# A Novel Pattern Search Engine for Time Series Supporting Dynamic Expected Patterns within a Short Period of Time

Hai T. Mai
Department of Computer Engineering
Hanbat National University
125 Dongseo Daero, Yuseong-gu,Daejeon 305-719, ROK
hai.maituan@gmail.com

Young-chan Kim
Department of Computer Engineering
Hanbat National University
125 Dongseo Daero, Yuseong-gu,Daejeon 305-719, ROK
yckim@hanbat.ac.kr

## ABSTRACT

Recently, the world gets more and more distributed, big data now not only come from websites as Google, Bing, Yahoo having now or social networking service as Facebook etc..; there are sensors everywhere reporting millions of data each second. Among all the types of big data, data from sensors which is the most widespread is referred as time-series data. There are many attempts have been taken to recognize or retrieve the pattern of time-series such as recommender system, machine learning with pattern recognition and classification but all of them are push model. Once the expected patterns change, the whole system must be trained again it is great pain and it takes a huge of time. In other words, the existing systems cannot support dynamic expected patterns for retrieving the information. This paper proposes a novel pattern search engine for time-series which allows us to use any expected pattern or the combination of them as a query for searching information in a very short period of time without being trained or indexed again.

## Categories and Subject Descriptors

**Information systems→search engine**

## Keywords

Data mining; search engine; text mining; control chart; machine learning; pattern recognition, big data; sensor data;

\setcopyright{acmcopyright}
\conferenceinfo{BigDAS '15,}{October 20-23, 2015, Jeju Island, Republic of Korea}
\isbn{978-1-4503-3846-2/15/10}\acmPrice{\$15.00}
\doi{http://dx.doi.org/10.1145/2837060.2837064}

## 1 INTRODUCTION

In manufacturing processes, big data from sensors is always available; people can gather them and store in a historian database in duration of 10 years without any difficulty. Moreover, there are many advanced search engines such as Google, Bing, Yahoo as well as Facebook with graph search. However, there are still very pre-matured search engines for manufacturing data in process industries where big data from sensors is the most widespread.

In order to do mining on these data, many systems based on machine learning or statistical process control (SPC) has been developed, some of them are success to provide pattern recognition and classification on manufacturing data.

Nevertheless, all of them follow the push model in which a tiny modification in expected patterns causes the entire system must be trained again. In fact, the criteria of retrieving data are always changed; some people want these expected patterns; the others want the other ones. Because of using push model, dynamic criteria is a huge burdensome for not only existing systems but also the novices who get confused about why the needed data cannot be retrieved with such a tiny modification in criteria.

As a result, it makes a quite big gap between existing systems and the need for retrieving data with dynamic criteria easily.

To fill this gap, we have proposed a novel pattern search engine which not only allows us to reduce data dimension of time series but also provides a new way to retrieve any expected patterns dynamically without training or indexing the data again. The search engine is based on SAX (Symbolic Aggregation approXimation) [9] and Lucene a well-known and matured open-source search engine [10].

Our contribution is making time-series data become accessible for text mining search engine and we develop a new scoring system that is more suitable for ranking the time-series than Lucene core itself.

In our experiments, we have used SPC Control chart, a very popular and significant important tool in manufacturing. With proposed search engine, user can create any rules by himself or freely utilize well-known rules for retrieving the best relevance data such as Western Electric rules, Wheeler rules, Nelson rules etc...

# 2 PREVIOUS WORK

## 2.1 Cutting-edge Search Engines

A massive data volume is available on the websites; many search engines such as Google, Bing and Yahoo crawling and indexing these data and finally provide the key words based search engines. Because of heavily depending on key words in these search engines, user must break the information they need into separated queries by themselves.

To fix this problem, some attempts have done to improve the search engine by accepting natural language as a query then internally, search engine will parse the query using natural language processing (NLP) [2] or developing specific domain search engine with specialized word ranking systems [3][4][5].

Despite of having so many advanced techniques from information retrieval and NLP, these search engines still do not fulfill the need of user in manufacturing data. In the matter of fact, what the user need are not only the expected key words but the patterns of data.

## 2.2 Pattern of Data

Truly, pattern of data is much more important than just the key words themselves or specific data points because with pattern of data user can do anomaly detection and perform other tasks that related to pattern of data. The knowledge and the characteristic of data lay behind the pattern of data, not data points.

While big data is growing rapidly everywhere, it makes the pattern of data become so important that trend information which is one kind of pattern of data is clearly defined as "a kind of summarization of temporary statistical data, obtained through synthesis rather than simple enumeration" by the MuST workshop (Multimodal Summarization for Trend information). Trend-related Search Engine [6] was introduced by Yanjun Zhu, Yasufumi Takama, Yu Kato, Shogo Kori and Hiroshi Ishikawa. It provides trend-related search with semantic search engine using SPARQL query language. The queries can be MAX/MIN; PEAK/BOTTOM; SI (Sudden Increase) / SD (Sudden Decrease).

Web usage data are processed by Crawler that calculates the characteristics of data such as MAX/MIN etc… and save them into database, later the client will retrieve these characteristics from the web server, and database by SPARQLs. The biggest limitation of this approach is that after Crawler has processed data, there is no chance for user to select other patterns of data and the format of queries is quite complex compared to search engine based on key words.

# 3 BACKGROUND

## 3.1 SAX (Symbol Aggregation approXimation)

In 2002, Lin and Keogh [9] proposed new approach called SAX. Basically, SAX is based on PPA [16, 17] and there is an assumption on normality of the resulting aggregated values. Lin and Keogh also state that SAX is the first symbolic representation for time series that allow for dimensionality reduction and indexing with a lower distance measure.

There are two steps of doing SAX, firstly the data is transformed into PPA representation and then the transformed PPA is symbolized into a sequence of discrete string.

With PPA [16, 17], "A time series C of length n can be represented in a k-dimensional space by a vector k and the ith element of C is calculated by the following equation":

$$x_i = \frac{k}{n}\sum_{j=\frac{n}{k}(i-1)+1}^{\frac{n}{k}i} c_j \qquad (1)$$

Below figure shows how time-series C can be represented by SAX.

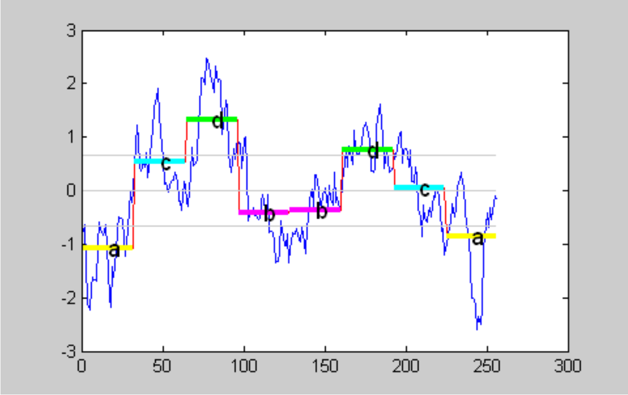


**Figure 1: A time series C is represented by SAX (k=4)**

By doing that we will have two major advantages:

- **Dimensionality Reduction**: Dimensionality reduction of PPA [16, 17] is automatically carried over to this representation.
- **Lower Bounding:** Distance measure between two symbolic strings can be proved "by simply pointing to the existing proof for the PAA representation itself".

### 3.2 Lucene

Dough Cutting originally wrote Lucene in 1999 [10], it is a free open source information retrieval software library. While suitable for any application that requires full text indexing and searching capability, Lucene has been widely recognized [12] for its utility in the implementation of internet search engine and local or single-site searching. Lucene offer Scalable and High-Performance indexing with Efficient Scoring & Search algorithm.

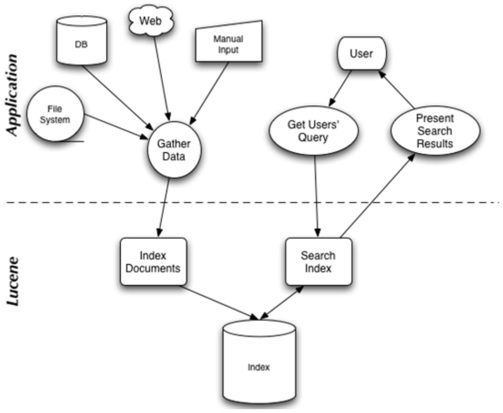


**Figure 2: A typical application integration with Lucene**

Lucene has three approaches for text mining: VSM (Vector Space Model), Probability Model (BM25) and Language Model. VSM is widely accepted because of the simple in implementation and the reasonable result. In VSM, the most important things are tf (term frequency), idf (inverse of document frequency) and Boost. The final score of any term mainly depends on them.

In DefautSimilarity implementation, tf is defined as:

$$tf(t) = \sqrt{frequency} \qquad (2)$$

tf states that although, the word that appears many times in a document has higher ranking than others but it should be reduced by root square function. So tf is not linear proportional to frequency.

idf is defined as:

$$idf(t) = 1 + log\left(\frac{numDocs}{docFreq+1}\right) \qquad (3)$$

idf states that if the word appears in many documents then it should have low ranking.

### 3.3 Control Chart

Statistical Process Control (SPC) charts also known as Shewhart Charts [1] have been used widely in many processes. Big data that comes from SPC charts is available in many data warehouse.

Control charts can be classified into two general types. Control charts for central tendecy and variability are collectively called ***variables control charts.***

The X Chart is the most widely used chart for controlling central tendecy , whereas charts based on either the sample range or sample standard deviation are used to control process variability.

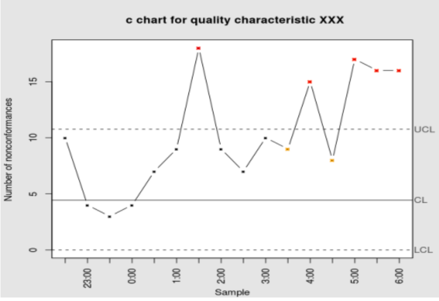


**Figure 3: Control chart**

Many quality charactiristics are not measured on a continuous scale or even a quatitative scale. Therefore, we can judge product as conforming or nonconforming based on it has specific attributes or the number of defects that appearing on product. Control charts support such quality characteristics are called ***attributes control charts.***

Control charts have had a long history of use in U.S and all over the world. There are at least five reasons for their popularity:

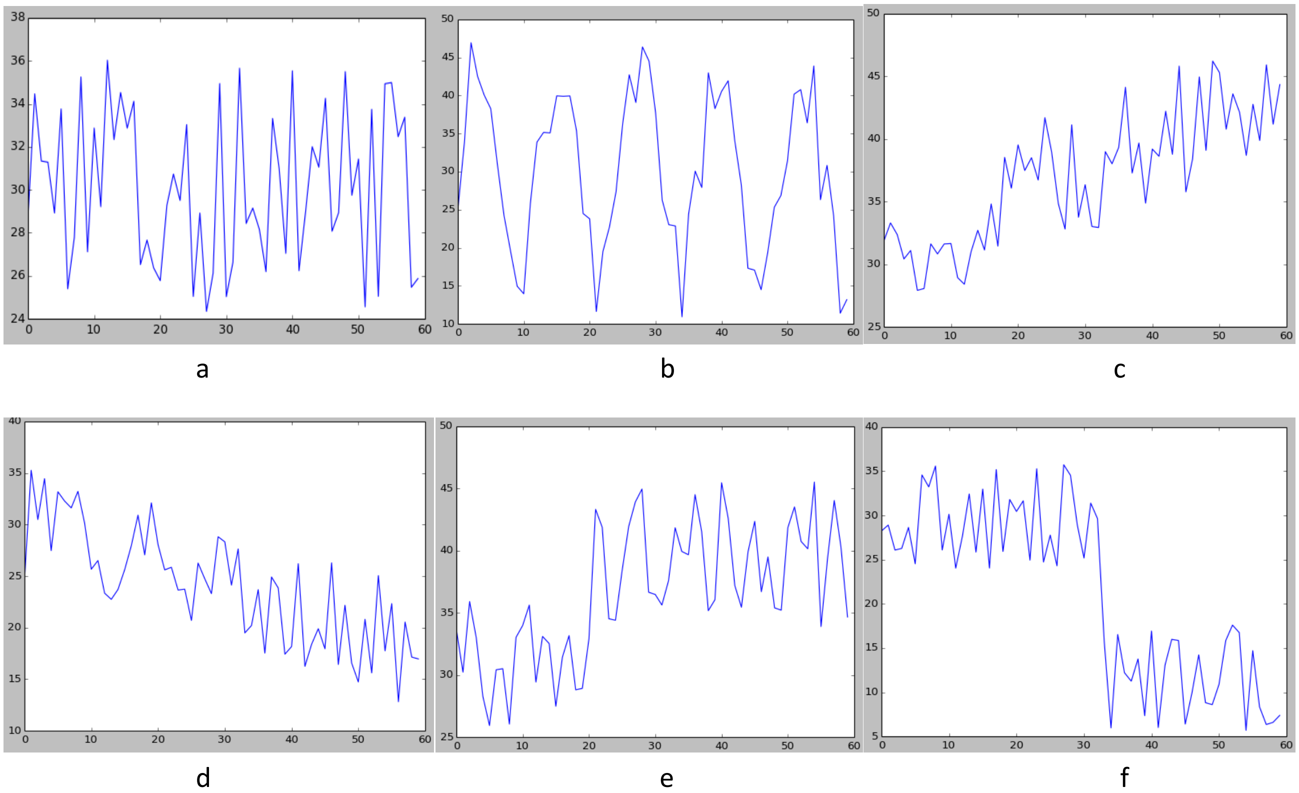
- Control charts are a proven technique for improving productivity
- Control charts are effective in defect prevention
- Control charts prevent unnecessary process adjustment
- Control charts provide diagnostic information
- Control charts provide information about process capability.

### 3.4 Detect "out of control"

Although, Control Chart is very popular in big data of process industries, how to interpret it still is a tough question to answer. To solve this problem some rules have been developed for recognizing whether a control chart is anomaly as below [13]:

- Western Electric Rules
- Wheeler Rules
- Nelson Rules
- Custom Rules

Some attempts in **software** development are already taken to answer the question whether a control chart is in normal mode. Among them, the famous one is Minitab [14] with Minitab control chart test that implement Nelson Rules. Minitab or other software still based on raw data point scanning mechanism for detecting anomaly of control charts. In this paper, we proposed a search engine which its data are fully indexed. It not only significantly speeds up the performance but also reduces dimension of data.



a b c

d e f

**Figure 4 (a) Normal pattern, (b) Cyclic pattern, (c) Upward trend pattern, (d) Downward trend pattern (e) Upward shift pattern, (f) Downward trend pattern**

## 4 PROPOSED SEARCH ENGINE

In order to extend Lucene to our Pattern Search Engine, we need to extend three the important classes which transform Lucene to be able to work with pattern search. They are Analyzer, Similarity and QueryParser .

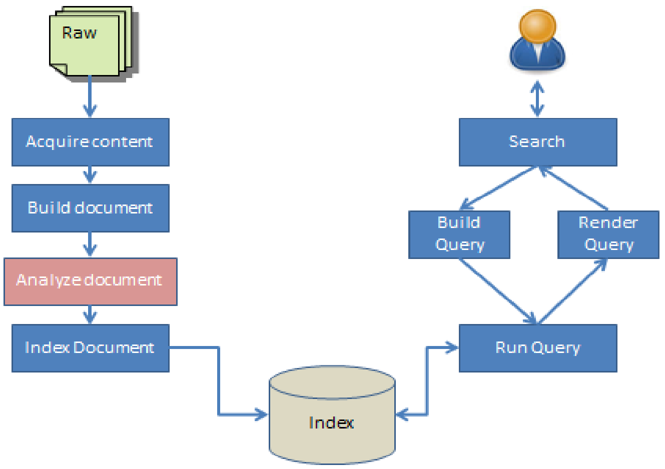


**Figure 5: Lucene Flow**

**Analyzer**: An Analyzer builds TokenStreams, which analyze text. It thus represents a policy for extracting index terms from text.

**Similarity**: defines the components of Lucene scoring.

**Query**: Provide query interface for user to retrieve information.

Our proposed pattern search engine can work with many kinds of data, but for validation we choose control chart data, a very popular big data in process industry.

In control chart representation, we define dimensional parameter k = 8 which means that any control chart can be represented as a string which constructed by only 8 characters a, b, c, d, e, f, g, h.

### 4.1 Proposed Analyzer for SAX

We have developed SAXAnalyzer based on pandas, numpy, scikit-learn libraries of Python [18] and StandardAnalyzer of Lucene. SAXAnalyzer helps transform control chart data to SAX directly and builds TokenStreams such like super class does. StandardAnalyzer must set stopword to empty in order to ignore stopword function.

### 4.2 Proposed Similarity for SAX

In Search Engine, all key words are the same, however, in control chart any point that is out of 3-sigma (a,h characters in SAX representation), for example, are much more important than points are located within 1 sigma (d,e). So the weight, scoring of them should be larger than others'.

We have developed SAXSimilarity that extends Similarity from Lucene but adding suitable boost factor for each data range.

### 4.3 Proposed QueryParser for SAX

In this paper, we proposed two Query Parsers; one is CustomQueryParser which allows user to input any expected pattern to search. The other is well-design QueryParser which implements the well-known control chart rules such as Western Electric Rule …

All of our Query Parsers are extended from ComplexPhraseQueryParser which can support the complex queries to fulfill all the rule conditions.

For example, with WesternElectricQueryParser, users don't need to provide any query to our Engine, we has already implemented all rules from Western Electric Rules in our Query Parser.

Here is the code for running Rule1 or All rules from Western Electric Rules:

```
Query bQuery = new
WesternElectricQueryParser("data",analyzer).Rule1()

Query bQuery = new
WesternElectricQueryParser("data",analyzer).AllRules();
```

**Implementation**: Let's check how we can implement rule 1 of Western Electric Rules in detail.

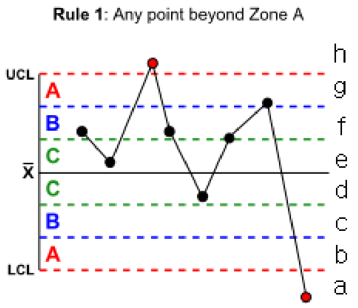


**Figure 6: Western Electric Rule 1**

Rule 1: Any single data point falls outside the 3σ limit from the centerline (i.e., any point that falls outside Zone A, beyond either the upper or lower control limit)

In order to implement this rule, we just developed a simple query to engine.

```
private String rule1="a h";
public Query Rule1() throws ParseException{
        return super.parse(rule1);
}
```

Query Parser breaks rule1 to query 2 terms "a" and "h" then search for term "a" or "h" from indexes.

Similarly, all rules of Western Electric can be done with very simple strings as below:

```
String rule1="a h";

String rule2_1="\"/[g-h]/ /[g-h]/\"";
String rule2_2="\"/[a-b]/ /[a-b]/\"";

String rule3_1="\"/[a-c]/ /[a-c]/ /[a-c]/ /[a-c]/ \"";
String rule3_2="\"/[f-h]/ /[f-h]/ /[f-h]/ /[f-h]/ \"";

String rule4_1="\"/[a-d]/ /[a-d]/ /[a-d]/ /[a-d]/ /[a-d]/
/[a-d]/ /[a-d]/ /[a-d]/ /[a-d]/\"";

String rule4_2="\"/[e-f]/ /[e-f]/ /[e-f]/ /[e-f]/ /[e-f]/
/[e-f]/ /[e-f]/ /[e-f]/ /[e-f]/\"";
```

Our QueryParser supports Regular Expression that allow user to write very sophisticated queries which makes query much easier to understand. Because some rules are combined from both side of center line ($\bar{X}$), so rule 2 are the combination of 2 queries rule2_1 and rule2_2.

```
public Query Rule2() throws ParseException{

BooleanQuery bQuery = new BooleanQuery();

bQuery.add(super.parse(rule2_1),
BooleanClause.Occur.SHOULD);

bQuery.add(super.parse(rule2_2),
BooleanClause.Occur.SHOULD);

        return bQuery;
}
```

## 5 EXPERIMENTS

In this paper, we have experimented with all possible queries that user frequently use. They are Term Search, Pattern Search and Anomaly Search. The dataset [8] contains 600 samples of control chart which are categorized into 6 groups as described in figure 4.

### 5.1 Term Search

Term search is quite important in case of peak points or out of control points (3-sigma). In the Rule1 of Western Electric Rules, the query is term queries which are tried to match the data with term.

In our experiment, Rule 1 runs with 600 samples of control charts:

**Table 1: Control chart with Western Electric Rule1**

| Control Chart type | Term Found |
|---|---|
| Normal | 0/100 |
| Cyclic | 100/100 |
| Increasing trend | 95/100 |
| Decreasing trend | 95/100 |
| Upward shift | 87/100 |
| Downward shift | 84/100 |

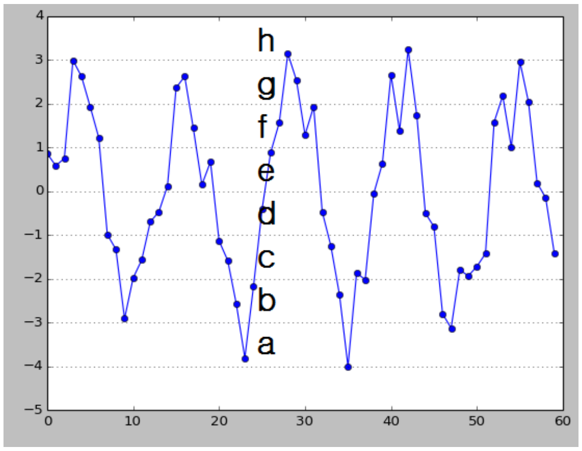


**Figure 7: SAX with Z-Transformation**

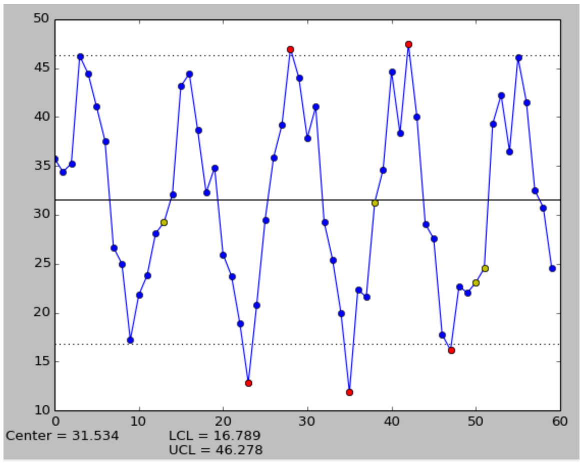


**Figure 8: X_MR_X Chart with SPC Package in Python**

In figure 7, because of being processed with Z-transformation, thus Y-Axis represents for sigma. Any point that its value is above 3 that means it is over 3-sigma. From the figure, we can easily point out what data is out of control points (3-sigma).

**Validation**: as we can see in these figures 6, 7, 8 above our result are completely same as standard SPC package does. Although, the results are the same but with big data, SPC Package cannot handle this burdensome because if the rules are change they must do all calculation again from beginning. In our approach, user just changes the interested terms or patterns without processing raw data again.

## 5.2 Pattern Search

Our proposed search engine satisfies not only the term search but also the pattern search. Pattern search are any shape of trend that can break into the sequence of SAX. For example, the Rule 4 of Western Electric Rule as below:

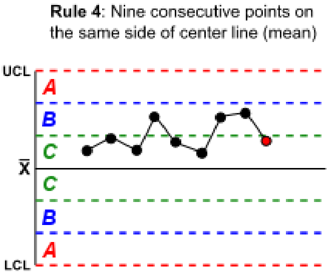

Rule 4: Nine consecutive points on the same side of center line (mean)
UCL
A
B
C
X̄
C
B
A
LCL

**Figure 9: Western Electric Rule 4**

User also can define their own interested patterns by themselves and retrieve the data without re-train or re-index the system again.

**Table 2: Control chart with Western Electric Rule4**

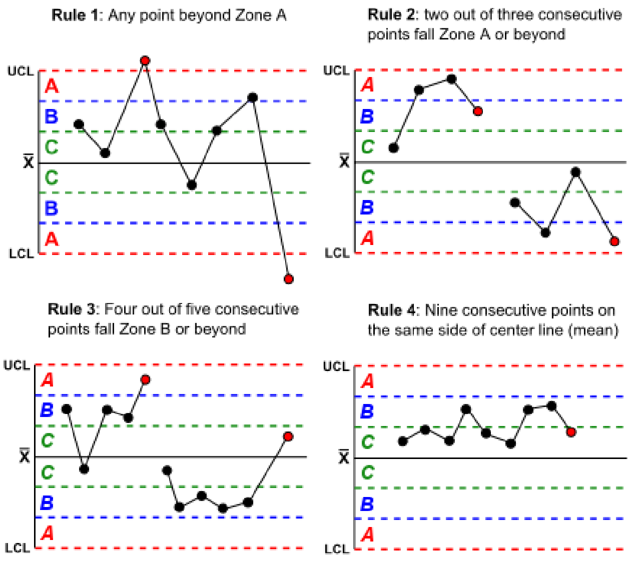
| Control Chart type | Pattern Found |
|---|---|
| Normal | 6/100 |
| Cyclic | 10/100 |
| Increasing trend | 94/100 |
| Decreasing trend | 96/100 |
| Upward shift | 96/100 |
| Downward shift | 97/100 |

As we can see from the table, even in the normal control chart, there are still 6 patterns that violate the Rule 4. As a result, the Term Search is not enough for implementing Western Electric Rules as user needed.

## 5.3 Anomaly Search

As have been said in 3.4, control chart together with rules can be used as a very important tool for detecting anomaly of processes which its massive data dominantly come from sensors.

In this section, we have tested with 600 samples of control chart data with Western Electric Rules. There are 21 cases are anomalies even in normal chart type. The results are in below table:



**Rule 1**: Any point beyond Zone A

**Rule 2**: two out of three consecutive points fall Zone A or beyond

**Rule 3**: Four out of five consecutive points fall Zone B or beyond

**Rule 4**: Nine consecutive points on the same side of center line (mean)

**Figure 10: Western Electric All Rules**

**Table 3: Control chart with Western Electric All Rules**

| Control Chart type | Anomaly Found |
|---|---|
| Normal | 21/100 |
| Cyclic | 100/100 |
| Increasing trend | 100/100 |
| Decreasing trend | 99/100 |
| Upward shift | 100/100 |
| Downward shift | 100/100 |

## 6 CONCLUSION

In process industry, big data which is pre-processed such as control charts is popular in every plant or manufacturing factory. As a result, Pattern Retrieval plays a significant role in big data analysis.

By using Pattern Retrieval, process engineers can get any interested pattern that he wants in order to solve the related problems in plant. However, Pattern Retrieval is not considered enough for improving the effectiveness and efficient of plant data.

Currently, with state-of-art search engine such as Google, Bing.., people get very familiar with text search engine but they're struggling with the question how to make time-series as easy as text search engine.

Our contribution bridges the gap between the time-series data and text search engine. After transforming to SAX, now time-series are available for indexing and scoring by advanced search engine. We have proposed the novel search engine which is more appropriate for scoring, analyzing and querying time-series than text search engine itself such as Lucene Core.

## 7 REFERENCES

[1] Douglas C.Montgomery, Introduction to Statistical Quality Control, USA: Willey, 2005.

[2] A. Ferreira, J. Atkinson, Intelligent Search Agents Using Web-Driven Natural-Language Explanatory Dialogs, IEEE Computer, Vol. 38, No. 10, pp. 44–52, 2005.

[3] K. Matsumoto, A. Monden, T. Kamei, Development of a Software Search Engine for the World Wide Web, Workshop on Software Product Archiving and Retrieving System, pp. 39–44, 2004.

[4] S. Oyama, T. Kokubo, T. Ishida, Domain-Specific Web Search with Keyword Spices, IEEE Transactions on Knowledge and Data Engineering(TKDE), Vol. 16, No. 1, pp. 17–27, 2004.

[5] H. Nabeshima, R. Miyagawa, Y. Suzuki, K. Iwanuma, Rapid Synthesis of Domain-Specific Web Search Engines Based on Semi-automatic Training-Example Generation, WI'06, pp. 769–772, 2006.

[6] Yanjun Zhu, Yasufumi Takama, Yu Kato, Shogo Kori and Hiroshi Ishikawa, Introduction of Search Engine Focusing on Trend-related Queries to Market of Data, 2014 IEEE International Conference on Data Mining Workshop, 2014.

[7] Eamonn Keogh, Jessica Lin, Ada Fu, HOT SAX: Finding the Most Unusual Time Series Subsequence: Algorithms and Applications, ICDM 2005.

[8]https://archive.ics.uci.edu/ml/datasets/Synthetic+Control+Chart+Time+Series

[9] Lin, J., Keogh, E., Lonardi, S. & Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA. June 13.

[10] http://lucene.apache.org/

[11] Wang, Chih-Hsuan, Guo, Ruey-Shan, Chiang, Ming-Huang and Wong, Jehn-Yih (2008) "Decision tree based control chart pattern recognition", International Journal of Production Research, 46:17, 4889 — 49017.

[12] McCandless, Michael; Hatcher, Erik; Gospodnetić, Otis (2010). Lucene in Action, Second Edition. Manning. p. 8. ISBN 1933988177.

[13] http://en.wikipedia.org/wiki/Control_chart

[14]http://support.minitab.com/en-us/minitab/17/topic-library/quality-tools/control-charts/basics/using-tests-for-special-causes/

[15] http://en.wikipedia.org/wiki/Nelson_rules

[16] Keogh, E,. Chakrabarti, K,. Pazzani, M. & Mehrotra "Dimensionality reduction for fast similarity search in large time series databases", Journal of Knowledge and Information Systems. (2000).

[17] Yi, B-K and Faloutsos, C., "Fast Time Sequence Indexing for Arbitrary Lp Norms", Proceedings of the VLDB, Cairo, Egypt, Sept, (2000).

[18] http://scikit-learn.org