Cleanits: A Data Cleaning System for Industrial Time Series

Xiaoou Ding, Hongzhi Wang, Jiaxuan Su, Zijue Li, Jianzhong Li, Hong Gao
Harbin Institute of Technology
92 West Dazhi Street
Harbin, China
dingxiaoou@stu.hit.edu.cn, {wangzh, itx351, lizijue, lijzh, honggao}@hit.edu.cn

ABSTRACT

The great amount of time series generated by machines has enormous value in intelligent industry. Knowledge can be discovered from high-quality time series, and used for production optimization and anomaly detection in industry. However, the original sensors data always contain many errors. This requires a sophisticated cleaning strategy and a well-designed system for industrial data cleaning. Motivated by this, we introduce Cleanits, a system for industrial time series cleaning. It implements an integrated cleaning strategy for detecting and repairing three kinds of errors in industrial time series. We develop reliable data cleaning algorithms, considering features of both industrial time series and domain knowledge. We demonstrate Cleanits with two real datasets from power plants. The system detects and repairs multiple dirty data precisely, and improves the quality of industrial time series effectively. Cleanits has a friendly interface for users, and result visualization along with logs are available during each cleaning process.

PVLDB Reference Format:
Xiaoou Ding, Hongzhi Wang, Jiaxuan Su, Zijue Li, Jianzhong Li, and Hong Gao. Cleanits: A Data Cleaning System for Industrial Time Series. *PVLDB*, 12(12): 1786-1789, 2019.
DOI: https://doi.org/10.14778/3352063.3352066

1. INTRODUCTION

Industrial big data evaluation plays a key role in intelligent manufactory. With the rapid growth of data from the industrial internet, knowledge discovery in industrial data contributes to the improvement in industrial process as well as anomaly detection. To achieve this, high-quality data is identified as the basic premise on accomplishing information extraction and valuable knowledge discovery. The demand for high quality industrial data has grown stricter [1].

The time series generated by sensors is a normal form of industrial data. To describe the status of a workflow with multiple machines, multi-dimensional time series are generated with each dimension (*i.e.*, attribute) belonging to one sensor. In reality, it is challenged to obtain high-quality time series due to the following major quality problems.

- **Missing values**. Some dimensions of the time series often contain missing values at discrete time points or during a time period. Machine sensors may fail to collect the value of a time point. Short time transmission fault is another reason for the absence of these values.
- **Inconsistent attribute values**. Signal interference may happen when equipments are undergoing working condition transition. As a result, an attribute may record the information of another attribute during a time period. It leads to subsequence inconsistency problems during a certain time duration.
- **Abnormal values and subsequences**. Imprecise or dirty values are prevalent in industrial time series [2]. Unplanned machine failures give rise to the sudden value changes at some time points. Unexpected troubles in sensor recording also lead to short-length abnormal sequences among various attributes.

Many data cleaning techniques have been developed to solve data quality problems [3, 4], and cleaning platforms are applied in various data repairing tasks. Unfortunately, most of them do not apply to time series cleaning, especially for industrial time series. On the one hand, data cleaning tasks are always industry-specific problems. Cleaning methods hardly perform well without the guidance from domain knowledge and adequate understanding about the industrial process. On the other hand, data reflects multi-modal working conditions (*e.g.*, smooth, step, sparse pattern [1]) of machines. Errors in different patterns are difficult to be either identified or repaired without a well-designed cleaning approach.

Thus, comprehensive time series cleaning approaches have become a urgent demand in intelligent manufacturing. Motivated by this, we develop *Cleanits*, which makes effective data *clean*ing in multi-dimensional *i*ndustrial *t*ime *s*eries. Our system focuses on the following objectives.

(1) **Effectiveness in industrial data**. After a thorough practice research, we develop three data cleaning functions in Cleanits, namely missing values imputation, matching inconsistent attribute values, and anomaly detection and repairing. These approaches with the process order among them are verified to be effective in the demo scenarios with real industrial datasets.

(2) **Domain knowledge support**. Domain knowledge has a meaningful influence in industrial data cleaning. Cleanits is designed to understand domain knowledge including sequence constraints and machine working patterns. Thus,

Proceedings of the VLDB Endowment, Vol. 12, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3352063.3352066

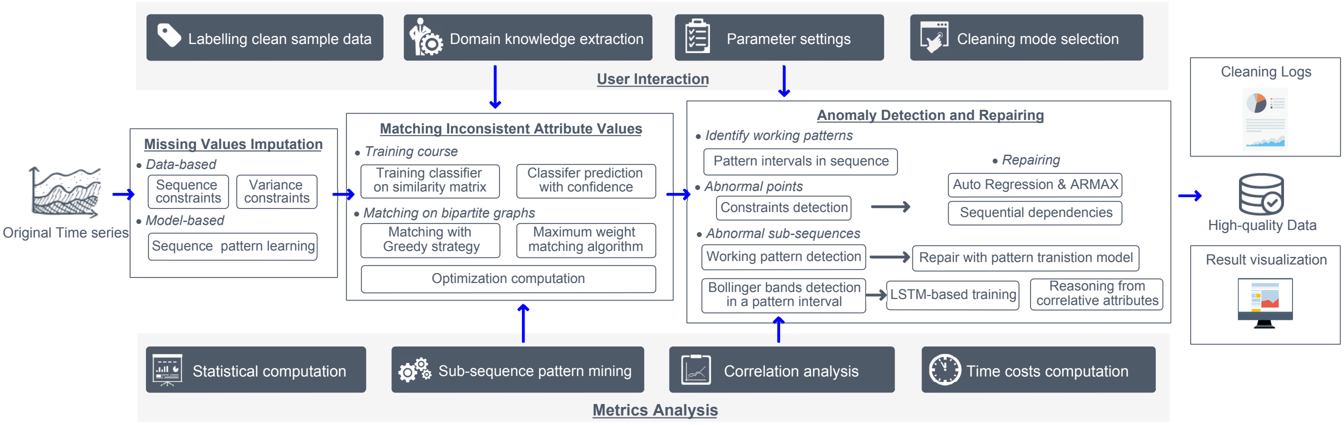

Labelling clean sample data
Domain knowledge extraction
Parameter settings
Cleaning mode selection
User Interaction
Cleaning Logs
Missing Values Imputation
Data-based
Sequence constraints
Variance constraints
Model-based
Sequence pattern learning
Matching Inconsistent Attribute Values
Training course
Training classifier on similarity matrix
Classifier prediction with confidence
Matching on bipartite graphs
Matching with Greedy strategy
Maximum weight matching algorithm
Optimization computation
Anomaly Detection and Repairing
Identify working patterns
Pattern intervals in sequence
Repairing
Auto Regression & ARMAX
Sequential dependencies
Abnormal points
Constraints detection
Abnormal sub-sequences
Working pattern detection
Repair with pattern tranistion model
Bollinger bands detection in a pattern interval
LSTM-based training
Reasoning from correlative attributes
Original Time series
High-quality Data
Result visualization
Statistical computation
Sub-sequence pattern mining
Correlation analysis
Time costs computation
Metrics Analysis

Figure 1: Framework overview of Cleanits

Cleanits provides reliable data repairing results on precise locations in each sequence of the time series.

(3) **Customized and user-friendly cleaning modes.** Considering the parameter settings and the balance between time costs and the performance of different cleaning strategies, we design multiple data cleaning modes. Users can choose a proper mode according to their data quality requirements.

To summarize, we make the following contributions in this paper. 1) We develop a data cleaning system Cleanits for industrial time series. 2) Cleanits implements three repairing functions to effectively improve the quality of the multi-dimensional time series. 3) Cleanits provides a well-considered interface designment for users to operate a customized data cleaning. 4) We run Cleanits on real machine sensors data from two power plants for system functions demonstration.

2. SYSTEM OVERVIEW

Figure 1 shows the framework of Cleanits, which has three cleaning functions: 1) Missing value imputation, 2) Matching inconsistent attribute values, and 3) Anomaly detection and repairing. Besides, it has a metric analysis module for parameter measurement and a user interaction module.

Missing value imputation. In the first step, Cleanits detects missing values in each attribute of the input time series, and fills the missing parts to satisfy the statistical properties of each attribute. The system provides two imputation strategies: data-based approach and model-based approach, respectively. In the data-based approach, important sequence features are described in semantics, according to the combination of domain knowledge and the clean sample data. Our system Cleanits cleans missing values with sequence constraints [5] and short-window variance constraints [3]. It also provides learning-based imputation results for sequences based on their corresponding patterns according to sub-sequence pattern mining results.

Matching inconsistent attribute value. We then detect and repair inconsistent attribute values. We develop an exact maximum weight matching based on Kuhn-Munkres (KM) algorithm in [6] and a fast greedy matching algorithm. We also propose pruning and optimizing matching algorithms for users to alternatively select a cleaning mode with the balance of the time costs and the cleaning accuracy.

Anomaly detection and repairing. After inconsistent subsequences have been matched to right attributes, we conduct anomaly detection and repairing as the third cleaning step. Cleanits not only cleans abnormal values of discrete data points, but also captures abnormal subsequences effectively. During this course, users are able to know exactly where the abnormal cases happen. Cleanits provides the visualization of abnormal detection, enabling users to modify and label anomaly cases via the interaction module. The system allows users to upload manual repairing result, and these labelled data will be applied in the later training process.

Considering the complex working conditions reflected by industrial data, we design three kind of anomaly repairing methods: statistics-based, model-based and data-based approaches. The system chooses proper algorithms among them, according to both statistical analysis and the cleaning mode selected by users.

Metrics analysis. This module provides measurement for data cleaning functions. Cleanits executes four computing sections in this module: statistical computation, subsequence pattern mining, correlation analysis and time costs computation.

Statistical computation. Important statistical indicator values are computed to capture sequence features reliably. These indicators guide the repair of errors, and further, they verify the reliability of the cleaning results.

Subsequence pattern mining. Various working conditions of machines appear as multi-modal sequence patterns in the whole dataset. We propose a subsequence pattern mining process in order to capture different patterns (*e.g.*, stationary, floating, and smooth pattern) in each attribute. The system makes decisions on appropriate cleaning solutions according to features of different sequence patterns. Thus, it provides targeted cleaning results instead of generic ones.

Correlation analysis. Since the data records working conditions of a machine group, some attributes may be relevant. On the one hand, some attributes record data belonging to the same machine. Thus, they describe similar sequence patterns. On the other hand, some attributes record collaboration working patterns among a local machine group. These attribute values probably have a correlation with each other. We compute the correlation among attributes according to both the domain knowledge and statistical metrics.

Thus, dirty data in an attribute can be repaired effectively according to its strong correlated attributes. Cleanits achieves effectiveness in both detection and repairing with the correlation analysis on multi-dimensional time series.

User interaction. This module is designed for an easy-to-use interaction between users and the system. Before the cleaning process starts, users are able to upload a sample of clean data. These labelled training data will be used in the model-based cleaning methods. In addition, the domain knowledge (*e.g.*, sequence constraints, working state transition, and attributes correlation) contributes to an accurate detection of errors.

During the cleaning process, users are able to set parameters depending on their own requirements. Since some cleaning functions contain quite a few parameters, Cleanits also allows default settings. We design four cleaning modes for users: 1) *Overall* mode implements a thorough cleaning with all parameters set by users. It achieves a good cleaning performance, but has the highest time cost; 2) *High-efficiency* mode implements a fast coarse-grained cleaning with the least time cost. Only a small part of parameters are determined manually; 3) *Automation* mode runs all cleaning functions with default parameter settings. All repairing methods are determined by the system, and users are not involved in the cleaning process; and 4) *Time restriction* mode repairs data according to a running time limitation set by users. The cleaning result reflects the balance between the time cost and the cleaning granularity.

3. IMPLEMENTATIONS

In this section, we first introduce basic definitions in time series quality, and then discuss cleaning functions designed in Cleanits. Note that missing values can be detected easily by checking null values between space characters in sequences, and the imputation strategy is similar with anomaly value repairing. We focus on two cleaning functions and introduce matching inconsistent attribute values in Section 3.2 and anomaly repairing in Section 3.3.

3.1 Preliminaries

A *time series* $\mathcal{S} \in \mathbb{R}^{N\times M}$ is denoted by $\mathcal{S} = \{S_1, ..., S_M\}$, where M is the total number of attributes: $M = |attr(\mathcal{S})|$. $S_i = \langle s_1, ..., s_N\rangle, (i \in [1, N])$ is a *sequence* on attribute A_i of $\mathcal{S}$, where $N = |S_i|$ is the total number of elements in S_i, *i.e.*, the length of S_i. $s_j = (x_j, t_j), (j \in [1, m])$, where x_j is a real-valued number with a timestamp t_j, and $(j < k) \Leftrightarrow (t_j < t_k)$. A *subsequence* $S_{i.[l,n]} = \langle s_l, ..., s_n\rangle$ is a continuous subset of S_i beginning from the element s_l and ending in s_n. $T_{[l:n]}$ is the *time interval* of $S_{i.[l,n]}$.

The inconsistency problem. A time interval $T_{[l:n]}$ is inconsistent iff it satisfies: (i) $\exists S_{i.[l,n]} \nsubseteq S_i$, but $S_{i.[l,n]} \subseteq S_j$, $(i, j \in [1, m]$ and $i \neq j)$. Each $S_{i.[l,n]} \nsubseteq S_i$ is one unmatch in $T_{[l:n]}$, and the total number of such unmatch is no less than 2, and (ii) the length of $T_{[l:n]}$ is no less than a given threshold, *i.e.*, $|n - l + 1| \geq \delta$.

The anomaly problem. For a sequence $S_i = \langle s_1, ..., s_n\rangle$ in $\mathcal{S}$, $\mathcal{F}_1(s) = [f_l, f_u]$ is a value prediction function of the element s in S_i, where f_l (resp. f_u) represents the min (resp. max) expected value of s. $s = (x, t)$ is identified as an abnormal point iff $x \notin [f_l, f_u]$. $\mathcal{F}_2(S_{[l,n]})$ is a pattern prediction function of subsequence $S_{[l,n]}$. $S_{[l,n]}$ is identified as a k-length abnormal sequence iff $Dist(S_{[l,n]}, \mathcal{F}_2(S_{[l,n]}))$ is larger than a required distance threshold.

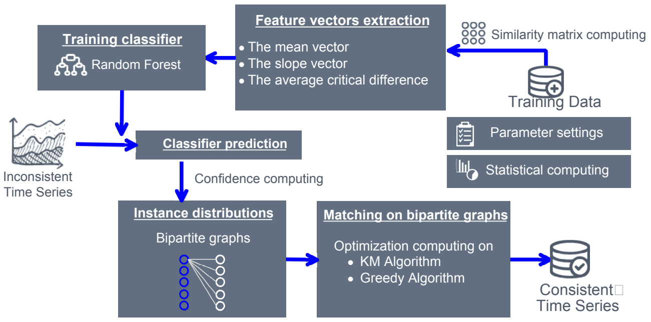


Figure 2: Matching inconsistent subsequences

3.2 Matching Inconsistent Attributes

The inconsistency repair solution in Cleanits is shown in Figure 2. We first make classifier prediction and then match inconsistent subsequences to their corresponding attributes. Each sequence is treated as a classification with several feature vectors extracted from the computed similarity matrix. We construct the classifier based on random forest considering its efficiency on large-scale data and high performance on multi-dimensional time series.

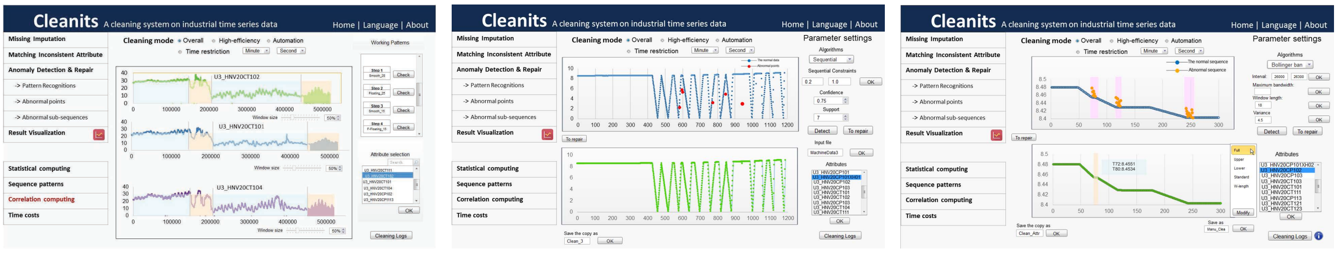
When the classifier prediction function begins, we compute instance distributions for each sequence, *i.e.*, a confidence value set of subsequences matched to each attribute. Accordingly, we construct a bipartite graph $G = (N_{S'}, N_S, E, W)$, and mark the confidence as the weight of an edge. That is, $w(S'_i, S_j)$ is the confidence of matching S'_i to S_j, $(S'_i \in N_{S'}, S_j \in N_S)$.

We design two matching approaches to achieve precise consistent repairing, according to the matching confidence distribution. If most of the matching weights are high, we use KM algorithm to obtain a matching pattern with the max sum of weights over all edges on G. If there exists several low matching confidences, we consider a greedy matching strategy. We first select the edge (S'_i, S_j) from $E(G)$ with $\max_{S'_i \in S', S_j \in S} w(S'_i, S_j)$, and add it to the result set. We then delete all the edges connecting with either S'_i or S_j. We iteratively process the above steps until $E(G)$ is empty, and obtain a consistent time series.

3.3 Anomaly Detection and Repairing

The unexpected change in either the value or the pattern is recognized as an anomaly in industrial time series [1]. We first recognize machine working conditions and divide each sequence into intervals according to different working patterns. After that, we detect and repair abnormal data points and subsequences.

For abnormal data points repairing, we identify the unexpected values according to both sequential dependencies (SD) [5] and window-variance constraints proposed in our previous work [7]. SDs express the required value difference between two consecutive time points, *i.e.*, $\text{SD}_i : T \rightarrow_{[G_1, G_2]} S_i$, where $0 < G_1 \leq G_2, S_i \in \mathcal{S}$, and T is the timestamp set. w-Variance constraints describe the required variance value δ for a w-length sequence. The value s_i on t_i is considered as an abnormal point if the variance of k intervals is over δ in w intervals involving s_i, *i.e.*, $s_{[i-\text{w}+1,i]}, s_{[i-\text{w}+2,i+1]}, ..., s_{[i,i+\text{w}-1]}$. After the detection, we use statistic-based methods as well as the SD solution to repair abnormal points with the maximum likelihood defined in our models.



(a) Correlation and pattern mining (b) Repairing abnormal data points (c) Repairing abnormal subsequences

Figure 3: System Interfaces

In abnormal subsequence repairing, we check pattern transitions in each divided sequence interval. Abnormal patterns are repaired by the known pattern transitions model. During each interval, we compute attributes correlation and detect candidate abnormal subsequences with Bollinger bands [8] computing. We then use LSTM training approach to determine and repair the real abnormal data.

4. DEMONSTRATIONS

We intend to demonstrate all the three data cleaning functions with the visualization of each cleaning step, and show how the system works with real-life industrial data.

Data sources. We demonstrate the whole data cleaning course with two real-life industrial time series.

(1) *Temperature control system data*. The data comes from a large-scale fossil-fuel power plant, with 80 attributes describing the water temperature control machine group. We process and analysis sequences on $1050K$ time points.

(2) *Fans group system data*. It is from a wind power plant, which has 150 attributes describing the working condition of fan-machine groups. It collects data each 5 seconds, and $1620K$ time points data have been used in the system.

Demo scenarios. Cleanits first allows users to input the clean sample data and the semantic domain knowledge. After that, users upload the industrial time series, and choose one cleaning mode. As shown in Fig.3(a), Cleanits first processes sequence pattern mining and correlation analysis on the time series, with logs begin to generate. Attributes with strong correlation are listed in the page, and different working patterns are recognized and shown in the right bar in Fig.3(a). Accordingly, different subsequence patterns are marked in different colors in each sequence.

The three cleaning functions are executed in order. In the missing imputation step, users are allowed to set the value range for sequence constraints and the window size in variance constraints. Cleanits repairs the missing data with proper methods discussed in Section 2. After that, the system begins to match inconsistent attribute values. Users can check the graphs of inconsistent subsequences and the matching result with a click on the corresponding box in the left-side column. Cleanits allows users to determine whether the matching result is correct. When it achieves consistency in attributes, the metrics computing (*e.g.*, statistical indicators measurement and correlation analysis) will be updated.

After both the missing values and the inconsistencies have been repaired, the anomaly detection starts. As shown in Fig.3(b), users can set parameters in the right-side bar. Abnormal data will be highlighted in the sequence shown in the middle of the page (Fig.3(b) and 3(c)). Thus, users know the anomaly part clearly via the sequence visualization. Figure 3(c) shows that Cleanits provides recommended repairing results and also allows users to determine the abnormal parts and repair the values manually. Accordingly, abnormal values modified by users will be considered as labelled sample data, which will be used to optimize the training models in cleaning functions for later data cleaning tasks.

5. ACKNOWLEDGMENTS

This work is supported by National Key Research and Development Program of China grant 2016YFB1000703; NSFC grant U1509216, U1866602, 61602129 and Microsoft Research Asia. Hongzhi Wang and Xiaoou Ding contributed to the work equally and should be regarded as co-first authors. Hongzhi Wang is the corresponding author.

6. REFERENCES

[1] Meir Toledano, Ira Cohen, Yonatan Ben-Simhon, and Inbal Tadeski. Real-time anomaly detection system for time series at scale. In *Proceedings of the KDD Workshop on Anomaly Detection*, pages 56–65, 2017.

[2] Aoqian Zhang, Shaoxu Song, Jianmin Wang, and Philip S. Yu. Time series data cleaning: From anomaly detection to anomaly repairing. *PVLDB*, 10(10):1046–1057, 2017.

[3] Aoqian Zhang, Shaoxu Song, and Jianmin Wang. Sequential data cleaning: A statistical approach. In *Proceedings of the International Conference on Management of Data, SIGMOD Conference*, pages 909–924, 2016.

[4] Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, and Christopher Ré. Holoclean: Holistic data repairs with probabilistic inference. *PVLDB*, 10(11):1190–1201, 2017.

[5] Lukasz Golab, Howard J. Karloff, Flip Korn, Avishek Saha, and Divesh Srivastava. Sequential dependencies. *PVLDB*, 2(1):574–585, 2009.

[6] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[7] Wei Yin, Tianbai Yue, Hongzhi Wang, Yanhao Huang, and Yaping Li. Time series cleaning under variance constraints. In *Database Systems for Advanced Applications - DASFAA International Workshops*, pages 108–113, 2018.

[8] Bar Koer. Bollinger bands approach on boosting abc algorithm and its variants. *Applied Soft Computing*, 49:292–312, 2016.