time data visualization software, which displays the sequence of the data points as they arrive (including the missing values). The anomaly detection algorithms would additionally highlight, in real time, all unusual data points, leaving it to the user to decide whether they are noise or interesting anomalies. We have run our experiments on raw data from a weather buoy near Newfoundland, maintained by the SmartBay project.

More precisely, our task is as follows. Let $X = \{x_1, x_2, x_3, ..., x_{t-1}, x_t, ...\}$ be a environmental monitoring time-series data stream as a sequence of measurements. The research problem is to decide whether x_{t+1} is a (contextual) anomaly in real-time: ideally, before arrival of x_{t+2}. The user can specify a number of parameters, from minimum/maximum acceptable values to algorithm-specific values such as the size of the sliding window. Note that even threshold values are data type and location-specific: for example, the maximum acceptable wind speed at Wreckhouse (Newfoundland, Canada) is much higher than average.

Our first approach was to use parametric statistical techniques. However, they did not seem to perform that well on our data (see more on that in the section on experimental results). Then, we have moved to kernel density estimation (KDE), which performed better, yet still misclassified some points. In order to improve its performance, we turned to explicit change detection: this allows for a "context" with respect to which a point is labeled as normal or anomalous to be better defined (and KDE trained on a more homogeneous stretch of data). This idea, implemented as a combination of KDE with change detection (ADWIN), is the main contribution of this thesis.

3.1 Combination of Change Detection with Kernel Density Estimation

We want to fine tune the definition of contextual anomalies in evolving data streams by a well defined context. Here, we consider context to be well-defined if there is no change (as could be detected by a change detector) within that stretch of data. In our approach, we first use ADWIN to detect a recent and significant change point in the window of data. If such change is detected, we discard the data before the change point and run KDE on the remaining current data to get a more accurate density estimation. Then we update the labels on the data points within this stretch of context.

If the incoming data point is a missing value, we flag it with appropriate flag value and replace by the mean of the current window. After that, we check whether the incoming data point is valid by comparing it with the user given minimum and maximum acceptable values. If it resides outside the range, that data point is obviously

an anomaly because of erroneous sensing. As a result, it is flagged with appropriate flag value and replaced by the mean. Now, if the data point is within the range, we check the probability of the data point being an anomaly against a threshold. If the data point is a contextual anomaly, it is flagged accordingly, otherwise the data point is flagged as normal. If the incoming data point is flagged as an anomaly, we may also replace it with the mean of the current window. As we get slightly different behaviour in these two cases (see section 3.4.2), we leave the decision to the user. Finally, we let the sliding window grow until it reaches its initial size. After that we run ADWIN again on this window and thus ADWIN runs on the sliding window periodically to compute the change points.

Table 3.1: Flag convention from IOOS manual [ioo]

Flag Short Name	Value
Good data	1
Anomalous data	3
Bad data	4
Missing data	9

There are several user-defined parameters for each stream, including maximum and minimum acceptable values, minimal sliding window size, and sensitivity threshold. The minimal sliding window size N will vary according to a particular data set. Typically it should be large enough to have a decent initial density estimation. The threshold parameter t is usually between 10^{-4} and 10^{-6}. There are also some internal parameters: ADWIN's confidence parameter δ, kernel bandwidth h, change point limit l, fixed number of previous data points from last change point p. The internal parameters are also chosen based on the nature of the data and the application domain.

At the start of execution, the sliding window W will contain the initial N values. ADWIN will run on W, detecting change points. But ADWIN will stop at change

point c where $|x_1...x_c| < N * l$. That is, if we cut $W = \{x_1, x_2, x_3, ...x_c, ...x_t\}$ at point c into to sub-windows then the size of first sub-window W_{prev} must be less than $N * l$, where l is an internal parameter (change point limit). This is done to ensure that the second sub-window W_{cur} will contain enough data so that the KDE can produce a fairly accurate density estimation. Now, data will be discarded from the beginning up to index $c - p$ where p is the fixed number of previous data points from last change point. As the change is sometimes detected with some delay, keeping some previous data from the change point c will not lose any data generated from current distribution. After discarding the data up to $c - p$, W is allowed to grow until $|W| = N$ again. Thus ADWIN will run on W periodically when it will reach the initial window size.

If an incoming data point x_{t+1} is within acceptable range, KDE will run on W_{cur} with respect to x_{t+1} using the following equation.

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3.1}$$

We have used the following Gaussian kernel for our framework:

$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}} \tag{3.2}$$

The bandwidth parameter h was calculated using the Silverman's rule-of-thumb [Sil86, Han09]:

$$h = \hat{\sigma} C_v(k) n^{-1/(2v+1)} \tag{3.3}$$

Here, $\hat{\sigma}$ is the sample standard deviation, v is the order of the kernel and $C_v(k)$ is the constant from the following table:

Table 3.2: Rule of Thumb Constants [Han09]

Kernel	$v = 2$
Gaussian	1.06
Epanechnikov	2.34
Biweight	2.78

This gives bandwidth $h = 1.06\hat{\sigma}n^{-1/5}$ for our Gaussian kernel.

Now, the returned probability of the x_{t+1} being generated from the same distribution will be checked against the threshold t. If the probability is less than t, then it will be flagged as an anomaly, otherwise as normal.

It is observed that if we decrease the value of h, the sensitivity of anomaly detection will also decrease. That is the KDE will be more restrictive. To further verify whether x_{t+1} is an anomaly, we can repeat the same steps again by decreasing the bandwidth. This would provide us with a score of how anomalous the point is. This might be a potential direction to extend the framework for sequential ensembles of outlier detection.

3.2 Data Set

We have used a publicly available data set from the SmartAtlantic Alliance project called SmartBay [sma]. This data set was suggested by EMSAT as it was similar to their working data set, and avoided issues of dealing with proprietary data. The data presented below is from a buoy at the Placentia Bay, Newfoundland. This is a three meters diameter meteorological/oceanographic buoy built by AXYS Environmental Technologies of Sidney, British Columbia. This buoy produces raw data available in near-real time. It has a variety of sensors, measuring average wind speed, peak wind speed, wind direction, air temperature, barometric pressure, dew point, sea surface temperature, maximum wave height and several others. We have used data from August 18th, 2006 to October 16th, 2014. The total number of data points is around 120,000. Each measurement is taken within 20-30 minutes interval. We have used five different types of data sets for our experiments. The visualization of the raw data sets is the following:

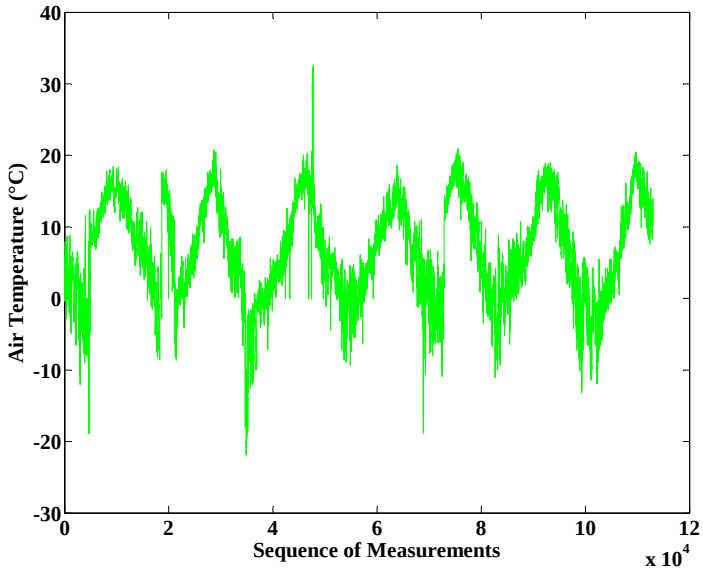


Figure 3.1: Air temperature

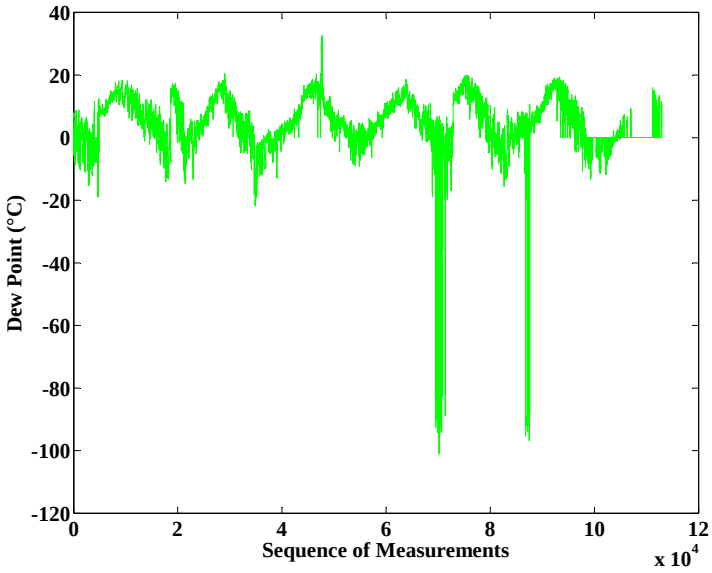


Figure 3.2: Dew point

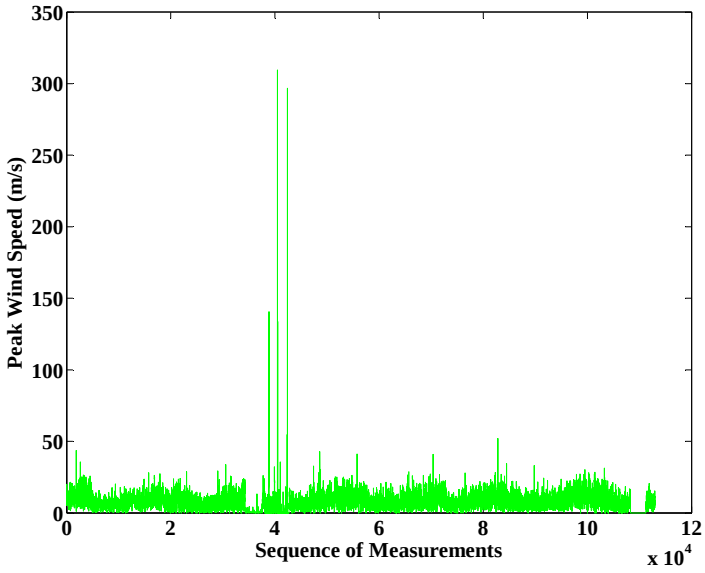


Figure 3.3: Peak wind speed

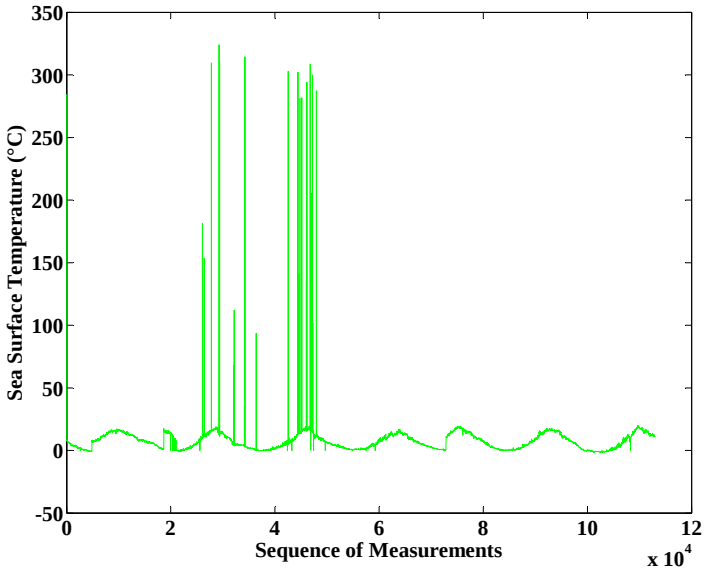


Figure 3.4: Sea surface temperature

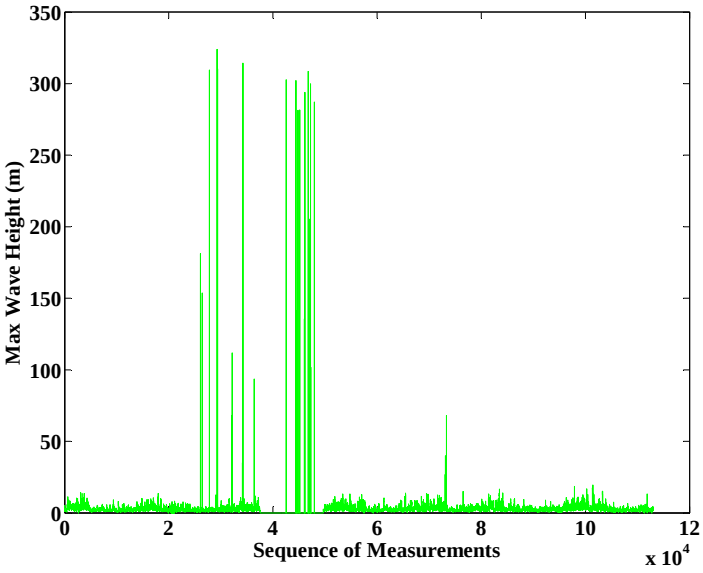


Figure 3.5: Max wave height

3.3 Experiment Design

In all cases, we are using the first N points for our initial density estimation. Thus these points are excluded for anomaly detection. The internal parameters for all cases are: ADWIN's $\delta = 0.03$, change point limit $l = 0.83$, points since last change $p = 70$. We have used Gaussian kernel for the density estimation and the bandwidth h is calculated using Silverman's rule-of-thumb as an optimal choice.

We have used the window size $N = 7000$ and different threshold values t for different data types, in particular for air temperature data $t = 10^{-4}$ and for the dew point, sea surface temperature, maximum wave height and peak wind speed data sets $t = 10^{-5}$.

We have performed all our experiments with the same parameter setting for general KDE and ADWIN+KDE, and considered both discarding and keeping anomalous

Table 3.3: Threshold values for different data set

Data type	Threshold (t)
Air Temperature	10^{-4}
Dew Point	10^{-5}
Sea Surface Temperature	10^{-5}
Peak Wind Speed	10^{-5}
Max Wave Height	10^{-5}

data.

The data sets are available in CSV format. For implementation, we have used Java as the primary programming language. The graphs below were generated using Matlab.

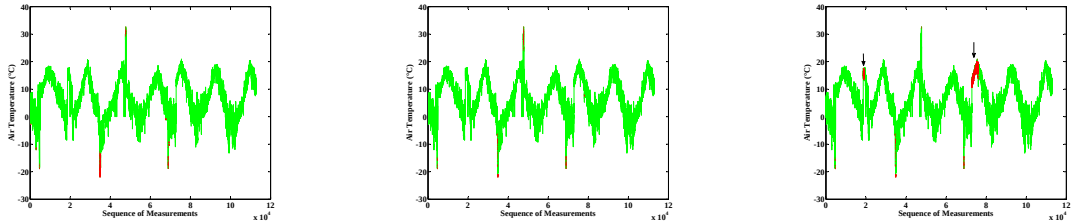
Below, we present some experimental results and comparison of different methods.

3.4 Experimental Results

3.4.1 Comparison of Three Methods: Gaussian, KDE and ADWIN+KDE

Here, we present the experimental results of the above three methods sequentially. These are the results when the anomalous data points are replace with the mean value of the previous window. The Gaussian based method works well when the underlying distribution fits properly. But KDE works better when the underlying distribution is non-Gaussian. Moreover, ADWIN+KDE detects more anomalies in such evolving data streams. Larger figures of these results are presented in the appendix. In all the following figures, the green color represents the raw data where the red dots are used to mark only the anomalous points. Moreover, the black arrows have been used to point the anomalies which are only detected by ADWIN+KDE.

3.4.1.1 Comparison of Air Temperature Anomalies

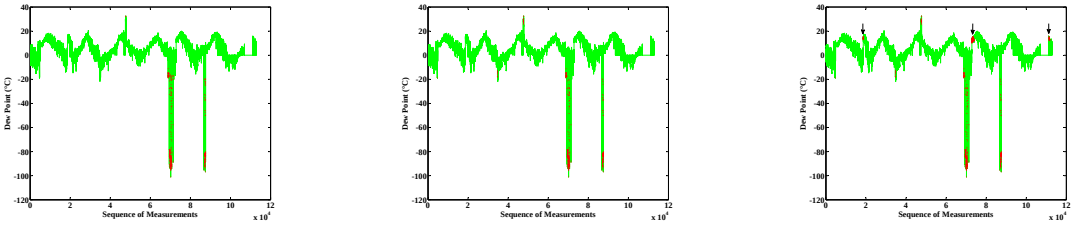


(a) Gaussian based model (b) KDE (c) ADWIN+KDE

Figure 3.6: Anomaly detection in air temperature data

In case of air temperature data set, the Gaussian based model detects most of the spikes. The general kernel density estimation method performs almost similar to the Gaussian based model. Now, the proposed method (ADWIN+KDE) detects two significant anomalous regions where the gradual change of temperature is abnormal. On the other hand, it has correctly detected more anomalies than the general KDE.

3.4.1.2 Comparison of Dew Point Anomalies



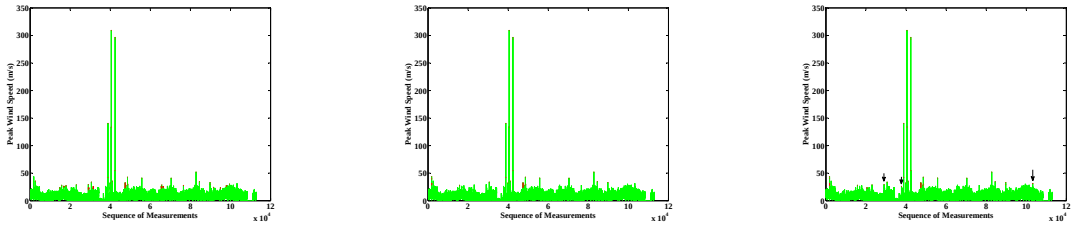
(a) Gaussian based model (b) KDE (c) ADWIN+KDE

Figure 3.7: Anomaly detection in dew point data

In case of dew point data set, the Gaussian based model misses several anomalous points comparing with the detection of the general kernel density estimation method. On the other hand, the proposed method detects three significant anomalous regions where the the change of measurement is unusual. Specially, in the last region, the

measurement suddenly shifts after a fairly flat region. Again, the general purpose KDE fails to detect such events.

3.4.1.3 Comparison of Peak Wind Speed Anomalies

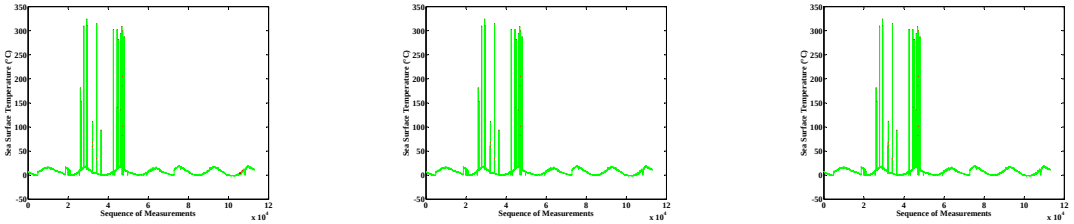


(a) Gaussian based model (b) KDE (c) ADWIN+KDE

Figure 3.8: Anomaly detection in peak wind speed data

In case of peak wind speed data set, the Gaussian based model has a very high false positive rate. Although it detects most of the anomalies but the high false positive rate is not acceptable. Here, the general kernel density estimation methods performs fairly well comparing with the Gaussian based model. Now, the ADWIN+KDE method detects at least three significant anomalous points which were undetected by the previous method. That is, it has correctly detected more anomalies than the general KDE.

3.4.1.4 Comparison of Sea Surface Temperature Anomalies

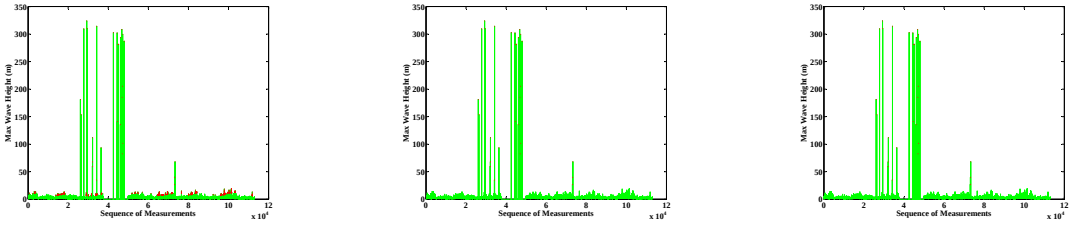


(a) Gaussian based model (b) KDE (c) ADWIN+KDE

Figure 3.9: Anomaly detection in sea surface temperature data

In case of Sea Surface Temperature data set, these three methods perform almost identical and detect most of the anomalies. However the Gaussian based model has some false positives.

3.4.1.5 Comparison of Max Wave Height Anomalies



(a) Gaussian based model (b) KDE (c) ADWIN+KDE

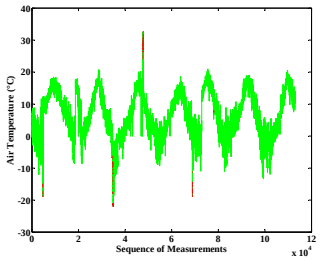
Figure 3.10: Anomaly detection in max wave height data

In case of maximum wave height data set, the Gaussian based model has a very high false positive rate. Although it detects most of the anomalies but the high false positive rate is not acceptable. Here, the general kernel density estimation and the proposed method detect the same anomalies as no significant anomalous region is present in this data set.

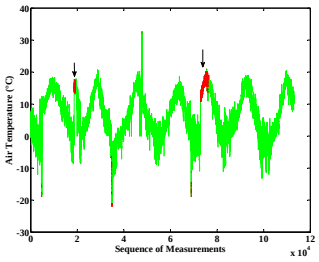
3.4.2 The Effect of Replacing the Anomalous Points with Mean

Here, we present the experimental results of KDE and ADWIN+KDE showing the effect of replacing the anomalous data points with mean. We have done experiments on both replacing with mean value and without replacement. The former detects more points as contextual anomalies whereas the later detects only the starting point of a significant change region as an anomaly. Larger figures of these results are presented in the appendix.

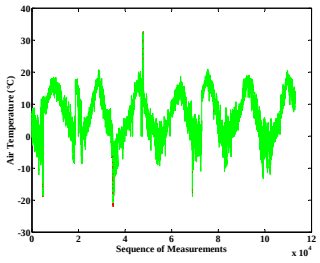
3.4.2.1 Comparison of Air Temperature Anomalies



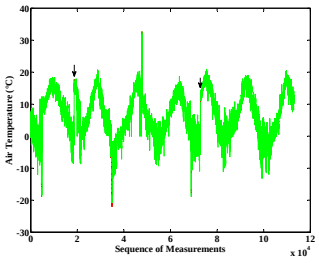
(a) KDE (replacing with mean)



(b) ADWIN+KDE (replacing with mean)



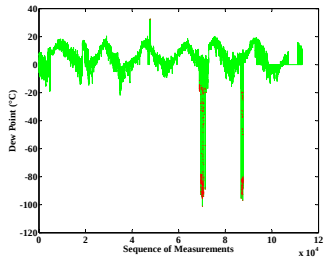
(c) KDE (without replacing with mean)



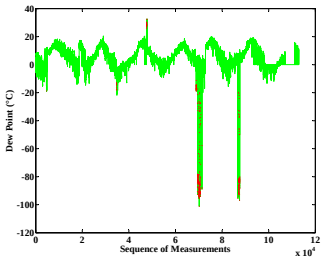
(d) ADWIN+KDE (without replacing with mean)

Figure 3.11: Effects of replacing anomalies with mean in air temperature data

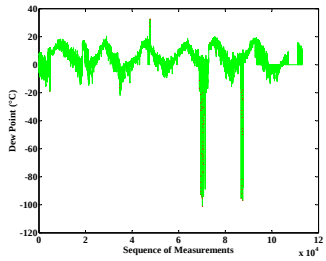
3.4.2.2 Comparison of Dew Point Anomalies



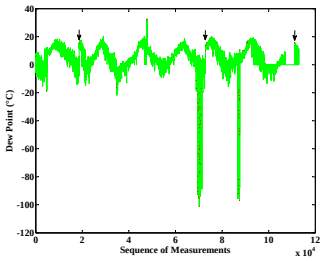
(a) KDE (replacing with mean)



(b) ADWIN+KDE (replacing with mean)



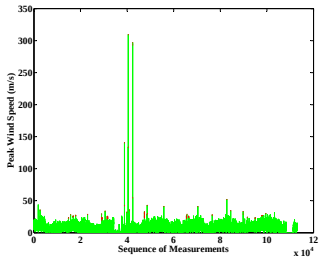
(c) KDE (without replacing with mean)



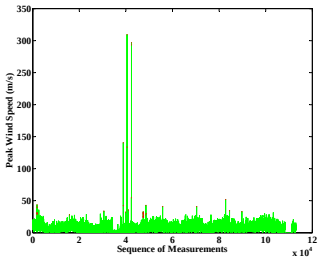
(d) ADWIN+KDE (without replacing with mean)

Figure 3.12: Effects of replacing anomalies with mean in dew point data

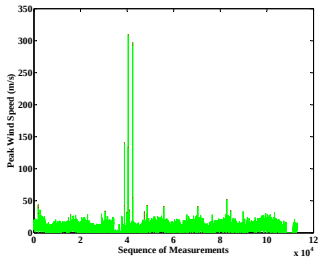
3.4.2.3 Comparison of Peak Wind Speed Anomalies



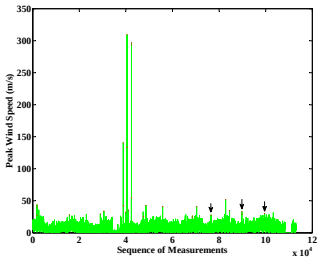
(a) KDE (replacing with mean)



(b) ADWIN+KDE (replacing with mean)



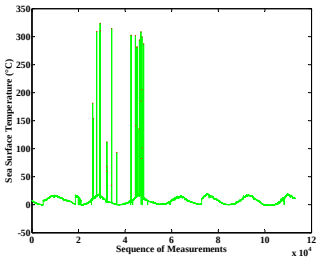
(c) KDE (without replacing with mean)



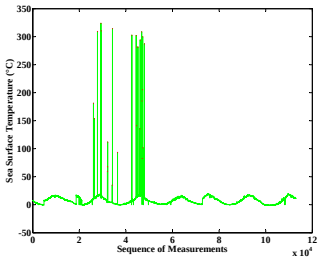
(d) ADWIN+KDE (without replacing with mean)

Figure 3.13: Effects of replacing anomalies with mean in peak wind speed data

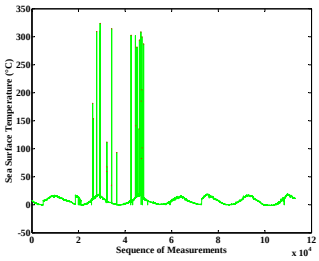
3.4.2.4 Comparison of Sea Surface Temperature Anomalies



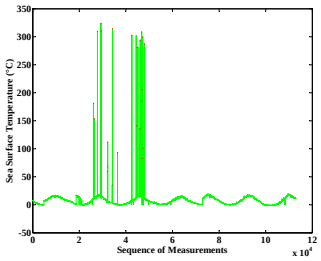
(a) KDE (replacing with mean)



(b) ADWIN+KDE (replacing with mean)



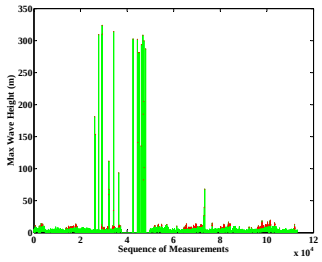
(c) KDE (without replacing with mean)



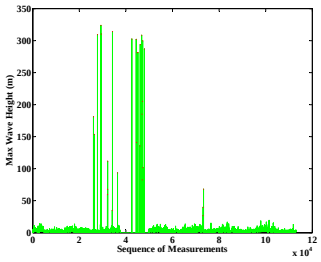
(d) ADWIN+KDE (without replacing with mean)

Figure 3.14: Effects of replacing anomalies with mean sea surface temperature data

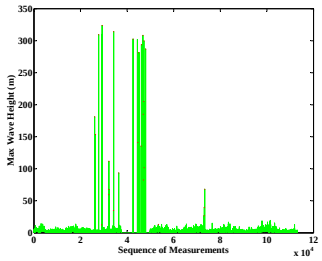
3.4.2.5 Comparison of Max Wave Height Anomalies



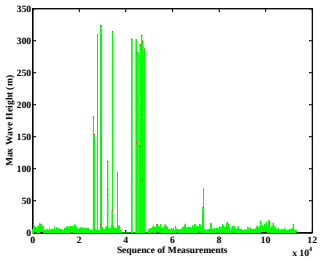
(a) KDE (replacing with mean)



(b) ADWIN+KDE (replacing with mean)



(c) KDE (without replacing with mean)



(d) ADWIN+KDE (without replacing with mean)

Figure 3.15: Effects of replacing anomalies with mean in max wave height data

3.4.3 Summary of the Experimental Result

It seems that combining kernel density estimation with an external change detector does lead to detection of more anomalies on some data sets. Moreover, the proposed framework detects several unusual anomalous regions which are undetected by the other methods. However, we have seen some data sets where anomaly detection of kernel density estimation did not change significantly with introduction of a separate change detector. The reason behind such outcome is that such data sets do not contain any significant anomalous event. However, these experiments suggest that an external change detection mechanism enhances the results of kernel density estimation anomaly detection when applied to such natural time-changing environmental data stream.

We show the results for both settings when anomalous data points are replaced with mean (similar to outliers being discarded in the initial processing on some buoys). In this case, more continuous data points are marked as contextual anomalies until the change is stable. On the other hand, only the starting point of the significant change region is detected anomalous when we do not replace anomalous points with the mean value. Analysts might choose a particular approach based on their interest and the nature of the application domain.

Chapter 4

Conclusion

4.1 Summary

Motivated by a specific problem coming from a real-world application for EMSAT real-time environmental monitoring, we have explored statistical techniques and their combination with change detection for unsupervised anomaly detection in environmental sensor data streams. In general, KDE performed better than parametric methods, and combination of ADWIN and KDE was able to detect possible events of interest that KDE by itself could not detect. EMSAT has found these results promising, and plans to incorporate these techniques into their product in the near future.

4.2 Future Work

There are many possible directions of research and other applications of this approach. The framework requires a large-scale sensitivity analysis of its parameters. In short term, we are interested in creating ensembles of anomaly detection techniques and change detection, and evaluating their performance on environmental sensor data. We plan to include both variants of the same technique with differing parameters (for

example, KDE with different kernels and/or bandwidth), and a range of different techniques. Exploring ways to address challenges specific to multivariate data is another part of our work in progress.

Another direction is to incorporate detection of others, more complex types of anomalies. In addition to better detection of collective anomalies, we would like to investigate detecting discords, unusual patterns in the data streams. This would depend crucially on the types of data we would have access to, as we expect different types of data to have very different structure with respect to frequent/unusual pattern occurrences.

Overall, for a longer term project, we would like to understand what properties of data streams and anomaly definition make certain techniques or classes of techniques more applicable. Our current work with statistical techniques and change detection already shows that anomaly detection on some data sets benefits from adding change detection, while for others KDE by itself detects anomalies just as well. Analyzing performance of ensembles may shed more light on such differences between types of data and anomalies.

Bibliography

[adw] Adwin source. `http://adaptive-mining.sourceforge.net`.

[AF07] Fabrizio Angiulli and Fabio Fassetti. Detecting distance-based outliers in streams of data. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 811–820. ACM, 2007.

[Agg03] Charu C Aggarwal. A framework for diagnosing changes in evolving data streams. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 575–586. ACM, 2003.

[Agg07] Charu C Aggarwal. *Data streams: models and algorithms*, volume 31. Springer, 2007.

[Agg13a] Charu C Aggarwal. *Managing and mining sensor data*. Springer Science & Business Media, 2013.

[Agg13b] Charu C Aggarwal. *Outlier analysis*. Springer Science & Business Media, 2013.

[Agg13c] Charu C Aggarwal. Outlier ensembles: position paper. *ACM SIGKDD Explorations Newsletter*, 14(2):49–58, 2013.

[Agg15] Charu C Aggarwal. An introduction to data mining. In *Data Mining*, pages 1–26. Springer, 2015.

[BG06] Albert Bifet and Ricard Gavalda. Kalman filters and adaptive windows for learning in data streams. In *Discovery Science*, pages 29–40. Springer, 2006.

[BG07] Albert Bifet and Ricard Gavalda. Learning from time-changing data with adaptive windowing. In *SDM*, volume 7, page 2007. SIAM, 2007.

[BL94] Vic Barnett and Toby Lewis. *Outliers in statistical data*, volume 3. Wiley New York, 1994.

[BM07] Sabyasachi Basu and Martin Meckesheimer. Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems*, 11(2):137–154, 2007.

[BPŽ+09] Jorn Bakker, Mykola Pechenizkiy, I Žliobaitė, Andriy Ivannikov, and Tommi Kärkkäinen. Handling outliers and concept drift in online mass flow prediction in cfb boilers. In *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data*, pages 13–22. ACM, 2009.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[CBK12] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection for discrete sequences: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 24(5):823–839, 2012.

[DKP11] Tamraparni Dasu, Shankar Krishnan, and Gina Maria Pomann. Robustness of change detection algorithms. In *Advances in Intelligent Data Analysis X*, pages 125–137. Springer, 2011.

[DKVY06] Tamraparni Dasu, Shankar Krishnan, Suresh Venkatasubramanian, and Ke Yi. An information-theoretic approach to detecting changes in multi-dimensional data streams. In *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*, 2006.

[EA12] Philippe Esling and Carlos Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012.

[ems] Emsat corporation. `http://www.emsatcorp.com`.

[Fu11] Tak-chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[Gam10] João Gama. *Knowledge Discovery from Data Streams*. Chapman and Hall / CRC Data Mining and Knowledge Discovery Series. CRC Press, 2010.

[GGAH14] Manish Gupta, Jing Gao, Charu Aggarwal, and Jiawei Han. Outlier detection for temporal data. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 5(1):1–129, 2014.

[GGO+08] Auroop R Ganguly, Joao Gama, Olufemi A Omitaomu, Mohamed Gaber, and Ranga Raju Vatsavai. *Knowledge discovery from sensor data*. CRC Press, 2008.

[Gru69] Frank E Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[GŽB+14] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014.

[HA04] Victoria J Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[Han09] Bruce E Hansen. Lecture notes on nonparametrics. *Lecture notes*, 2009.

[Haw80] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[HC15] Michael A Hayes and Miriam AM Capretz. Contextual anomaly detection framework for big sensor data. *Journal of Big Data*, 2(1):1–22, 2015.

[HKP11] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[HM10] David J Hill and Barbara S Minsker. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling & Software*, 25(9):1014–1022, 2010.

[HMA09] David J Hill, Barbara S Minsker, and Eyal Amir. Real-time bayesian anomaly detection in streaming environmental data. *Water resources research*, 45(4), 2009.

[HSD01] Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106. ACM, 2001.

[ioo] Manual for the use of real-time oceanographic data quality control flags. http://www.ioos.noaa.gov/qartod/temperature_salinity/qartod_oceanographic_data_quality_manual.pdf.

[JZXL14] Yexi Jiang, Chunqiu Zeng, Jian Xu, and Tao Li. Real time contextual collective anomaly detection over multiple data streams. *Proceedings of the ODD*, pages 23–30, 2014.

[KBDG04] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 180–191, 2004.

[KKZ09] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Outlier detection techniques. In *Tutorial at the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2009.

[KS09] Yoshinobu Kawahara and Masashi Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *SDM*, volume 9, pages 389–400. SIAM, 2009.

[LRU14] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[MMA14] Barbora Micenková, Brian McWilliams, and Ira Assent. Learning outlier ensembles: The best of both worlds–supervised and unsupervised. 2014.

[moa] Massive online analysis. http://moa.cms.waikato.ac.nz.

[MSME15] Dylan McDonald, Stewart Sanchez, Sanjay Madria, and Fikret Ercal. A survey of methods for finding outliers in wireless sensor networks. *Journal of Network and Systems Management*, 23(1):163–182, 2015.

[MvdBW07] S Muthukrishnan, Eric van den Berg, and Yihua Wu. Sequential change detection on data streams. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 551–550. IEEE, 2007.

[noa] Handbook of automated data quality control checks and procedures. `http://www.ndbc.noaa.gov/NDBCHandbookofAutomatedDataQualityControl2009.pdf`.

[Pag54] ES Page. Continuous inspection schemes. *Biometrika*, pages 100–115, 1954.

[Pal09] G Palshikar. Simple algorithms for peak detection in time-series. In *Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence*, 2009.

[PBŽ+10] Mykola Pechenizkiy, Jorn Bakker, I Žliobaitė, Andriy Ivannikov, and Tommi Kärkkäinen. Online mass flow prediction in cfb boilers with explicit detection of sudden concept drift. *ACM SIGKDD Explorations Newsletter*, 11(2):109–116, 2010.

[PPKG03] Themistoklis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. Distributed deviation detection in sensor networks. *ACM SIGMOD Record*, 32(4):77–82, 2003.

[PSK14] Russel Pears, Sripirakas Sakthithasan, and Yun Sing Koh. Detecting concept change in dynamic data streams. *Machine Learning*, 97(3):259–293, 2014.

[Sco15] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[SG09] Raquel Sebastiao and Joao Gama. A study on change detection methods. In *4th Portuguese Conf. on Artificial Intelligence, Lisbon*, 2009.

[SG14] Shiblee Sadik and Le Gruenwald. Research issues in outlier detection for data streams. *ACM SIGKDD Explorations Newsletter*, 15(1):33–40, 2014.

[Sil86] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[sma] Smartatlantic alliance. `http://www.smartatlantic.ca/Home`.

[SPK13] Sripirakas Sakthithasan, Russel Pears, and Yun Sing Koh. One pass concept change detection for data streams. In *Advances in Knowledge Discovery and Data Mining*, pages 461–472. Springer, 2013.

[SPP+06] Sharmila Subramaniam, Themis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd international conference on Very large data bases*, pages 187–198. VLDB Endowment, 2006.

[SWJR07] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Statistical change detection for multi-dimensional data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 667–676. ACM, 2007.

[SZLH13] Wei-xing Su, Yun-long Zhu, Fang Liu, and Kun-yuan Hu. On-line outlier and change point detection for time series. *Journal of Central South University*, 20:114–122, 2013.

[TY06] Jun-ichi Takeuchi and Kenji Yamanishi. A unifying framework for detecting outliers and change points from time series. *Knowledge and Data Engineering, IEEE Transactions on*, 18(4):482–492, 2006.

[Vit85] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

[YSGG10] Yuan Yao, Abhishek Sharma, Leana Golubchik, and Ramesh Govindan. Online anomaly detection for sensor systems: A simple and efficient approach. *Performance Evaluation*, 67(11):1059–1075, 2010.

[ZCS14] Arthur Zimek, Ricardo JGB Campello, and Jörg Sander. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):11–22, 2014.

[ZMH10] Yang Zhang, Nirvana Meratnia, and Paul Havinga. Outlier detection techniques for wireless sensor networks: A survey. *Communications Surveys & Tutorials, IEEE*, 12(2):159–170, 2010.

Appendix A

Appendix

A.1 Air Temperature Anomalies

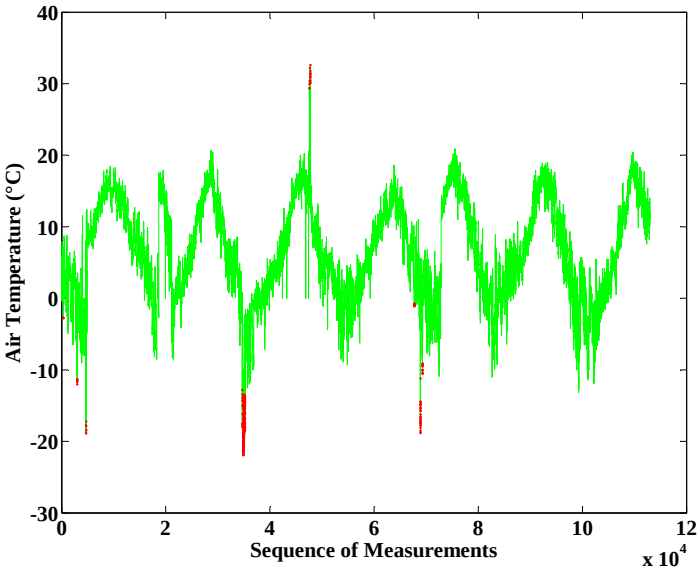


Figure A.1: Air temperature anomalies detected by Gaussian based model

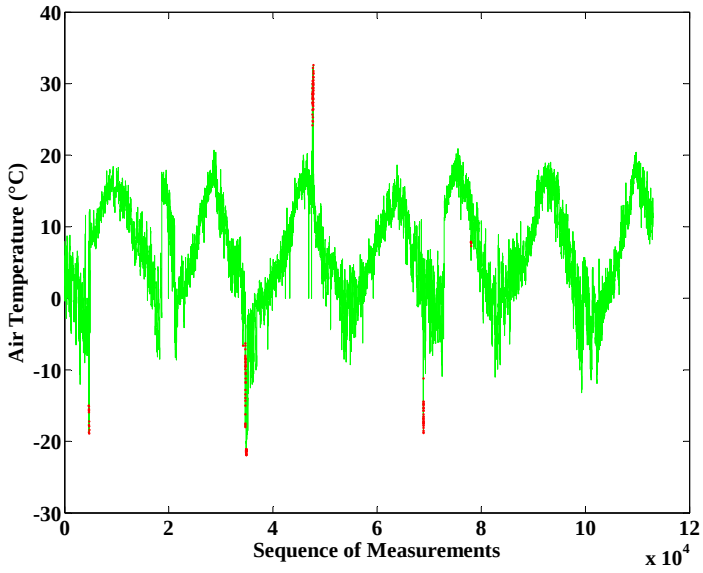


Figure A.2: Air temperature anomalies detected by KDE (replacing with mean)

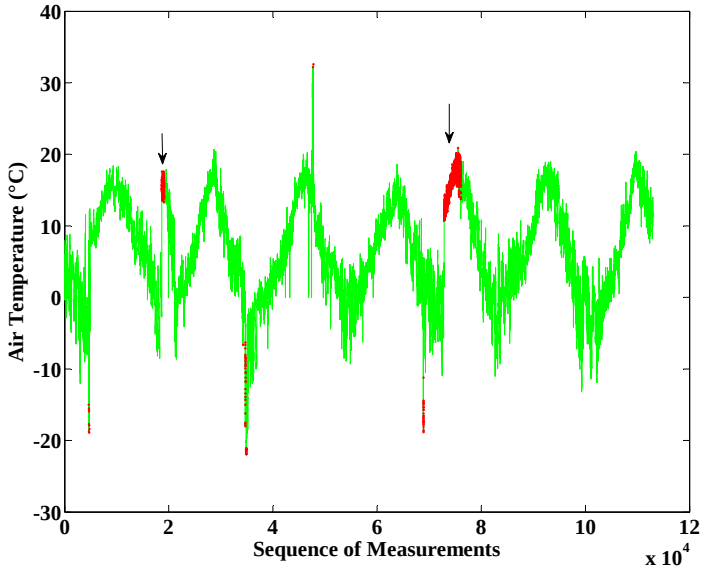


Figure A.3: Air temperature anomalies detected by ADWIN+KDE (replacing with mean)

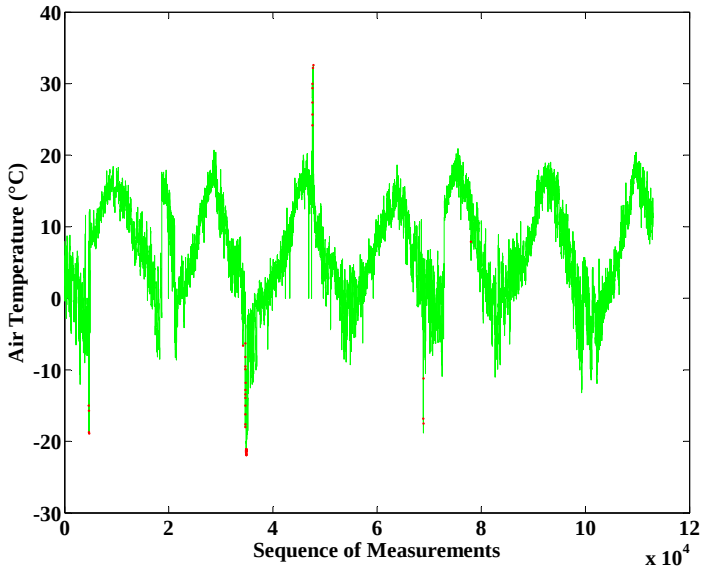


Figure A.4: Air temperature anomalies detected by KDE (without replacing with mean)

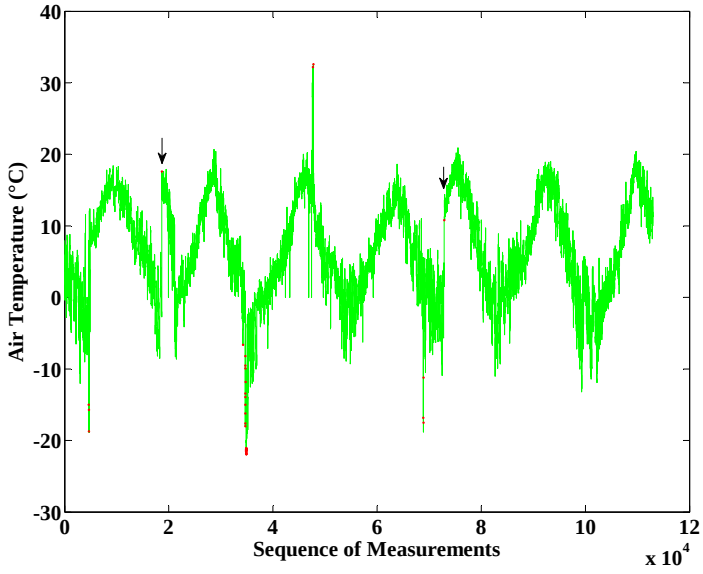


Figure A.5: Air temperature anomalies detected by ADWIN+KDE (without replacing with mean)

A.2 Dew Point Anomalies

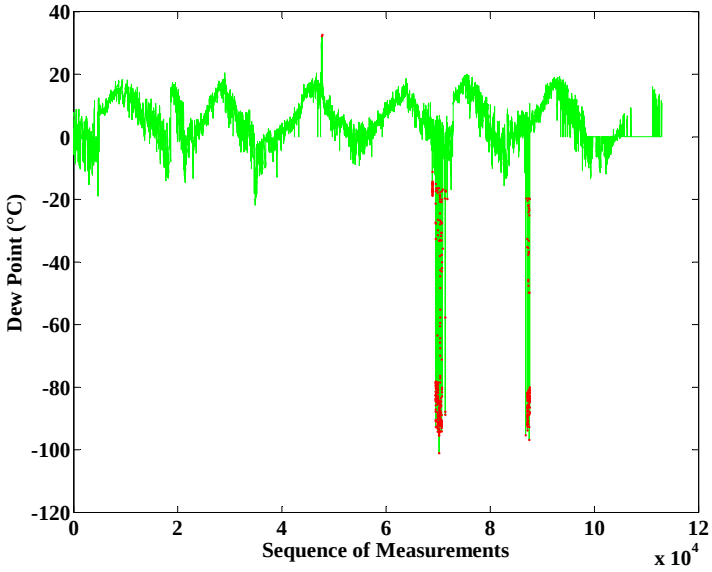


Figure A.6: Dew point anomalies detected by Gaussian based model

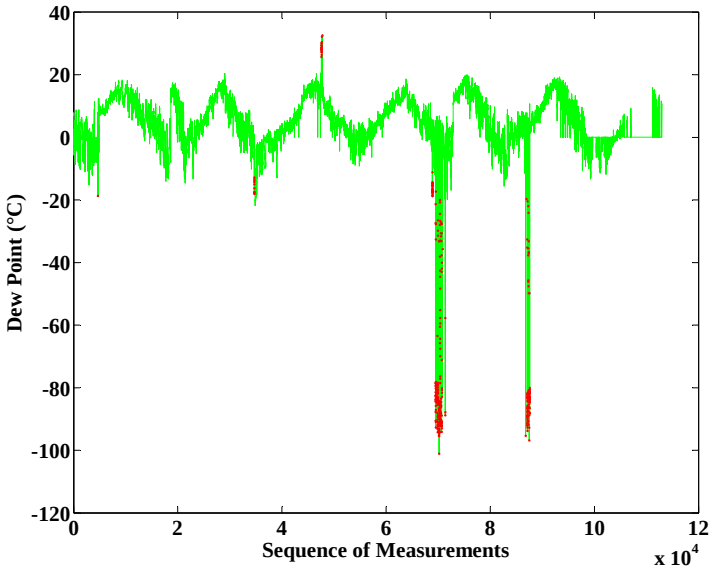


Figure A.7: Dew point anomalies detected by KDE (replacing with mean)

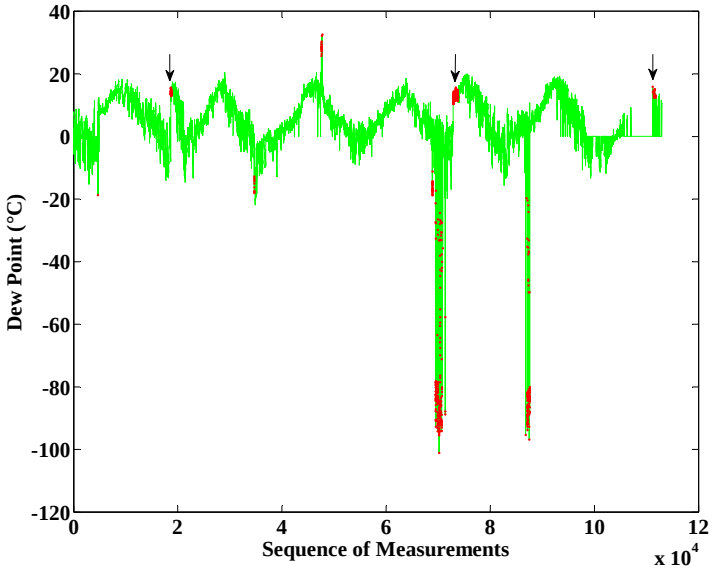


Figure A.8: Dew point anomalies detected by ADWIN+KDE (replacing with mean)

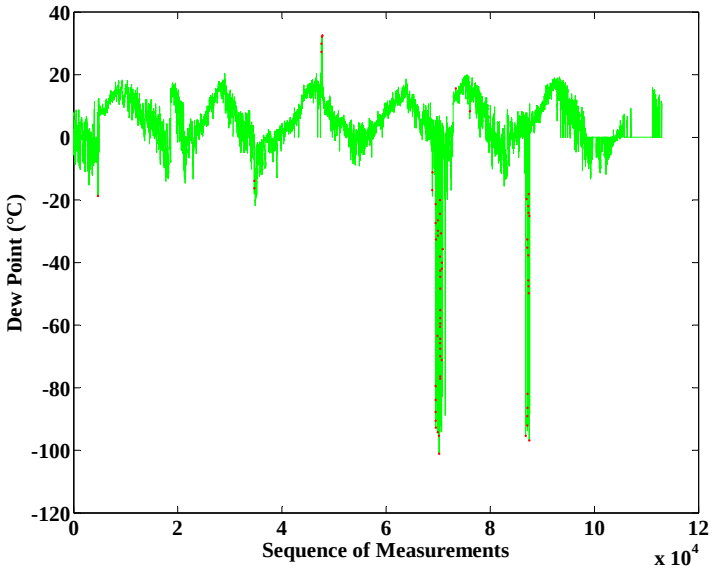


Figure A.9: Dew point anomalies detected by KDE (without replacing with mean)

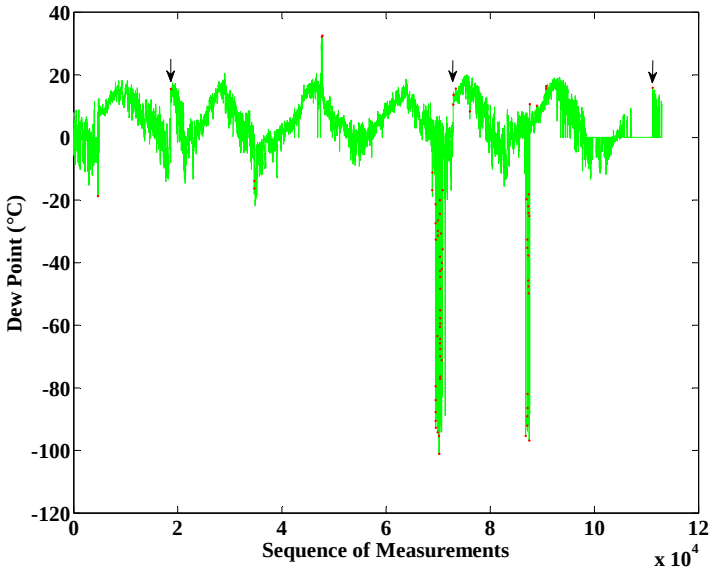


Figure A.10: Dew point anomalies detected by ADWIN+KDE (without replacing with mean)

A.3 Peak Wind Speed Anomalies

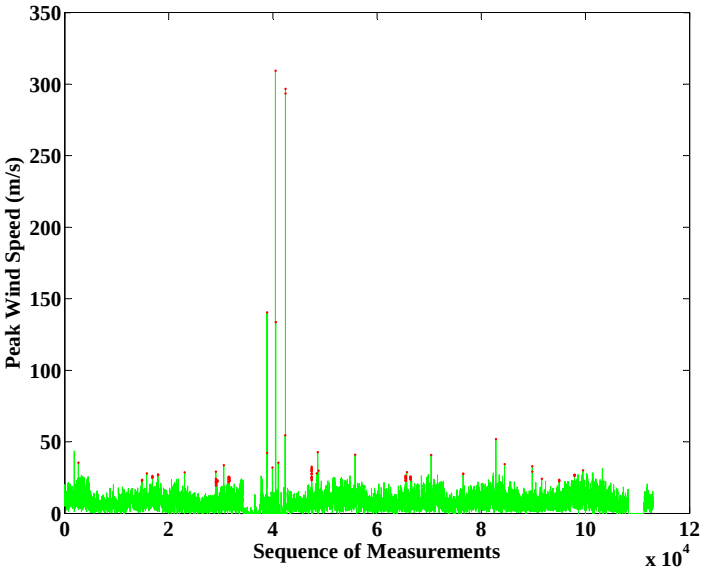


Figure A.11: Peak wind speed anomalies detected by Gaussian based model

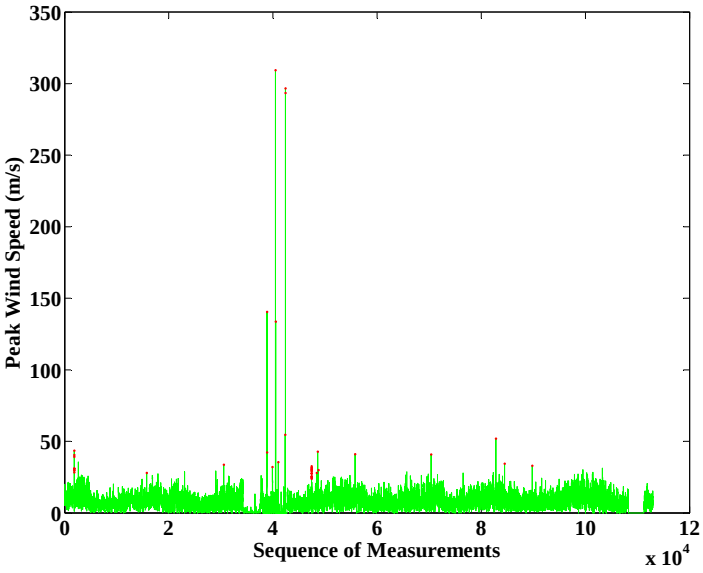


Figure A.12: Peak wind speed anomalies detected by KDE (replacing with mean)

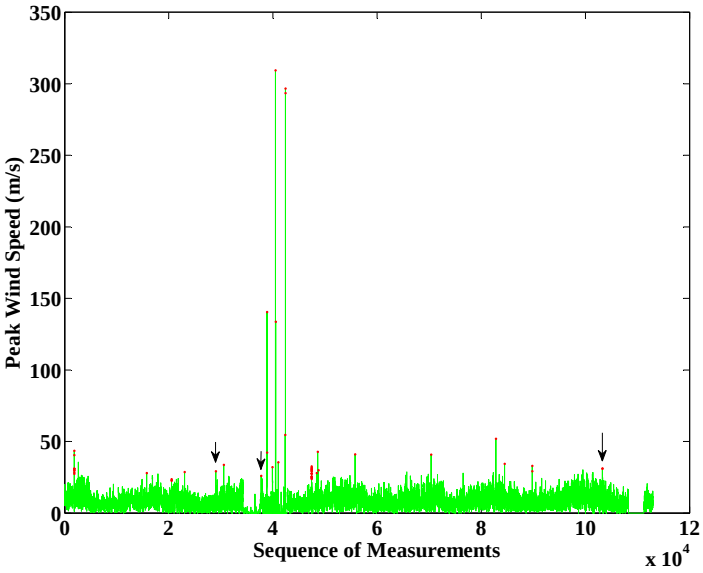


Figure A.13: Peak wind speed anomalies detected by ADWIN+KDE (replacing with mean)

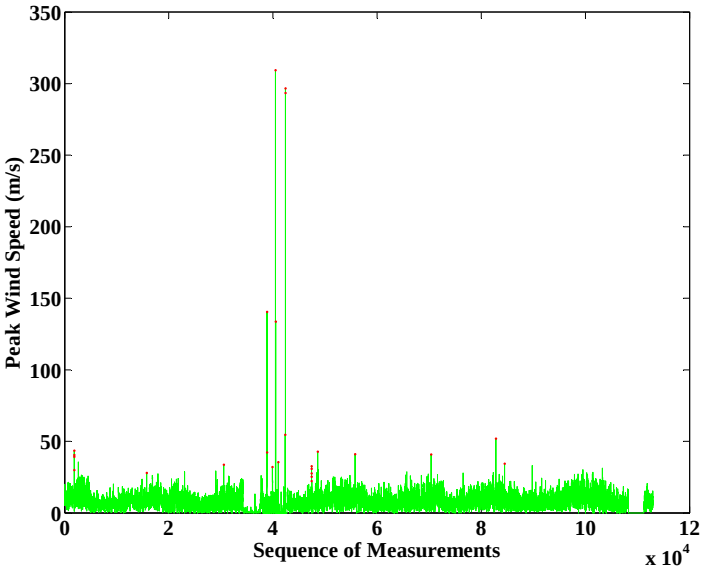


Figure A.14: Peak wind speed anomalies detected by KDE (without replacing with mean)

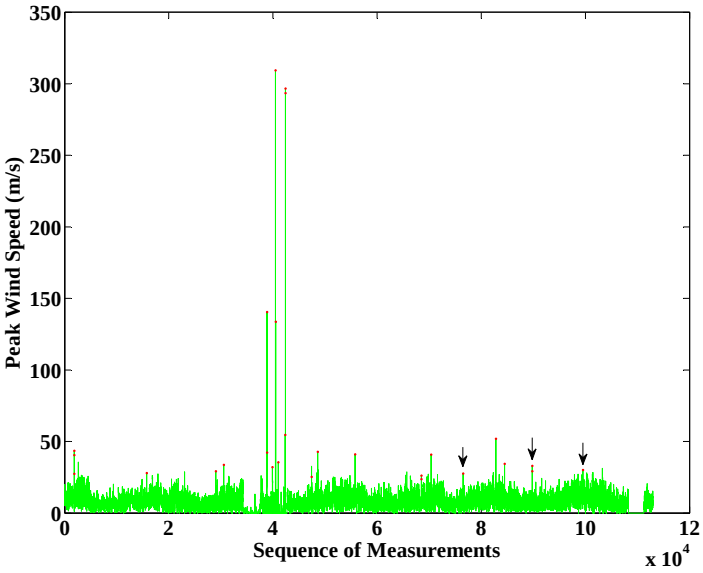


Figure A.15: Peak wind speed anomalies detected by ADWIN+KDE (without replacing with mean)

A.4 Sea Surface Temperature Anomalies

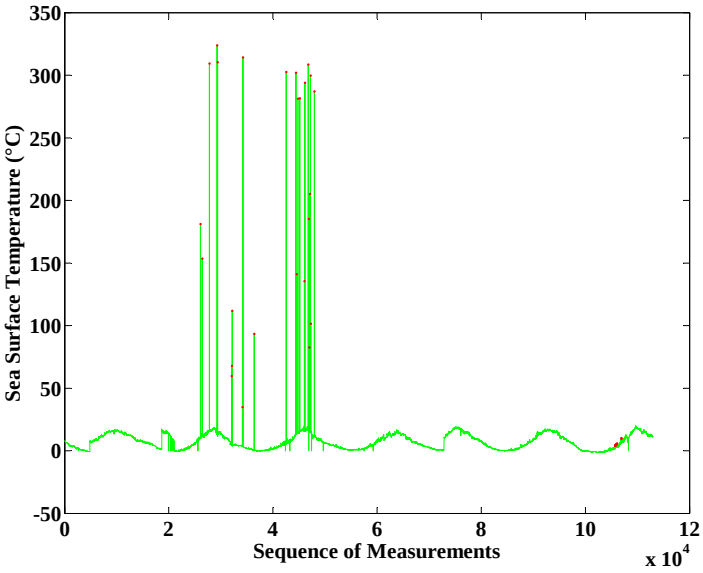


Figure A.16: Sea surface temperature anomalies detected by Gaussian based model

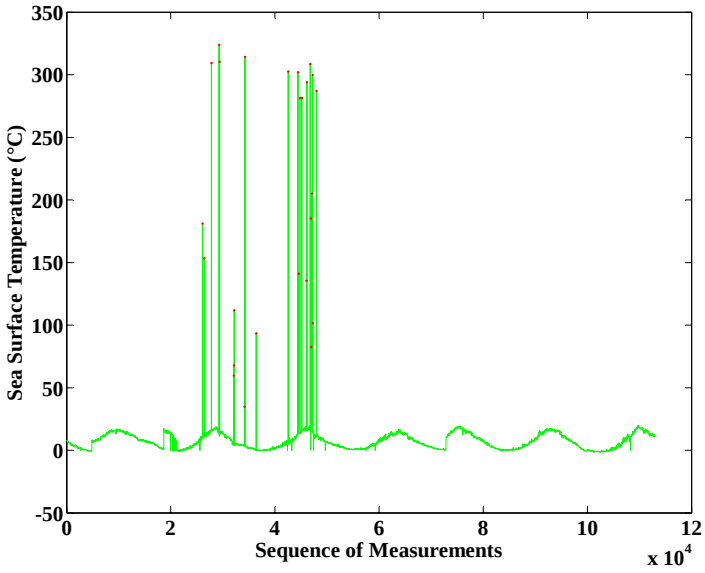


Figure A.17: Sea surface temperature anomalies detected by KDE (replacing with mean)

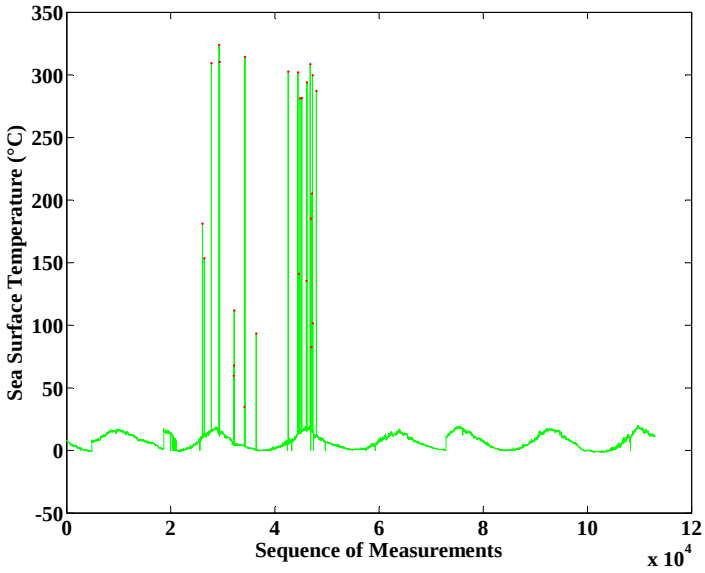


Figure A.18: Sea surface temperature anomalies detected by ADWIN+KDE (replacing with mean)

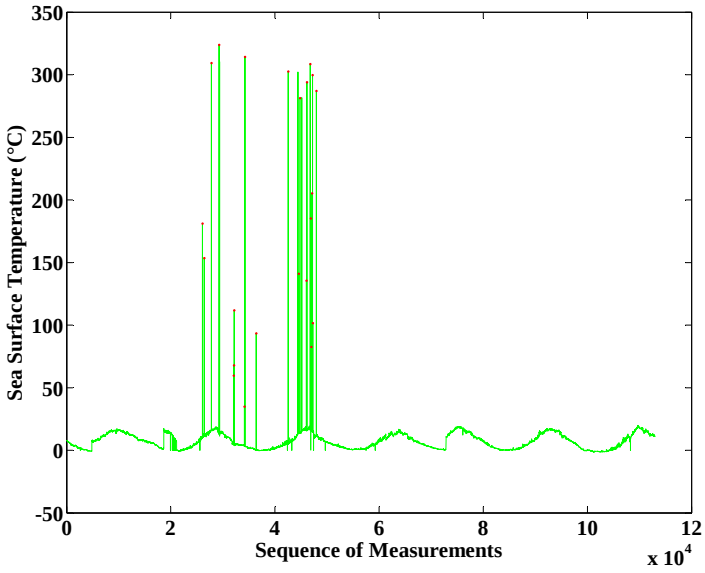


Figure A.19: Sea surface temperature anomalies detected by KDE (without replacing with mean)

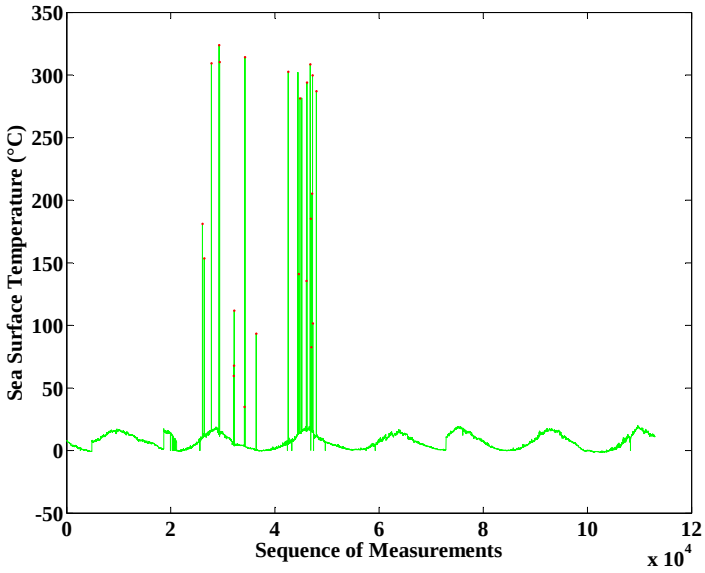


Figure A.20: Sea surface temperature anomalies detected by ADWIN+KDE (without replacing with mean)

A.5 Max Wave Height Anomalies

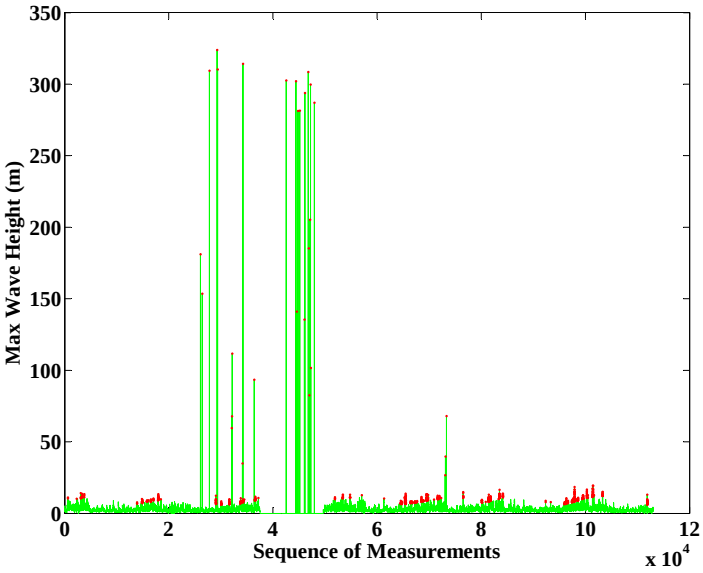


Figure A.21: Max wave height anomalies detected by Gaussian based model

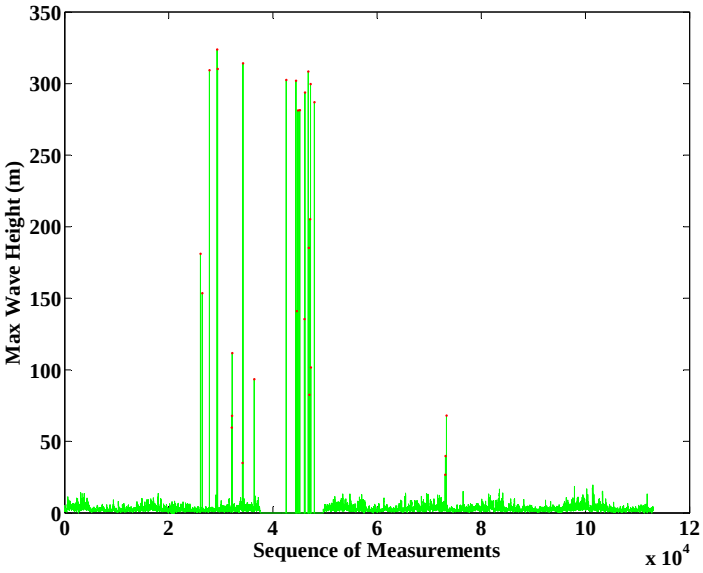


Figure A.22: Max wave height anomalies detected by KDE (replacing with mean)

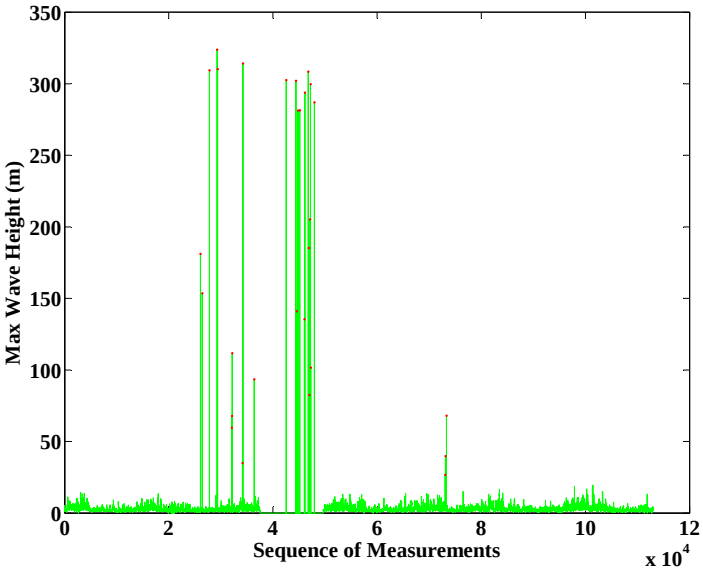


Figure A.23: Max wave height anomalies detected by ADWIN+KDE (replacing with mean)

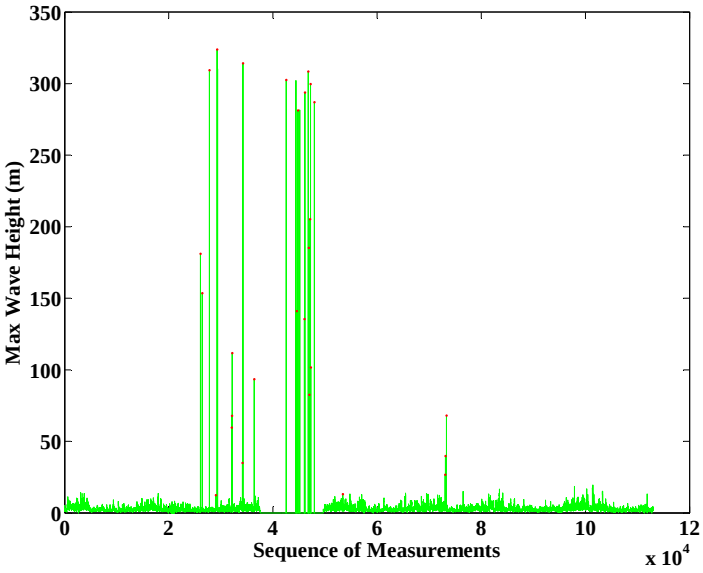


Figure A.24: Max wave height anomalies detected by KDE (without replacing with mean)

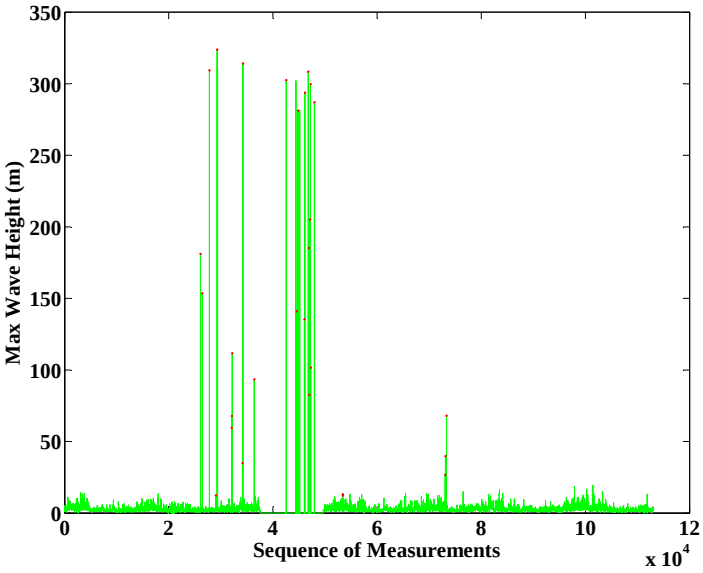


Figure A.25: Max wave height anomalies detected by ADWIN+KDE (without replacing with mean)