

# Cluster-based Data Filtering for Manufacturing Big Data Systems

Yifu Li<sup>1</sup>, Xinwei Deng<sup>2</sup>, Shan Ba<sup>3</sup>, William R. Myers<sup>4</sup>, William A. Brenneman<sup>4</sup>, Steve J. Lange<sup>4</sup>, Ron Zink<sup>4</sup>, and Ran Jin<sup>5</sup>

<sup>1</sup>School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK, USA, 73069

<sup>2</sup>Department of Statistics, Virginia Tech, Blacksburg, VA, USA, 24060

<sup>3</sup>Linkedin Corporation, 1101 W Maude Ave, Mountain View, CA, 94043

<sup>4</sup>Procter and Gamble Corporation

<sup>5</sup>Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA, USA, 24060

Correspondence author: Yifu Li, email: liyifu@ou.edu

## Abstract

A manufacturing system collects big and heterogeneous data for tasks such as product quality modeling and data-driven decision-making. However, as the size of data grows, timely and effective data utilization becomes challenging. We propose an unsupervised data filtering method to reduce manufacturing big data sets into informative small data sets. Furthermore, to determine the appropriate proportion of data to be filtered, we propose a filtering information criterion (FIC) to balance the trade-off between the filtered data size and the information preserved. The case study of a babycare manufacturing and a simulation study have shown the effectiveness of the proposed method.

*Keywords:* Big Data, Data Filtering, Data Quality, Hub Clustering, Smart Manufacturing.

## 1 Introduction

With the advancement of sensor and communication technologies, Industrial internet-based sensing systems can collect data over a long period of time from various manufacturing environments and machine settings. While such a sensing system is capable of collecting a large amount of data, there may contain redundant information, which significantly limits the quality and efficiency of data analysis (Kenett and Shmueli 2014). For example, a manufacturing process may generate a lot of sensor data from conformance manufacturing status, where the readings of the sensors have small variation through process modeling or control analysis, and little information can be extracted. As a result, selecting a meaningful or representative subset to reduce the overall size of the data while ensuring the quality of the subset is important to improve the efficiency of data analysis. Here the data quality is strongly related to the information richness, which can be measured by entropy (Gray 2011) and the performance of the data analysis. It is expected that a small but information-rich subset filtered from the massive raw data sets can effectively support various data analyses and reduce the time latency in computation without significantly sacrificing the performance of the analysis.

The objective of this work is to reduce the sample size of manufacturing big data while maintaining the amount of useful information for efficient data analysis. This work is motivated by a continuous baby-care manufacturing system, where the data were continuously collected as multi-dimensional time-series data. The cloud-storage is typically expensive for data from such a process as there will be more than three million data points collected from a single production line over one month. Under this motivation, we focus on proposing a method to reduce the information loss (i.e., entropy loss) when data is filtered/sampled for storage. After preliminary investigation, we found out that within a time window (e.g., several hours or days), the manufacturing process produces conforming products and the data collected may contain redundant information, e.g., constant values with little information for manufacturing analysis. Big data with redundancy pose challenges to efficient data storage and timely data analysis in a continuous manufacturing process. We believe that a properly filtered data set can preserve the majority of the original data set’s useful information leading to computational saving and improved data analysis performance.

Reducing the size of data while preserving the information was a major focus for data reduction under a data-rich environment (Liu et al. 2015; Xian et al. 2018). In this work, we study sampling methods that do not explicitly incorporate any data distribution assumption (Mak et al. 2018) or downstream task, such as regression modeling (Wang et al. 2019), into algorithm derivation. Along this direction, the most widely adopted approaches are probability-based filtering, which means that the probability of preserving each observation is pre-determined for filtering. Popular probability-based filtering methods include random sampling (Clarkson and Shor 1989; Liu et al. 2004) and stratified sampling (Trost 1986; Liberty et al. 2016). However, the probability-based data filtering methods often overlook inherent group structures or clustering patterns among data, and may easily preserve excessive data from large clusters, while ignoring small clusters.

In manufacturing, similar processes and equipment can produce clusters of data having similar information in terms of statistical moments (Yamaoka et al. 1978). Filtering guided by such clustering patterns can effectively reduce the information redundancy by removing a large number of observations with similar information. A natural outcome is to utilize data

clustering patterns for sampling. One recent work along this direction is Singh and Masuku (2014), which first clusters the raw data, then select a number of large data clusters to be fully preserved, while ignoring the smaller ones. However, one improvement opportunity in this work is that the clustering algorithm performed on the full data set may be over-complex, as the size of data can easily become too large to cluster. Furthermore, as the size of data is reduced through filtering, the information preserved will inevitably degrade and the resulting data analysis performance becomes worse. There is a pressing need to propose a method to optimize the selection of the filtering ratio, which is the proportion of data to be preserved after filtering, by considering the information loss.

We propose an unsupervised data filtering method along with a filtering information criterion (FIC) to automatically determine the proportion of data preserved in filtering. The proposed method aims to select representative subsets from raw data. Specifically, the unsupervised data filtering method includes two steps. The first step is to use certain index tags, such as time index tags, to segment the raw data into different segments (denoted as hubs) whenever there is a large gap between index tag values for two adjacent data observations. We assume that each hub has different characteristics compared with other hubs, as large index gaps usually indicate manufacturing events that impact the characteristics of *in situ* variables. An example of a large index gap can be caused by equipment shutdown, and the manufacturing process may run under different conditions after the equipment is restarted. The second step is to partition each hub into clusters, extract the centroid of each cluster, and perform cluster-wise random sampling. The proposed two-step method can recover the clustering pattern from raw data and help to better preserve the information by retaining the data from each cluster with the determined filtering ratio. Furthermore, the computational speed will be significantly accelerated by performing clustering within-hubs, instead of using the full data set.

To determine the best filtering ratio, we propose a filtering information criterion (FIC) to balance a trade-off between the information preserved and the size of the filtered data set. It is worth mentioning that the proposed sampling method does not rely on any data distribution assumption while deriving the analytical form of FIC relies upon data normality

assumption. Furthermore, although we are not considering filtering for categorical or discrete variables, we suggest using dummy variable transformation on them before applying the proposed method. A babycare manufacturing case study is used to evaluate the filtering performance based on the optimal filtering ratio selected by FIC. We further conducted simulation studies to systematically evaluate the filtering method at multiple levels of filtering ratios. The numerical results from studies show the promising performance of the proposed filtering method, compared with the benchmark methods, such as random sampling and stratified sampling.

The remainder of the paper is organized as follows. In Section 2, we introduce the proposed data filtering method and the FIC. In Section 3, we perform a case study in babycare manufacturing to test the proposed filtering method with FIC. In Section 4, we perform a simulation study inspired by the manufacturing data set for comparing the proposed data filtering method with other benchmark methods. In Section 5, we summarize the paper and discuss future work.

## 2 The Proposed Data Filtering Method

Denote the full manufacturing data set as  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}'$ , which contains  $n$  observations  $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})'$ ,  $i = 1, \dots, n$ . Here we assume that there are  $p$  continuous variables of interest (e.g., signals from  $p$  sensors). Each observation  $\mathbf{x}_i$  is often associated with its index tag, denoted as  $t_i$ . For example, the index tag can be the operation timestamp (a quantitative variable) or the current machine operational status (a categorical variable). Note that although the size of data,  $n$ , is usually very large, the corresponding manufacturing process conforms to the specifications for the majority of the time.  $\hat{\mathbf{X}}$  contains the raw data points or summary statistics (e.g., centroids of clusters for clustered data) from  $\mathbf{X}$ , to preserve the information in a smaller size. Here the data filtering can be formulated as,

$$\begin{aligned} & \min_{\hat{\mathbf{X}}} l(\hat{\mathbf{X}}, \mathbf{X}) \\ & s.t. \quad \frac{|\hat{\mathbf{X}}|}{n} \leq r, \end{aligned} \tag{2.1}$$

where  $l(\hat{\mathbf{X}}, \mathbf{X})$  is a loss function, such as negative log-likelihood or entropy loss (Gray 2011), to quantify the information loss between the filtered data and the full data. The  $|\cdot|$  is the cardinality or the sample size of a data set. Here  $0 \leq r \leq 1$  is a tunable filtering ratio.

To filter the raw data considering the data clustering patterns, we propose to incorporate clustering into the loss function in (2.1). For example, assume that  $\mathbf{X}$  can be partitioned into  $k$  clusters, where  $\mathbf{X} = \mathbf{C}_1 \cup \dots \cup \mathbf{C}_k$ . The data in different clusters are heterogeneous in terms of means, variances, etc. A representative subset of such a data set after the data filtering, denoted as  $\hat{\mathbf{X}} = \hat{\mathbf{C}}_1 \cup \dots \cup \hat{\mathbf{C}}_k$ , where  $\hat{\mathbf{C}}_i$  contains either the raw data or the summary statistic of  $\mathbf{C}_i$ . Therefore, we incorporate the clustering into (2.1) and propose cluster-based data filtering as

$$\begin{aligned} \min_{\hat{\mathbf{X}}, \boldsymbol{\mu}_i, \mathbf{C}_i, \hat{\mathbf{C}}_i} \quad & l(\hat{\mathbf{X}}, \mathbf{X}) + \lambda \sum_{i=1}^k \sum_{\mathbf{x}_j \in \mathbf{C}_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|_2^2 \\ \text{s.t.} \quad & \frac{\sum_{i=1}^k |\hat{\mathbf{C}}_i|}{\sum_{i=1}^k |\mathbf{C}_i|} \leq r, \end{aligned} \tag{2.2}$$

where  $\boldsymbol{\mu}_i$  is the centroid of each data cluster,  $\|\cdot\|_2$  is the Euclidean norm, and  $\lambda$  is the coefficient for the clustering term, which minimizes the within-cluster distances. Here, we assume that the size of data extracted from each cluster is proportional to the sample size of each cluster. Although the loss function in (2.2) is well-defined, there are two major computational challenges when using it in filtering. The first one is that the optimization problem involves finding the optimal data partitioning and sub-sampling, which is NP-hard (Burdakov et al. 2015). Furthermore, solving the multi-objective problem presented in loss function in (2.2) requires the selection of  $\lambda$ , which can significantly increase computational cost if iterative parameter tuning procedures are adopted (e.g., 5-fold cross-validation (Ojelund et al. 2002)).

To address these challenges, we propose a heuristic data filtering approach by combining the index-based data partition and the cluster-based data sampling. The method will divide the data into hubs and clusters and extract subsets of data from clusters. In the first step, we adopt the index-based data partition, which is computationally fast, to partition the full data into hubs and pave an efficient way to enable cluster-wise data filtering. In the second step, within each hub, we perform clustering and randomly sample each cluster of data proportionally. The proposed filtering method does not rely on any assumption about

data distribution, such as normality assumption.

## 2.1 Cluster-based Data Filtering Method

The index-based data partition consecutively partitions the full data along the index tags into  $q$  data hubs as  $\mathbf{X} = \mathbf{H}_1 \cup \dots \cup \mathbf{H}_q$ . Here,  $\mathbf{H}_j$ , where  $j = 1, \dots, q$ , are consecutive and non-overlapping data subsets of  $\mathbf{X}$  over time. When index tags are categorical variables, it is straight-forward to hub partition, as each unique value of a categorical variable will form one data hub. However, for continuous variables, it is not intuitive, and we propose a quantile-based method for hub partition. The data partition (segmentation) for continuous variables has many variants, such as using a likelihood criterion (Guralnik and Srivastava 1999), minimum message length approximation (Fitzgibbon et al. 2002), and landmark identification (Ibragimov et al. 2014). Note that for manufacturing processes with index tags, it is not easy to assume certain probabilistic distribution properties for the index tag variable. Thus, segmentation based on likelihood and message length approximation is not applicable to our problem. Alternatively, we adopt the idea of finding landmarks (Perng et al. 2000) or perceptually important points (PIPs) (Zhang et al. 2007) based on index tags for segmentation.

Note that the index tags in the manufacturing data often reflect the dynamics of a manufacturing system. A large gap between two consecutive index tags often reflects a change in the manufacturing system. For example, when the operation times are used as the index tags, a gap for several hours between two consecutive observations may indicate that there is a product change-over. The aforementioned manufacturing events can significantly vary the *in situ* conditions of manufacturing processes, and generate data in hubs. Partitioning the data into hubs will reduce the complexity of clustering, compared with the clustering in the full data set. Although practitioners can always use domain knowledge to choose significant gaps among data hubs, such as identifying the process change-over, maintenance events, product receipt changes etc. However, we would like to propose a data-driven rule to identify significant gaps among data hubs when there is no such domain knowledge existed. Specifically, to incorporate the information on the gaps of index tags in the proposed method,

we consider to use the second-order tag gap at  $t_i$ , defined as  $\delta_i = (t_{i+1} - t_i) - (t_i - t_{i-1})$ . When time tags of data collection are adopted as the index tags, the second-order tag gap can identify the large index tag gaps, either in the time domain (the first-order information) or the frequency domain (the second-order information) among index tags. We consider to form the segments by partitioning the raw data  $X$  at the following locations

$$\{j : \delta_j \geq q_{1-\alpha}(\delta_1, \dots, \delta_n)\}, \quad (2.3)$$

where  $q_{1-\alpha}(\delta_1, \dots, \delta_n)$  is the  $(1 - \alpha)$  percentile of all  $\delta_1, \dots, \delta_n$ . It means that the gap values at these locations are larger than the  $(1 - \alpha)$  percentile of all  $\delta_j$ 's.

For each identified hub, we cluster it into  $k_j$  clusters for hub  $\mathbf{H}_j$  as  $\mathbf{H}_j = \mathbf{C}_1^{(j)} \cup \dots \cup \mathbf{C}_{k_j}^{(j)}$ . Here we adopt the k-means clustering method (MacQueen et al. 1967) to form clusters  $\mathbf{C}_1^{(j)}, \dots, \mathbf{C}_{k_j}^{(j)}$  for each hub. Specifically, the k-means clustering method minimizes the total within-cluster sum of squares for the clusters in a hub as

$$\min \sum_{s=1}^{k_j} \sum_{\mathbf{x}_i \in \mathbf{C}_s^{(j)}} \|\mathbf{x}_i - \boldsymbol{\mu}_s^{(j)}\|_2^2, \quad (2.4)$$

where  $\boldsymbol{\mu}_s^{(j)}$  is the centroid of the cluster  $s$  from the hub  $\mathbf{H}_j$ . The number of clusters  $k_j$  for each hub is selected to maximize the average Silhouette distance of all data points in the hub (Rousseeuw 1987). Given any data point  $\mathbf{x}_i$ , the Silhouette distance calculates the difference between the average distance of  $\mathbf{x}_i$  to the other data points in the same cluster and the average distance of  $\mathbf{x}_i$  to the other data points in different clusters. The Silhouette distance is adopted because it is a state-of-the-art method at measuring how well that data points are matched to their respective clusters, hence evaluates the overall clustering performance at any given  $k_j$  (Tomasev et al. 2013). Finally, we generate samples  $\hat{\mathbf{C}}_s^{(j)}$  from each cluster  $\mathbf{C}_s^{(j)}$  to form the filtered data set  $\hat{\mathbf{X}} = \bigcup_{j=1}^q \bigcup_{s=1}^{k_j} \hat{\mathbf{C}}_s^{(j)}$ , where  $\hat{\mathbf{C}}_s^{(j)}$  consist of  $100r\%$  data points randomly sampled from cluster  $\mathbf{C}_s^{(j)}$  plus the cluster centroid  $\boldsymbol{\mu}_s^{(j)}$ . Here, the centroids are important summary statistics of clusters, and adding them to the filtered dataset is crucial for the proposed method to preserve representative information of clusters.



## 2.2 Selection of the filtering ratio $r$

The filtering ratio  $r$  determines the degradation of data quality. For example, the entropy loss quantitatively measures the information loss between the raw and filtered data sets, which is defined as  $EL = \text{trace}(\mathbf{\Sigma}^{-1}\hat{\mathbf{\Sigma}}) - \log(\det(\mathbf{\Sigma}^{-1}\hat{\mathbf{\Sigma}})) - p$ , where  $\text{trace}(\cdot)$  is the trace operator,  $\det(\cdot)$  is the determinant operator,  $\mathbf{\Sigma}$  is the sample covariance matrix of the raw data set,  $\hat{\mathbf{\Sigma}}$  is the sample covariance matrix of the filtered data set, and  $p$  is the number of variables in the data set.

Determining the filtering ratio  $r$  is not trivial. For example, in P&G babycare manufacturing data with a size of  $n = 24915$  observations and  $p = 32$  variables, the increase of loss due to the decrease of data size preserved is non-linear. In Figure 1, we compared the entropy loss acquired by random sampling under different filtering ratios. From Figure 1 we can see that the entropy loss remains stable when the filtering ratio is larger than  $r = 0.2$ , but increased dramatically when the filtering ratio was smaller than  $r = 0.2$ . As a result,  $r = 0.2$  may be considered as a good choice of the filtering ratio, since it has a balanced small filtering ratio with a relatively small entropy loss. We will make use of this observation to test if our proposal can identify the optimal filtering ratio.

In practice, the exact trade-off relationship between the filtering ratio and the entropy loss among data sets may vary significantly. The filtering ratio  $r$  can be determined by users based on their experience. However, there is a lack of statistical justification behind the *ad hoc* selection of the filtering ratio for achieving a good balance between the information preserved and the size of the filtered data  $\hat{\mathbf{X}}$ .

Here we propose a statistical framework on the selection of optimal value of  $r$ . Specifically, we propose FIC, as the criterion to find the optimal value of  $r$ . The FIC is motivated and modified from Akaike information criterion (AIC), a statistical model selection method (Akaike et al. 1998). Assuming that the data set  $\mathbf{X}$  has  $n$  observations and follows the independent and identically distributed normal distribution  $N(\boldsymbol{\mu}, \mathbf{\Sigma})$ , with  $\boldsymbol{\mu}$  as the mean

and  $\Sigma$  as the covariance matrix, then the log-likelihood of the full data can be written as

$$\begin{aligned} l(X|\boldsymbol{\mu}, \Sigma) &= -n\frac{1}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \\ &= -n\frac{1}{2}\log|\Sigma| - n\frac{1}{2}\text{trace}(\Sigma^{-1}\mathbf{S}), \end{aligned} \quad (2.5)$$

up to some constant and where

$$\mathbf{S} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})'.$$

The original AIC is to consider the balance between the fitness of the model to the data and the model complexity, which can be expressed as

$$AIC(\boldsymbol{\mu}, \Sigma) = -2l + 2k, \quad (2.6)$$

where  $l$  is the log likelihood function and  $k$  is the number of estimated parameters in the model.

We propose to use an AIC-like measure to find the optimal ratio  $r$ . To do this, we estimate the mean and covariance matrix from the filtered data, which are conducted by using the weighted sample mean and sample covariance matrix from the full data

$$\begin{aligned} \hat{\boldsymbol{\mu}} &= \frac{1}{m} \sum_{\mathbf{x}_i \in \hat{X}} \mathbf{x}_i = \frac{1}{m} \sum_{i=1}^n w_i \mathbf{x}_i, \\ \hat{\Sigma} &= \frac{1}{m} \sum_{\mathbf{x}_i \in \hat{X}} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})' = \frac{1}{m} \sum_{i=1}^n w_i (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})', \end{aligned}$$

where  $w_i \in \{0, 1\}$ ,  $\sum_{i=1}^n w_i = m$ , the total number of preserved data points,  $\hat{\boldsymbol{\mu}}$  and  $\hat{\Sigma}$  are the mean and covariance generated from the filtered data set, then a modified AIC, denoted as filtering information criterion (FIC), for evaluating the quality of the filtered data can be written as

$$\begin{aligned} FIC(\hat{\boldsymbol{\mu}}, \hat{\Sigma}) &= -2l(X|\hat{\boldsymbol{\mu}}, \hat{\Sigma}) + 2|\hat{X}| \\ &= n \log|\hat{\Sigma}| + \sum_{i=1}^n (\mathbf{x}_i - \hat{\boldsymbol{\mu}})' \hat{\Sigma}^{-1} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}) + 2|\hat{X}| \\ &= n \left[ \log|\hat{\Sigma}| + \text{trace}(\hat{\Sigma}^{-1}\mathbf{S}^*) + 2\frac{|\hat{X}|}{n} \right], \end{aligned} \quad (2.7)$$

where  $\mathbf{S}^* = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})'$ .

Given a pre-defined set  $\mathbf{r} = \{r_1, \dots, r_m\}$ , the optimal filtering ratio  $r^*$  is chosen to have the smallest corresponding FIC value resulted from the filtering. In the case study and simulation that follows, we will evaluate the performance of the proposed filtering method and the FIC.

A pseudo-code for the proposed filtering method is summarized as follows.

---

**Algorithm 1** Pseudo code for the proposed filtering method

---

Step 1: Split raw data into hubs  $\mathbf{H}_j$ , where  $j = 1, \dots, q$  satisfying (2.3).

Step 2: Identify  $k_j$  clusters  $\mathbf{C}_s^{(j)}$ , where  $s = 1, \dots, k_j$ , within each hub  $\mathbf{H}_j$  using k-means clustering and Silhouette distance.

Step 3: Randomly sample from each cluster  $\mathbf{C}_s^{(j)}$  based on the filtering ratio  $r$ :

**for** each hub  $\mathbf{H}_j$ , where  $j = 1, \dots, q$ , **do**

**for** each cluster  $\mathbf{C}_s^{(j)}$ , where  $s = 1, \dots, k_j$ , **do**

        Randomly sample 100 $r$ % data points and the cluster centroid  $\boldsymbol{\mu}_s^{(j)}$  from cluster  $\mathbf{C}_s^{(j)}$ .

        Denote these observations for each cluster as  $\hat{\mathbf{C}}_s^{(j)}$ .

Step 4: Combine all cluster  $\hat{\mathbf{C}}_s^{(j)}$  and form the filtered data  $\hat{\mathbf{X}} = \bigcup_{j=1}^q \bigcup_{s=1}^{k_j} \hat{\mathbf{C}}_s^{(j)}$ .

---

We would remark that although the derivation of FIC starts with the normal assumption, one can use FIC for data from the exponential family distribution with proper mean and covariance matrix. Note that the formulation in (2.7) can be viewed as an information discrepancy between  $\hat{\Sigma}$  and  $S^*$ . Following this observation, it is possible to use FIC as a general guideline to select the optimal filtering ratio for data with proper mean and covariance matrix. The analysis of the case study provides further evidence on the use of FIC.

### 3 Case Study

In collaboration with the P&G baby care manufacturing sector, we evaluate the performance of the proposed method. A data collected from the P&G production line has a size of

$n = 24915$  observations and  $p = 32$  variables including the time tag. Although the physical meanings of variables are not disclosed here due to the data non-disclosure agreement, we summarized the summary statistics and the histogram of each centralized variable in the supplementary materials. Typically, a manufacturing big data system can have millions of observations for analysis. This selected data set is to provide a representative performance evaluation in a big data environment for two reasons. The first reason is that some engineering-driven partition can divide a big data set into sub-data with smaller sample sizes for filtering. The second reason is that production engineers often avoid waiting until the collected data for analysis becomes big sizes, which can cause a significant delay in decision-making.

The objective of this case study is two-fold. First, we demonstrate that an effective filtering ratio can be selected by FIC to achieve a good balance between the size of data preserved and the data quality. Second, we evaluate the performance of the proposed filtering method in comparison with several benchmark methods.

We used the entropy loss (Gray 2011) to evaluate the entropy loss of the filtered data sets compared with the raw data efficiently. Two benchmark methods, denoted as  $BM_1$  and  $BM_2$  were included for comparison. The  $BM_1$  is random sampling (Liu et al. 2004) which extracts observations randomly from the full data set, and the  $BM_2$  is stratified sampling (Liberty et al. 2016) which extracts one observation within every equally spaced and non-overlapping time window. For example, given a filtering ratio  $r$  and  $n$  data points, we stratified the raw data into approximately  $nr$  consecutive and equally sized data segments and preserved the first observation of each segment. The threshold for forming hubs by segmentation in (2.3) set  $\alpha = 0.01$ . To illustrate the impact of different  $\alpha$  on gap identification, we compare the gaps above the  $(1 - \alpha)$  percentile with  $\alpha = 0.02, 0.01, 0.001$  versus on the gaps identified in the natural log space in the appendix. To evaluate the performance of the proposed filtering method, 50 replications with 90% of randomly extracted data were performed on varying and fixed filtering ratios to ensure reproducibility. Here the time order of the randomly extracted data is preserved in each replication.

In Figure 2, we summarized the average entropy loss for each filtering method with vary-

ing (Figure 2(a)) and fixed (Figure 2(b)) filtering ratios. The mean value and the standard error of the optimal filtering ratios selected by FIC over 50 replications are symbolized as solid and dashed vertical lines respectively in Figure 2(a). In Figure 2(a), the entropy loss appears not to improve significantly when the filtering ratio is larger than 0.4 and increased dramatically as the filtering ratio is smaller than 0.1. As a result, the desirable filtering ratio should be between 0.1 and 0.4. In Figure 2(a), the average optimal filtering ratio was 0.235, which is within the desirable range (0.1-0.4) with very small standard errors. Furthermore, we observed that the cluster-based filtering method (the solid curve in Figure 2(a)) showed superior performance compared with the benchmark methods at the selected filtering ratio. We further use the same optimal filtering ratio determined by FIC for the proposed filtering method to the two benchmark methods to compare the entropy loss over 50 replications. As a summary, the boxplots in Figure 2(b) show that the cluster-based filtering method provides a significantly smaller entropy loss than the benchmark methods.

We also summarize the filtering performance for different methods when the filtering ratios are determined by FIC over 50 replications. Specifically, we used FIC to select the optimal filtering ratios, which yielded the lowest FIC scores for the three filtering methods, e.g., random sampling, stratified sampling, the proposed method, in each replication. The optimal filtering ratios and the corresponding entropy loss from the optimal filtering ratios are summarized in the boxplots shown in Figure 3. Figure 3(a) shows the boxplot of the selected filtering ratios over 50 replications. Figure 3(b) shows the boxplot of the entropy loss obtained under the selected filtering ratios over 50 replications. We can see that the proposed filtering method favored higher filtering ratios, which were between 0.2-0.5, while the benchmark filtering methods favored much smaller filtering ratios (0.01-0.05). As FIC jointly leverages the likelihood and the size of data preserved, it is straightforward to see that FIC favors preserving more data points for the proposed filtering method. As a result, we conclude that the proposed filtering method can efficiently preserve the likelihood with higher filtering ratios to yield the lowest FIC. On the other hand, a much smaller filtering ratio is preferred by FIC for the benchmark filtering methods as they are less effective at preserving the likelihood where only decreasing the filtering ratio will lead to a lower FIC score. Furthermore, although

the cluster-based filtering method had a relatively higher filtering ratio, it offered much smaller entropy loss. We summarize that: (1) the filtering information criterion (FIC) can successfully help to identify the optimal filtering ratio, balancing the trade-off between the entropy loss and the size of data preserved; (2) the cluster-based filtering method can more efficiently preserve the data likelihood compared to the benchmark filtering methods; and (3) the FIC will favor saving more data when the underlying filtering methods can better preserve the likelihood.

Furthermore, to show that the proposed method can be easily applied to dataset with much larger size, we performed hub identification using  $\alpha = 0.01$  on a dataset with  $n = 24,915$  (the case study data) and a dataset with  $n = 1,003,708$  data points from the P&G babycare manufacturing process. The result is that hub identification generated data hubs in a comparably average size of 199.902 and 196.1811 from two datasets. Such similar hub sizes indicate that the scale-up of the proposed method to new data sets to the same process can be straight-forward since hub identification partitions raw data in significantly different sizes into similar-sized hubs for further processing.

## 4 The Simulation Study

We further evaluated the filtering performance in a simulation study. In particular, we considered varying four simulation settings: the inherent number of data clusters, the signal-to-noise ratios, the model sparsity, and the distributions that variables follow. To mimic the characteristics of the real manufacturing data set, the simulation data were generated as follows: for two settings on the inherent number of clusters, we respectively clustered the real data set into  $NC$  clusters using k-means clustering, each denoted as  $\hat{\mathbf{X}}_f$ . Then we extracted the original time stamp for each observation in the cluster  $f$  as  $\mathbf{t}_f$  and the means of all variables as  $\boldsymbol{\mu}_f$ , where  $f = 1, \dots, NC$  from real data. Then we simulated each data cluster with 1.  $\mathbf{X}_f \sim N(\boldsymbol{\mu}_f, \Sigma)$ , 2.  $\mathbf{X}_f \sim t(\boldsymbol{\mu}_f, \Sigma, df = 10)$ , and 3.  $\mathbf{X}_f \sim Uniform(\boldsymbol{\mu}_f)$  in three different scenarios,  $\mathbf{X}_f$  has the same size as  $\hat{\mathbf{X}}_f$ , and the regularized covariance matrix  $\Sigma = [\sigma_{ij}]$  with  $\sigma_{ij} = 1$  when  $i = j$ , and  $\sigma_{ij} = 0.3$  when  $i \neq j$ . To investigate the different impact of covariance structure, we can investigate the results from datasets

following t-distribution with the same covariance matrix and Uniform distribution, which did not rely on covariance. Finally, we aggregated all simulated clusters  $\mathbf{X}_f$  with the time stamps  $\mathbf{t}_f$  to produce the simulated data matrix  $\mathbf{X}$ .

We further created simulation models to evaluate the modeling performance. To generate the response variable  $\mathbf{y}$ , we used a linear model given by  $\mathbf{y} = \tilde{\mathbf{X}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ , where  $\mathbf{y} = (y_1, \dots, y_n)'$  was the response,  $\tilde{\mathbf{X}}$  was the simulated data matrix with all the main effect variables  $(t, x_1, \dots, x_p)'$  in  $\mathbf{X}$  and the two-factor interactions in the multiplication form  $(tx_1, \dots, x_{p-1}x_p)'$ .  $\boldsymbol{\beta} = (\beta_t, \beta_1, \dots, \beta_p, \beta_{t1}, \dots, \beta_{(p-1)p})'$  is the model parameter vector corresponding to main effect variables and the interaction terms,  $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)'$  were the residual terms with  $\varepsilon_i \sim N(0, \sigma^2)$  as independent and identically distributed. To obtain the value of  $\boldsymbol{\beta}$ , we first trained a linear model using the real data set, and extracted the model coefficients as  $\hat{\boldsymbol{\beta}}$ . Then, we randomly set  $MS\%$  of main effect parameters  $(\beta_t, \beta_1, \dots, \beta_p)'$  and 10% of interaction term parameters  $(\beta_{t1}, \dots, \beta_{(p-1)p})'$  in  $\boldsymbol{\beta}$  as significant (non-zeros), with values randomly chosen from  $\hat{\boldsymbol{\beta}}$ . Furthermore, we varied the signal-to-noise ratio ( $SNR$ ) when generating the error term  $\boldsymbol{\varepsilon}$ , where  $SNR = \frac{var(\tilde{\mathbf{X}}\boldsymbol{\beta})}{var(\boldsymbol{\varepsilon})}$ , and  $var(\cdot)$  represents the variance (Friedman et al. 2001). In summary, two levels of the three factors are shown in Table 1.

We evaluated the performance of the proposed filtering method with 90% of data randomly extracted from the raw data over 50 replications to ensure reproducibility. Same as the case study, We normalized all the variables in this study. Instead of optimizing the filtering ratio, we evaluate the performance measures when different filtering ratios are used. Specifically, we selected three levels of the pre-determined filtering ratios to filter the training data in each replication. The  $\alpha$  value for identifying the significant second-order index gap in (2.3) is set to be  $\alpha = 0.01$ . Then the filtered data sets were used to estimate a linear model for the evaluation goodness-of-fit.

Five performance measures were used including entropy loss (Gray 2011),  $R^2$ , adjusted- $R^2$ , the CPU time for the data filtering step, and the CPU time for the modeling step.  $R^2$  quantifies the goodness-of-fit for the model on the filtered training data while the adjusted- $R^2$  (Gelman and Pardoe 2006) simultaneously evaluates the goodness-of-fit and the complexity of the model. Besides the previously adopted random sampling and stratified sampling

methods, the third benchmark method was to directly use the full data set for modeling. Using the full data yielded zero entropy loss but is not necessarily the best model performance. The simulation was performed on a workstation with CPU Xeon Processor E5-2687W, 3.10 GHz, 64 GB RAM.

For the simplicity of presentation, we only include the tables of the results with varying inherent number of clusters in the manuscript (Table 2 and Table 3), while presenting the rest of tables corresponding to the other scenarios in the supplementary materials. Both tables show that the proposed method significantly outperformed the two benchmark methods on entropy loss as the filtering ratio decreased. Furthermore, the proposed method outperformed all methods in comparison on  $R^2$  and adjusted- $R^2$ . As the inherent number of clusters increased from 30 (Table 2) to 1000 (Table 3), each cluster contained less data and information. As a result, the clustering pattern is becoming less significant in data composition and the modeling performance is less affected by such a pattern. However, the proposed method always outperformed the benchmark filtering methods for the majority of performance measures in both scenarios. Although the proposed filtering method took a longer time at the filtering step, it costs less than 10 seconds on average and was deemed as a time-efficient method by production engineers. The rest of the results, corresponding to the Normal, t, and Uniform distribution, show that the proposed methods outperformed benchmark methods for the majority of scenarios at reducing entropy loss and achieving better goodness-of-fit.

Additionally, we generated the Figure 4 based on the information from Table 2. It is seen that there was a significant reduction of entropy loss achieved by the proposed filtering method over the two benchmark filtering methods. The proposed filtering method outperformed the benchmark methods for the adjusted  $R^2$  as well. Such improvements became more significant as the filtering ratio decreased.

## 5 Discussion

As sensor and communication technologies advance, Industrial Internet-based sensing systems are capable of collecting massive data for process modeling and control. However, such



systems also generate a lot of redundant information which significantly limits the quality and efficiency of the data analysis. As a result, extracting representative and high-quality data subsets is important to improve the efficiency of the data analysis. In this work, we proposed a filtering method and new criteria (FIC) to facilitate data analysis by selecting a small but effective data subset. Specifically, the proposed method partitions raw data into clusters and proportionally extracts a data subset from each cluster. The proposed cluster-based data filtering method outperformed the benchmark filtering methods on performance measures, such as information loss, the modeling goodness-of-fit in the case study and the simulation. Furthermore, the new filtering ratio selection criteria (FIC) has shown its effectiveness in terms of balancing the trade-offs between the size of data preserved in filtering and the quality of the filtered data.

Although we focus on reducing the sample size in this paper, we can combine the proposed method with other variable screening methods when both the sample size and the number of variables are large. State-of-the-art variable screening methods include LASSO (Tibshirani et al. 2005), Elastic Net (Zou and Hastie 2005), sure independence screening (Fan and Lv 2008). This paper leads to a few future research directions. Data analysis under big data environments have become prevalent and even sometimes necessary because of the computational performance requirement. However, efficiently utilizing big data through analysis that is both time-efficient and accurate is still a challenging problem. The proposed method can also facilitate internet-of-things (IoT)-based data collection, communication, storage, and analysis. For example, as the inline data de-duplication (real-time data filtering for continuous data streaming), has become more popular in recent years (Zhou et al. 2013), future work will focus on the conversion of the current offline data filtering to online. Specifically, we need to investigate when to update the filtering ratio due to process changes, product changes, etc. Furthermore, an integer programming heuristic or relaxation technique (Schrijver 1998) can be derived to help directly solve the data filtering problem shown in the objective function (2.1) so that the filtering performance may be further improved. Lastly, we will investigate filtering for functional data (Sun et al. 2017) and imaging data (Li et al. 2019) as *in situ* data measurement for manufacturing.

## **6 Acknowledgement**

This work was supported by Procter & Gamble; and the National Science Foundation under Grant CMMI-1634867.

## **7 Declaration of Interest**

No potential conflict of interest was reported by the authors.

## References

- Akaike, H., Parzen, E., Tanabe, K., and Kitagawa, G. 1998. *Selected Papers of Hirotugu Akaike*. New York: Springer Science & Business Media.
- Burdakov, O., Kanzow, C., and Schwartz, A. 2015. On a Reformulation of Mathematical Programs with Cardinality Constraints. *Advances in Global Optimization*, 3–14.
- Clarkson, K. L. and Shor, P. W. 1989. Applications of Random Sampling in Computational Geometry, II. *Discrete & Computational Geometry* 4: 387–421.
- Fan, J. and Lv, J. 2008. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70: 849–911.
- Fitzgibbon, L. J., Dowe, D. L., and Allison, L. 2002. Change-point Estimation Using New Minimum Message Length Approximations. *Proceedings of Pacific Rim International Conference on Artificial Intelligence*, 244–254.
- Friedman, J., Hastie, T., and Tibshirani, R. 2001. *The Elements of Statistical Learning*. New York: Springer.
- Gelman, A. and Pardoe, I. 2006. Bayesian Measures of Explained Variance and Pooling in Multilevel (Hierarchical) Models. *Technometrics* 48: 241–251.
- Gray, R. M. 2011. *Entropy and Information Theory*. Heidelberg: Springer Science & Business Media.
- Guralnik, V. and Srivastava, J. 1999. Event Detection from Time Series Data. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 33–42.
- Ibragimov, B., Likar, B., Pernuš, F., and Vrtovec, T. 2014. Shape Representation for Efficient Landmark-based Segmentation in 3-D. *IEEE Transactions on Medical Imaging* 33: 861–874.

- Kenett, R. S. and Shmueli, G. 2014. On Information Quality. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 177: 3–38.
- Li, Y., Sun, H., Deng, X., Zhang, C., Wang, H. P., and Jin, R. 2019. Manufacturing Quality Prediction using Smooth Spatial Variable Selection Estimator with Applications in Aerosol Jet® Printed Electronics Manufacturing. *IISE Transactions* just-accepted: 1–17.
- Liberty, E., Lang, K., and Shmakov, K. 2016. Stratified Sampling Meets Machine Learning. *Proceedings of 33rd International Conference on Machine Learning*, 2320–2329.
- Liu, H., Sadygov, R. G., and Yates, J. R. 2004. A Model for Random Sampling and Estimation of Relative Protein Abundance in Shotgun Proteomics. *Analytical Chemistry* 76: 4193–4201.
- Liu, K., Mei, Y., and Shi, J. 2015. An adaptive sampling strategy for online high-dimensional process monitoring. *Technometrics* 57: 305–319.
- MacQueen, J. et al. 1967. Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.
- Mak, S., Joseph, V. R., et al. 2018. Support points. *The Annals of Statistics* 46: 2562–2592.
- Ojelund, H., Brown, P. J., Madsen, H., and Thyregod, P. 2002. Prediction based on mean subset. *Technometrics* 44: 369–378.
- Perng, C., Wang, H., Zhang, S., and Parker, D. S. 2000. Landmark: A New Technique for Similarity-based Pattern Querying in Time Series Databases. *Proceedings of International Conference on Data Engineering*, 1–17.
- Rousseeuw, P. J. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics* 20: 53–65.
- Schrijver, A. 1998. *Theory of Linear and Integer Programming*. New York, NY: John Wiley & Sons.

- Singh, A. S. and Masuku, M. B. 2014. Sampling Techniques & Determination of Sample Size in Applied Statistics Research: An Overview. *International Journal of Economics, Commerce and Management* 2: 1–22.
- Sun, H., Huang, S., and Jin, R. 2017. Functional Graphical Models for Manufacturing Process Modeling. *IEEE Transactions on Automation Science and Engineering* 14: 1612–1621.
- Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67: 91–108.
- Tomasev, N., Radovanovic, M., Mladenic, D., and Ivanovic, M. 2013. The role of hubness in clustering high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering* 26: 739–751.
- Trost, J. E. 1986. Statistically Nonrepresentative Stratified Sampling: A Sampling Technique for Qualitative Studies. *Qualitative Sociology* 9: 54–57.
- Wang, H., Yang, M., and Stufken, J. 2019. Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association* 114: 393–405.
- Xian, X., Wang, A., and Liu, K. 2018. A nonparametric adaptive sampling strategy for online monitoring of big data streams. *Technometrics* 60: 14–25.
- Yamaoka, K., Nakagawa, T., and Uno, T. 1978. Statistical Moments in Pharmacokinetics. *Journal of Pharmacokinetics and Biopharmaceutics* 6: 547–558.
- Zhang, Z., Jiang, J., and Wang, H. 2007. A New Segmentation Algorithm to Stock Time Series Based on PIP Approach. *Proceedings of 2017 Wireless Communications, Networking and Mobile Computing*, 5609–5612.
- Zhou, R., Liu, M., and Li, T. 2013. Characterizing the Efficiency of Data Deduplication for Big Data Storage Management. *Proceedings of 2013 Workload Characterization, IEEE International Symposium*, 98–108.

Zou, H. and Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)* 67: 301–320.

## 8 Tables

Table 1: Simulation Setting Summary

Parameters	Low	High
Number of Clusters ( $NC$ )	30	1000
Model Sparsity ( $MS$ )	0.3	0.7
Signal-to-Noise Ratio ( $SNR$ )	3	10

<sup>1</sup> The same settings were applied on Normal, t, and Uniform distribution

Table 2: Means and standard errors (within parenthesis) of five performance measures based on 50 simulation replications for Normal distribution with  $NC = 30$ ,  $MS = 0.3$ , and  $SNR =$

3

Filtering ratio	Filtering Method	Entropy Loss	$R^2$	Adjusted- $R^2$	Filtering Time (in seconds)	Modeling Time (in seconds)
0.1	Full Data	—	0.748	0.748	—	39.664
		—	( 0.000 )	( 0.000 )	—	( 0.220 )
	$BM_1$	0.270	0.750	0.750	<b>0.106</b>	<b>10.073</b>
		( 0.005 )	( 0.001 )	( 0.001 )	( 0.008 )	( 0.146 )
	$BM_2$	0.247	0.752	0.752	0.123	10.292
	( 0.004 )	( 0.001 )	( 0.001 )	( 0.009 )	( 0.223 )	
	Proposed	<b>0.237</b>	<b>0.753</b>	<b>0.753</b>	9.496	10.359
		( 0.003 )	( 0.001 )	( 0.001 )	( 0.178 )	( 0.154 )
0.01	Full Data	—	0.748	0.748	—	39.664
		—	( 0.000 )	( 0.000 )	—	( 0.220 )
	$BM_1$	5.194	0.764	0.762	0.052	0.725
		( 0.111 )	( 0.005 )	( 0.005 )	( 0.004 )	( 0.013 )
	$BM_2$	4.987	0.768	0.765	<b>0.051</b>	0.745
	( 0.108 )	( 0.005 )	( 0.005 )	( 0.004 )	( 0.013 )	
	Proposed	<b>3.849</b>	<b>0.789</b>	<b>0.787</b>	9.622	<b>0.717</b>
		( 0.056 )	( 0.006 )	( 0.006 )	( 0.179 )	( 0.017 )
0.005	Full Data	—	0.748	0.748	—	39.664
		—	( 0.000 )	( 0.000 )	—	( 0.220 )
	$BM_1$	10.875	0.774	0.769	<b>0.049</b>	0.457
		( 0.298 )	( 0.009 )	( 0.010 )	( 0.003 )	( 0.011 )
	$BM_2$	10.790	0.791	0.786	0.068	0.484
	( 0.281 )	( 0.009 )	( 0.009 )	( 0.007 )	( 0.010 )	
	Proposed	<b>7.312</b>	<b>0.833</b>	<b>0.829</b>	9.595	<b>0.403</b>
		( 0.101 )	( 0.011 )	( 0.011 )	( 0.176 )	( 0.012 )
0.005	Full Data	—	0.748	0.748	—	39.664
		—	( 0.000 )	( 0.000 )	—	( 0.220 )
	$BM_1$	10.875	0.774	0.769	<b>0.049</b>	0.457
		( 0.298 )	( 0.009 )	( 0.010 )	( 0.003 )	( 0.011 )
	$BM_2$	10.790	0.791	0.786	0.068	0.484
	( 0.281 )	( 0.009 )	( 0.009 )	( 0.007 )	( 0.010 )	
	Proposed	<b>7.312</b>	<b>0.833</b>	<b>0.829</b>	9.595	<b>0.403</b>
		( 0.101 )	( 0.011 )	( 0.011 )	( 0.176 )	( 0.012 )



Table 3: Means and standard errors (within parenthesis) of five performance measures based on 50 simulation replications for Normal distribution with  $NC = 1000$ ,  $MS = 0.3$ , and  $SNR = 3$

Filtering ratio	Filtering Method	Entropy Loss	$R^2$	Adjusted- $R^2$	Filtering Time (in seconds)	Modeling Time (in seconds)
0.1	Full Data	—	0.750	0.750	—	39.821
		—	( 0.000 )	( 0.000 )	—	( 0.169 )
	$BM_1$	0.319	0.753	0.753	<b>0.104</b>	7.680
		( 0.013 )	( 0.001 )	( 0.001 )	( 0.007 )	( 0.104 )
	$BM_2$	<b>0.261</b>	0.752	0.752	0.129	<b>7.589</b>
	( 0.006 )	( 0.001 )	( 0.001 )	( 0.010 )	( 0.110 )	
	Proposed	0.302	<b>0.754</b>	<b>0.754</b>	10.019	7.705
		( 0.013 )	( 0.001 )	( 0.001 )	( 0.337 )	( 0.096 )
0.01	Full Data	—	0.750	0.750	—	39.821
		—	( 0.000 )	( 0.000 )	—	( 0.169 )
	$BM_1$	3.836	0.785	0.782	<b>0.050</b>	0.721
		( 0.254 )	( 0.005 )	( 0.005 )	( 0.003 )	( 0.012 )
	$BM_2$	3.992	0.772	0.769	0.062	0.743
	( 0.245 )	( 0.005 )	( 0.005 )	( 0.007 )	( 0.012 )	
	Proposed	<b>3.203</b>	<b>0.792</b>	<b>0.789</b>	9.994	<b>0.649</b>
		( 0.176 )	( 0.009 )	( 0.009 )	( 0.334 )	( 0.009 )
0.005	Full Data	—	0.750	0.750	—	39.821
		—	( 0.000 )	( 0.000 )	—	( 0.169 )
	$BM_1$	6.812	0.791	0.786	<b>0.054</b>	0.431
		( 0.478 )	( 0.008 )	( 0.008 )	( 0.005 )	( 0.010 )
	$BM_2$	8.652	0.788	0.782	0.058	0.505
	( 0.740 )	( 0.010 )	( 0.010 )	( 0.005 )	( 0.008 )	
	Proposed	<b>5.480</b>	<b>0.838</b>	<b>0.833</b>	9.997	<b>0.371</b>
		( 0.363 )	( 0.013 )	( 0.013 )	( 0.330 )	( 0.009 )
0.005	Full Data	—	0.750	0.750	—	39.821
		—	( 0.000 )	( 0.000 )	—	( 0.169 )
	$BM_1$	6.812	0.791	0.786	<b>0.054</b>	0.431
		( 0.478 )	( 0.008 )	( 0.008 )	( 0.005 )	( 0.010 )
	$BM_2$	8.652	0.788	0.782	0.058	0.505
	( 0.740 )	( 0.010 )	( 0.010 )	( 0.005 )	( 0.008 )	
	Proposed	<b>5.480</b>	<b>0.838</b>	<b>0.833</b>	9.997	<b>0.371</b>
		( 0.363 )	( 0.013 )	( 0.013 )	( 0.330 )	( 0.009 )

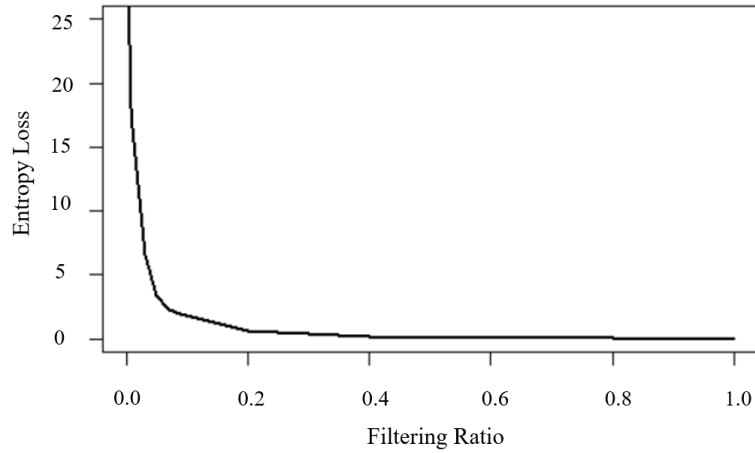


Figure 1: The increase of entropy loss as the filtering ratio decreases.

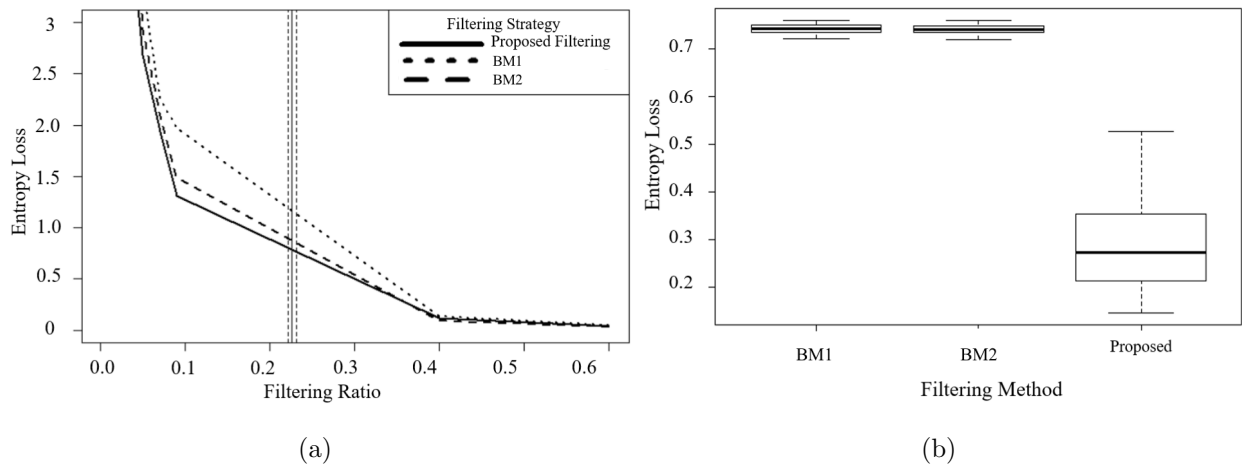


Figure 2: (a) The mean and standard errors (in vertical solid and dotted lines) of the optimal filtering ratio over 50 replications using the proposed filtering method versus the mean entropy loss at different filtering ratios for all methods over 50 replications. (b) The comparison in a boxplot of entropy loss for all methods using the fixed filtering ratio over 50 replications.

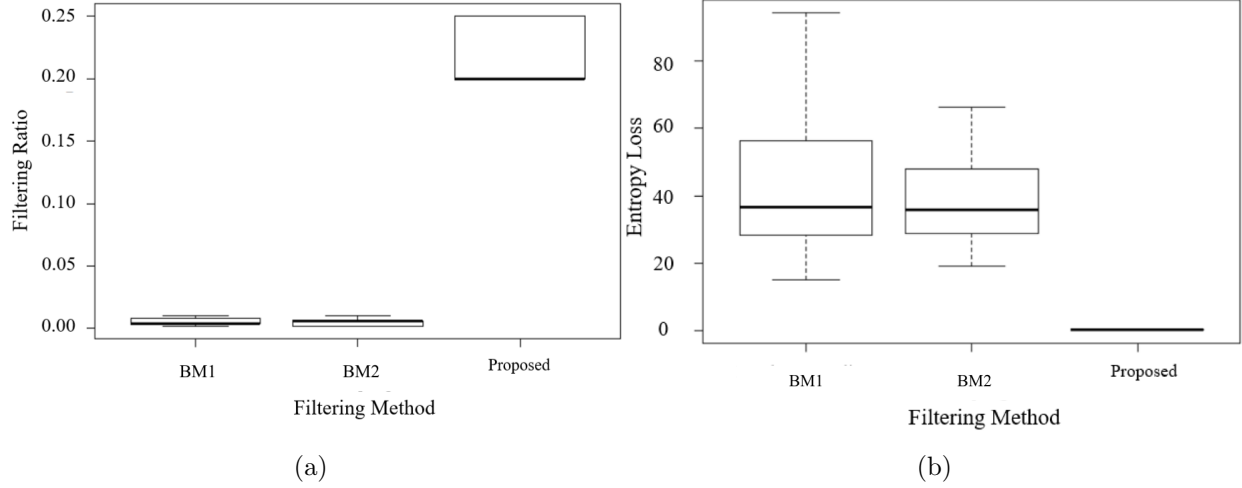


Figure 3: (a) The boxplots of the optimal filtering ratios determined over 50 replications. (b) The entropy losses obtained under the optimal filtering ratio over 50 replications.

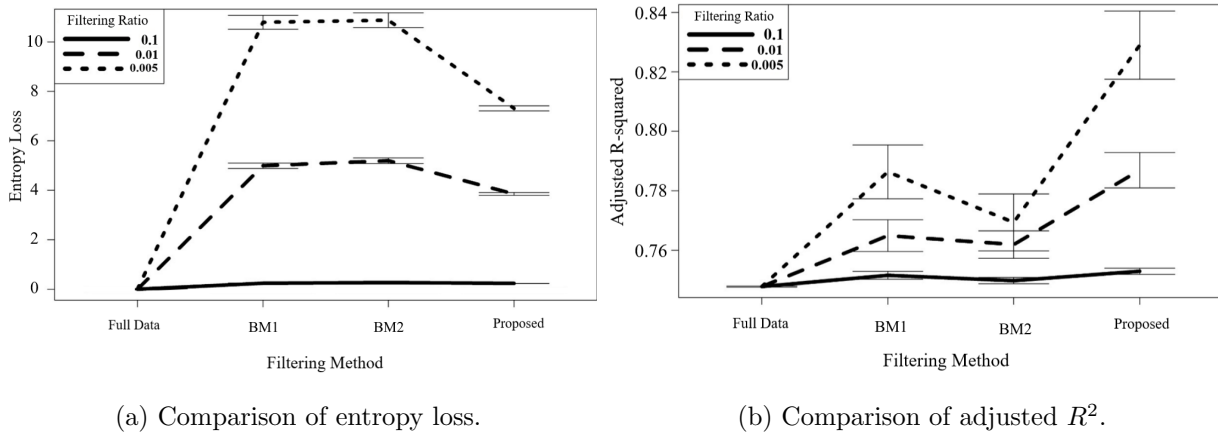


Figure 4: The means and standard errors (in black error bars) of two performance measures based on 50 simulation replications.

## 9 List of Figures

1. The increase of entropy loss as the filtering ratio decreases.
2. (a) The mean and standard errors (in vertical solid and dotted lines) of the optimal filtering ratio over 50 replications using the proposed filtering method versus the mean entropy loss at different filtering ratios for all methods over 50 replications. (b) The comparison in a boxplot of entropy loss for all methods using the fixed filtering ratio over 50 replications.
3. (a) The boxplots of the optimal filtering ratios determined over 50 replications. (b) The entropy losses obtained under the optimal filtering ratio over 50 replications.
4. The means and standard errors (in black error bars) of two performance measures based on 50 simulation replications.