A Big Data Change Detection System

Carsten Lanquillon[1] and Sigurd Schacht[2]

Abstract: Detecting changes in data streams is a crucial issue in many real-world applications. Big data scenarios entail new challenges. We describe suitable change detection techniques for big data streams, focusing on robust control charts for either univariate or multivariate observations of properties derived from data streams. Our generic big data change detection system integrates common big data technology on the one hand and proven change detection approaches on the other hand based on a modification of the Lambda architecture in order to enable scalable real-time change detection for arbitrary data streams.

Keywords: Big Data, Change Detection, Lambda Architecture, Robust Control Charts.

1 Introduction

In many business application areas change detection is a vital issue in identifying crucial problems. In *compliance monitoring*, e.g., continuous controls monitoring systems typically scan business transactions for non-compliant activities [CDL07, FP99]. From a technical perspective, non-compliant activities are shifts in data streams or gradual drifts in comparison to a defined process description. In *risk management*, change detection allows an early and active monitoring of previously identified risk points to ensure failure-free business operations and enable management to react in time [Ap02, Kl00].

In big data, often the combination of different data sources [LSW07] such as internal trans-action data with external data like tweets unleashes new potential for business value. For example, if a company combines their internal product quality data with information about the public opinion about their products, an early warning system for customer satisfaction can be established. Once again, change detection is the foundation for identifying shifts in customer perception. The combination of both sources may help to identify the root cause of fulminating changes.

Furthermore, the digital transformation of businesses entails extensive automation of business processes and decisions. To achieve this, various forms of analytics for unlocking insights from the underlying data are being applied. Many data analysis algorithms and models react unexpectedly if the underlying data streams change.

[1] Heilbronn University, Business Information Systems, Max-Planck-Str. 39, D-74081 Heilbronn, Germany, carsten.lanquillon@hs-heilbronn.de

[2] Heilbronn University, Business Information Systems, Max-Planck-Str. 39, D-74081 Heilbronn, Germany, sigurd.schacht@hs-heilbronn.de

To identify erroneous results and avoid misinterpretation [Ki04] and, also, to keep processes of interest in control, changes in the relevant data streams have to be identified in an early stage.

Apparently, in all application areas, a common pattern emerges: Identifying relevant properties that significantly deviate from a state of normality or from our expectation and issuing alerts allows businesses to react appropriately and, thus, to avert damages and to enable future opportunities. In the face of big data, how can we build a change detection system that efficiently scales with high volume and high velocity data streams and copes well with inherent variety and veracity?

2 Change Detection Challenges

When talking about change, we assume differences in the state of an object or process over time or space. Change detection aims at indicating whether any data generating process associated with the objects or processes of interest have changed or are currently changing [TGS14]. First, we briefly review the core challenges of detecting changes in data streams. Then, we discuss how change detection with big data adds further challenges.

2.1 Core Change Detection Challenges

From a statistical perspective, data streams resemble sequences of random variables which naturally have inherent variation. Thus, the key challenge in change detection is to distinguish between normal variation caused by stochastic fluctuations, typically referred to as noise, and variation due to true changes. In statistical process control, this is known as distinguishing between *chance causes* and *assignable causes* of variation [Mo09].

Especially if a data stream is noisy, it may contain inconsistent data points with unusually small or large values, so-called *outliers*. By definition, outliers are spurious and may never appear again. So, regarding an outlier as change is considered a *false alarm*. Note, however, that extreme data points which seem to be outliers may also herald change [LMZ06]. Consequently, there will always be a trade-off between reducing the risk of false alarms and minimizing change detection latency.

The underlying data generating processes may change for various reasons. Irrespective of these reasons, we may characterize changes by the rate at which they occur. Commonly, we distinguish gradual changes (*drift*) from abrupt changes (*shift*). The rate at which changes occur strongly affects the ability to detect them [Mo09]. Generally, drifts are more difficult to identify, especially at an early stage.

2.2 Big Data Challenges

Big data is commonly characterized by the so-called three Vs: *volume*, *velocity* and *variety* [Sc12]. In addition, some people add Vs for *veracity* to stress possible uncertainty and likely lack of trust in the data and, also, to emphasize that, eventually, business relevant *value* ought to be created by means of analytics, e.g. see [GH15].

Volume: Big data is mostly associated with a huge volume of data. Owing to the pure amount of data and its continuous creation, analyzing all data at the same time is infeasible. Data stream analysis permits the analysis of this data to react in a timely manner focusing on individual observations [Sc12]. In change detection, however, often properties derived from samples or batches are compared to properties on reference data to identify possible differences [TGS14]. One big challenge with respect to big data is the trade-off between analysis speed and providing enough information to identify possible changes.

Velocity: Data is not only created in high speed. Also, the usefulness of data and analytic results tends to decrease as the latency between data creation and analysis increases. Often, data is not even stored between creation and analysis [Sc12]. As this prohibits later re- scans of data sets, change has to be detected on the fly reading data only once.

Variety: Analyst are faced with many different data types–structured, semi-structured or unstructured. To detect changes in data of various types, appropriate techniques are to be used [Sc12]. Hence, a generic change detection system ought to be flexible enough to support various change detection techniques.

Veracity: Data quality is crucial for successful data analysis. Especially in big data settings, data quality issues are severe due to inconsistencies, incompleteness and the inherent uncertainty of the data [Lu14]. Thus, a change detection system must be robust enough to cope with incomplete, dirty and noisy data.

Value: In the long run, any economically driven big data endeavor has to create or enhance value for a business, e.g. by minimizing risks, increasing sales, decreasing costs or increasing product quality. A value driven big data project must be based on a good question, which has to be answered. Understanding the business domain and defining the right questions to be answered by the big data projects is crucial to deliver business value.

To summarize, high-volume and high-velocity big data streams mainly impact the required efficiency of a change detection framework which can be achieved by scalable approaches. Further, the veracity property calls for robust change detection approaches. Finally, in the face of big data, an analyst should be pointed at relevant attributes and issues by a change detection system to overcome information overload.

3 Change Detection Techniques

Change detection has a long history in statistical process control. In particular, variants of so-called *control charts* are broadly applied. First, we distinguish the two phases commonly considered in real-world change detection applications of statistical process control and translate them to the big data setting. Next, we provide a brief overview of common control charts with a focus on those which suit the requirements of big data scenarios as described above. We continue our overview on change detection techniques by briefly introducing common processing setups. Finally, action points of how to derive properties (statistics) based on which changes in data streams are to be detected are covered.

3.1 Process Control Overview

Getting and keeping a process of interest in control is the key objective of process monitoring and control. In the field of statistical process control, a variety of tools to support this has been developed. Among those, surely the control chart is the most dominant tool to check whether a process is in control or out of control [Mo09].

In the statistical process control, two phases are distinguished: a *retrospective phase I* and a *prospective* or *monitoring phase II* [Wo00]. In phase I, historical data, also known as reference data, is used to explore a process so as to examine whether it was in control. If a process is not yet in control, as an important part of phase I, actions to get it in control are to be triggered. Once a process is statistically in control, a model that describes this assumed *state of normality* is derived from the reference data or defined based on expert knowledge or known specifications [Wo00]. In phase II, control charts based on the models of normality of phase I are used in a continuous monitoring process to check whether the underlying process is still in control. Significant deviations from normality are regarded as hints for changes. This may be regarded as a sequence of hypothesis tests with the null hypothesis being that the underlying process is in control [Wo00].

In big data applications, we assume that there are many data generating processes which are very likely to change over time. As such, they may be investigated like other processes with tools from statistical process control to detect changes. Yet, we have to be aware of a crucial difference: In most big data scenarios, controlling or adapting the data generating processes so as to get them into a state of control as described above is often not possible. Experience in data quality management shows that even internal data sources are hard to improve without appropriate management support. For external big data sources, which often tend to dirty and noisy according to the veracity property, this is even more true. So, instead of checking whether the data generation process was in control during an initial retrospective phase, we simply assume that any data available in this phase resembles the normality of the process under consideration. Any models generated during the initial phase should describe this state well. To achieve this, using *robust methods* for the estimation of any control chart parameters is a crucial issue.

Note, that in data analysis parlance, we distinguish between a modeling phase and a model deployment phase. While modeling is conducted on historic data, model deployment applies the models created to new data of the application at hand. The steps of the modeling phase have their equivalents in the activities carried out in the retrospective phase I of statistical process control. Taking aside data quality management initiatives, however, there is typically no attempt of trying to get a process in a state of control. Finally, the deployment phase strongly matches the phase II in statistical process control.

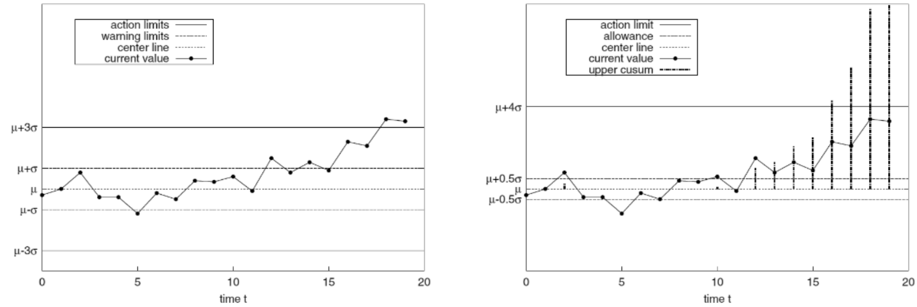

Fig. 1: Illustration of a Shewhart control chart (left) and a cusum control charts (right).

3.2 Control Chart Variants

Walter A. Shewhart developed the statistical control chart concept as an ongoing change detection method (monitoring) for product quality [Mo09]. In a nutshell, a control chart plots relevant process properties over time and checks whether they significantly deviate from an expected level. In order to do so, a control chart uses so-called *control limits* which are typically set some multiples of a suitable measure of spread around the expected level for the property being observed. Hence, defining appropriate process properties and deriving statistically sound parameters to define the control limits are key challenges.

The famous family of *Shewhart control charts* contains various control charts for different continuous process properties (variables) such as the sample mean or range as well as discrete properties (attributes) such as the fraction of non-conforming items in a sample. This leads to well-known variants such as the $\bar{x}$-chart, R-chart or p-chart [Mo09].

Assume that we observe a process property v_t over time t with expected value (mean) μ and standard deviation σ. Appropriate action points (properties) to be observed in big data scenarios will be described below. In general, the values vt are plotted in time-series fashion with appropriate warning and action limits indicating possible process change. Figure 1 (left) illustrates the Shewhart control chart. The action limits are equivalent to the control limits mentioned above and are suitable to detect reasonably large changes (shifts) in data streams.

A well-known drawback of Shewhart control charts is that small and persistent changes (drifts) often remain undetected since at any time t only the current value of the process property is taken into account. Rather than looking only at the current property value at time t, the *cumulative sum (cusum) control chart* considers the entire sequence of property values by accumulating their deviations from the expected value over time. Hence, it is more suitable to detect small, successive changes [Mo09]. Figure 1 (right) shows a cusum control chart with an upper cusum statistic. To benefit from the individual strengths in detecting shifts and drifts and, thus, to enhance change detection ability, Shewhart and cusum charts may also be used in combination [Mo09].

Univariate control charts are broadly applied in many real-world applications. Yet, some relevant types of changes may only be detected when looking at several properties. Of course, these could be monitored separately based on univariate control charts. In case interactions among these properties are important, though, jointly observing them in a *multivariate control chart* is mandatory to have the ability to identify all relevant changes. The *Hotelling's* T^2 control chart is the multivariate extension of the Shewhart control chart. For details on the control limit construction as well as on multivariate variants of the cusum control charts, see[BPP07] for example.

Multivariate control charts may work well for a small number of process properties to be observed jointly. As number of these properties increases, however, the *curse of dimensionality* is most likely to strike and change detection will lose efficiency [Mo09]. In these situations, applying dimensionality reduction or projection methods like *principle component analysis (PCA)* or *partial least squares (PLS)* might be helpful. Change detection will then take place in the reduced feature space [BPP07, KX].

Another way to look at multivariate data is to set up context-dependent or conditional control chart. Instead of comparing process properties to global control limits, specific control limits may be established for certain subsets of the data stream generated by the underlying process under observation [Gr04, Cu07]. In OLAP terminology, this corresponds to providing control charts for each cell of a multi-dimensional data cube.

Standard control charts are commonly set up based on the assumption that the process is initially in control, i.e. the process is *stationary* [Mo09]. As we have seen above, however, in big data change detection scenarios, the underlying data-generating processes are not necessarily in a state of control. Often, the initial state is taken as it is and deviations from that state are to be detected. Furthermore, many real-world data streams are dirty and noisy and, therefore, missing values and outliers are likely to deteriorate parameter estimation for the control charts.

To remedy these issues, robust estimation of control chart parameters should be considered, e.g. [Ro89, AO08, Ho13]. Commonly, outlier-sensitive parameters such as the mean and standard deviation are replaced by more robust parameters like any trimmed estimates or the median and the interquartile range (IQR). Note, however, that robust control charts are typically less sensitive in detecting change. After all, some outliers may actually be the starting points of change. As discussed below, *data quality properties* should be monitored in addition to the regular process properties to cope with this issue.

3.3 Stream Processing Setup

Especially depending on the nature of the process property chosen to detect changes, the stream processing setup will vary. We distinguish three approaches on how to process observations in data streams: *individual observations*, *batches* of observations and the *sliding window*. A very straight forward approach to change detection is to treat each observation individually when plotting control charts [Ku08, ZZW10]. This may actually be considered as the extreme case of batch processing with batch size $n = 1$. Depending on the calculations required, this may be computationally more expensive than processing larger batches of observations. Especially in big data scenarios with high volume and velocity data streams, this approach may prove unfeasible.

Nevertheless, some approaches such as those for detecting outliers require individual processing of data points: Outlier detection does not make sense on batch aggregates. But still, it takes more than one extreme observation in a data stream to indicate change. Therefore, it is often preferred to consider data in batches of several data points rather than treating each observation individually. Still, note that it is possible to aggregate outlier statistics such as the fraction of outliers per batch. In statistical process control, most commonly batches of equal size n are constructed [BPP07]. When looking at data generating processes, however, it seems quite natural to segment data streams into batches according to discrete time slots such as hours or days. The time slot length should be chosen with respect to the volume and speed of the data arriving and the target upper bound for the change detection latency. Eventually, this results in batches with varying batch sizes n_t over time t. As these batches correspond to natural time slots, however, they are easy to interpret in real-world applications. Also, they simplify the detection of changes in data volume over time.

Note that although segmenting data streams by time is the most common choice, it is also possible to segment over location in addition to or instead of time. In the following, we assume that data is split into batches by time. In addition, batch processing of observations will also be helpful in situations where the process property does not naturally follow a normal distribution which is typically assumed for many standard control charts. If the average batch size is large enough, i.e. $n_t \geq 30$, a normal distribution may be used as an approximation to the unknown distribution of the process property under observation.

Approaches based on a sliding window behave like those processing batches of observations where the batch to be inspected for change contains the n most recent observations [Ki04]. As the inspection of this batch of observations may be conducted after the arrival of each new observation, detection delay is expected to be smaller for regular batch processing.

3.4 Action Points

Above, we have given a rather abstract view on how change detection by means of control charts works. In practice, it is crucial to define appropriate process properties based on which changes are to be detected. To determine possible properties, we take the data life cycle according to a general analysis process model into account. Hence, change is to be detected either based on characteristics which may be derived from the data directly (data properties) or based on properties derived from further processing steps such as model properties or model deployment properties.

Note that any property might not only be monitored by the appropriate variant of the Shew-hart control chart but instead or in addition also by cusum or other kinds of control charts. It is also possible to observe several parameters that describe an underlying distribution of a process property. This naturally leads to multivariate control charts.

Data Properties

Data properties are closest to the underlying data generating process and, thus, deserve a very close look. We will distinguish between content-based and diagnostic properties.

Using *content-based process data properties* is the most common approach. When processing observations individually, relevant values of the underlying process may be used as they are. In batch processing, statistics like the mean or a measure of spread like the range may be calculated. This leads to the well-known $\bar{x}$-chart and R-chart which are instances of variables control charts. For quality characteristics, p-charts for monitoring the fraction of non-conforming items is a common choice [Mo09].

Using *diagnostic* or *data quality properties* leads to a very important change detection enhancement especially in big data scenarios. The data quality literature provides are great number of different quality criteria and corresponding measures or properties, e.g. see [Sa13]. Any data quality property that can be evaluated on a batch of observations or individual observations and which are relevant for an application at hand could be evaluated. Among them the fraction of missing values and the fraction of outliers in a batch of observations are surely the most important properties. In addition, any other plausibility check on the values can be used to define nonconformity. All of these values can be easily monitored a simple p-chart.

Model Properties

Instead of directly looking at data properties, it is also possible to compare models or patterns derived from the reference data of the retrospective phase to models derived from current data. If the models significantly differ, change in the underlying data generating process is suspected. Depending on the type of model, we have to provide appropriate model comparison methods, e.g. see [Wo04, Bö06].

Model Deployment Properties

Instead of comparing models learned at different points in time, another approach to deriving process properties is to look at model deployment properties. When deploying predictive models, a straightforward approach is to assess prediction performance such as classification accuracy or the sum of squared errors. Yet, the calculations of these measures require the true target values to be known. In most situations, however, this not possible at all or in a timely manner as the prediction problem would not have existed otherwise. In- stead, unsupervised evaluation should be considered. For classification problems, we may use the average confidence of the decisions or the fraction of observations for which the decision should rather be rejected due to a lack of confidence.

4 System Architecture

Our main objective is to build a change detection system for monitoring arbitrary data integrated in a big data ecosystem from various sources and applications. In addition, the system should be easily extensible with new change detection techniques in simple plug and play fashion. This was achieved by creating standardized interfaces for change detection techniques considering the features described above.

In building the system, the aim was to make use of and benefit from existing big data technologies as much as possible. Consequently, our system is designed based on a modification of the so-called *Lambda architecture* for building scalable real-time information systems [MW15]. In the following we first describe the key components of our change detection system. Finally, we explain how these components fit into the fundamental components (layers) of the modified Lambda architecture, namely batch layer, serving layer and speed layer.

4.1 Change Detection Components

The *data collector* component provides a unified access to the data of interest. While some data sources already provide streams of data being generated like Twitter or general news feeds, special extractors may have to implement for other sources.

The *modeling engine* provides user-friendly access to the reference data to generate control charts. For the standard control charts with robust parameter estimates, simple statistical procedures may suffice. In some situations, however, more sophisticated learning algorithms may be required. In addition, interactive visualizations and analysis capabilities should help the users to gain insight of the underlying processes. This is equivalent to the phase II in statistical process control. The results of the modeling engine are implementations of available control chart interfaces.

The *monitoring engine* accepts the implementations of available control chart interfaces from the modeling engine. In particular, the interfaces allow the specification of control chart parameters regarding how to organize the stream processing, how to compute process properties from the data items available and how to setup the control limits. The monitoring engine works together with the *streaming engine* to actually compute the process properties based on the data processing specifications. The resulting values are then fed into the appropriately set up control charts. Web access to chart visualizations and generated change alerts are offered to end users.

4.2 Modified Lambda Architecture

Since any query can be expressed as a function on all data, the goal of an information system is simply the computation of arbitrary functions on arbitrary data. The Lambda architecture describes how to build scalable real-time information system based on this finding [MW15]. Below we describe how the change detection components fit into the Lambda architecture to yield our big data change detection system as shown in Figure 2.

Modified Batch and Serving Layer

The batch layer provides fault-tolerant and robust capabilities to store all relevant data. In addition, scalability is achieved by means of parallel processing frameworks. As this layer allows the computation of arbitrary functions on all the available data, we fit the modeling engine into this layer. Instead of any pre-computed result sets in the form of batch views, however, we generate implementations of control chart interfaces. Instead of providing an SQL-like interface for user access, our serving layer feeds the control charts into the monitoring engine.

Modified Speed Layer

Owing to the batch processing latency, the batch layer alone will not meet real-time requirements. The speed layer regards the most current data not yet available in any batch views as data streams based on which real-time views are computed. Finally, combining the matching batch and real-time views and eliminating duplicate results will yield scalable real-time computation [MW15]. Rather than computing real-time views for the corresponding batch views, however, the modified speed layer takes over the deployment phase or phase II in statistical process control terms. Hence, the streaming engine for the on-line calculation of property values on data streams and the charting and alarming functionality of the monitoring engine are placed here. From an architecture point of view, it could be argued whether offering web access to the resulting control chart showing the current data and possible change alerts should be handled by the serving layer.

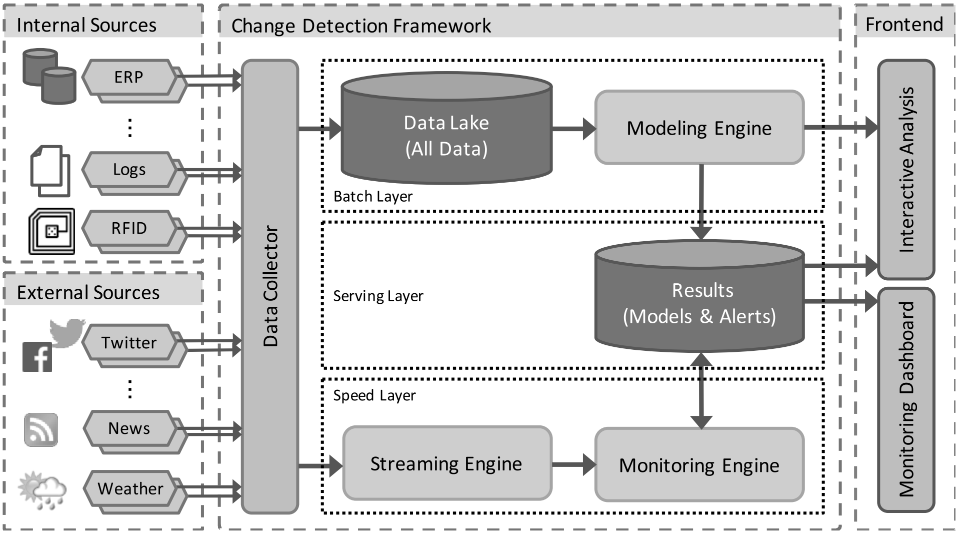

Fig. 2: System architecture of the generic big data change detection framework.

5 Conclusion

Detecting changes in data streams is a vital issue in many real-world applications. Despite the diversity in application and business areas and, thus, the variety in the relevant data, detecting changes has a common pattern: Relevant properties are defined and continuously monitored over time to detect significant deviations from an expected level.

Generally, detecting changes is not a trivial task. In big data scenarios, there are further challenges to cope with. In particular, dirty and noisy data call for robust change detection techniques and the sheer volume and velocity require scalable processing of data streams. Our change detection system uses common big data technologies and a modification of the Lambda architecture to enable scalable real-time change detection for arbitrary data streams and the selection of robust control charts is suitable to deal with real-world big data scenarios.

In the future, further change detection techniques such as model-based approaches will be integrated. Also, adding mechanisms to provide more complex plausibility checks for data items in a stream which require e.g. data from outside the stream will be relevant for some applications such as fraud detection. Finally, enabling the detection of changes not only over time but also over location will be a promising aspect for future research.

References

[AO08] Alfaro, J. L.; Ortega, J. F.: A Robust Alternative to Hotelling's T^2 Control Chart Using Trimmed Estimators. Quality and Reliability Eng. Int., 24(5):601–611, 2008.

[Ap02] Apte, C.; Liu, B.; Pednault, E.; Smyth, P.: Business Application of Data Mining. Comm. of the ACM, 45(8):49–53, 2002.

[Bö06] Böttcher, M.; Nauck, D.; Borgelt, C.; Kruse, R.: A Framework for Discovering Interesting Business Changes from Data. BT Technology Journal, 24(2), 2006.

[BPP07] Bersimis, S.; Psarakis, S.; Panaretos, J.: Multivariate Statistical Process Control Charts: An Overview. Quality and Reliability Eng. Int., 23(5):517–543, 2007.

[CDL07] Chou, C. L.; Du, T.; Lai, V. S.: Continuous Auditing with a Multi-Agent System. Decision Support Systems, 42(4):2274–2292, 2007.

[Cu07] Curry, C.; Locke, D.; Vejcik, S.; Bugajski, J.: Detecting Changes in Large Data Sets of Payment Card Data: A Case Study. In: KDD'07. San Jose, California, USA, 2007.

[FP99] Fawcett, T.; Provost, F.: Activity monitoring: Noticing Interesting Changes in Behavior. Proc. of the 5th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 1(212):53–62, 1999.

[GH15] Gandomi, A.; Haider, M.: Beyond the hype: Big Data Concepts, Methods, and Analytics. Int. Journal of Information Management, 35(2):137–144, 2015.

[Gr04] Grabert, M.; Prechtel, M.; Hrycej, T.; Günther, W.: An Early Warning System for Vehicle Related Quality Data. In: Industrial Conference on Data Mining. volume 3275 of Lecture Notes in Computer Science. Springer, pp. 88–95, 2004.

[Ho13] Howington, E. B.: Robust Monitoring of Contaminated Multivariate Data. Advances in Decision Sciences, 2013.

[Ki04] Kifer, D.; Ben-David, S.; Gehrke, J.: Detecting Change in Data Streams. In: Proc. of the 13th Int. Conf. on Very Large Data Bases (VLDB'04). pp. 180–191, 2004.

[Kl00] Kliem, R. L.: Risk Management for Business Process Reengineering Projects. Information Systems Management, 17(4):1–3, 2000.

[Ku08] Kuncheva, L. I.: Classifier Ensembles for Detecting Concept Change in Streaming Data: Overview and Perspectives. Proc. of the 2nd Workshop SUEMA, ECAI'08, 5–9, 2008.

[KX] Kruger, U.; Xie, L.: Statistical Monitoring of Complex Multivariate Processes: With Applications in Industrial Process Control.

[LMZ06] Li, Z.; Ma, H.; Zhou, Y.: A Unifying Method for Outlier and Change Detection from Data Streams. In: 2006 Int. Conf. on Computational Intelligence and Security. Vol. 1. IEEE, pp. 580–585, 2006.

[LSW07] Lambrigger, D. D.; Shevchenko, P. V.; Wüthrich, M. V.: The Quantification of Operational Risk using Internal Data, Relevant External Data and Expert Opinions. J. of Operational Risk, 2(3):3–27, 2007.

[Lu14] Lukoianova, T.; Hall, F.; Ave, N.; Rubin, V. L.: Veracity Roadmap: Is Big Data Objective, Truthful and Credible? Adv. in Classif. Research Online, 24:4–15, 2014.

[Mo09] Montgomery, D. C.: Introduction to Statistical Quality Control. John Wiley & Sons, sixth edition, 2009.

[MW15] Marz, N.; Warren, J.: Big Data: Principles and Best Practices of Scalable Real-Time Data Systems. Manning Publications Co., 2015.

[Ro89] Rocke, D. M.: Robust Control Charts. Technometrics, 31(2):173–184, 1989.

[Sa13] Sadiq, S.: Handbook of Data Quality. Springer, 2013.

[Sc12] Schroeck, M.; Shockley, R.; Smart, J.; Romero-Morales, D.; Tufano, P.: Analytics: The Real-World Use of Big Data. How Innovative Enterprises Extract Value from Uncertain Data. IBM Institute for Business Value, 2012.

[TGS14] Tran, D.-H.; Gaber, M. M.; Sattler, K.-U.: Change Detection in Streaming Data in the Era of Big Data: Models and Issues. SIGKDD Expl. Newsl., 16(1):30–38, Sep. 2014.

[Wo00] Woodall, W. H.: Controversies and Contradictions in Statistical Process Control. J. of Qual. Technol., 32(4):341–350, 2000.

[Wo04] Woodall, W. H.; Spitzner, D. J.; Montgomery, D. C.; Gupta, S.: Using Control Charts to Monitor Process and Product Profiles. J. of Qual. Technol., 36(3):309–320, 2004.

[ZZW10] Zhang, J.; Zou, C.; Wang, Z.: A Control Chart Based on Likelihood Ratio Test for Detecting Patterned Mean and Variance Shifts. Computational Statistics and Data Analysis, 54(6):1634–1645, 2010.