



Approximate Fault-Tolerant Data Stream Aggregation for Edge Computing

Daiki Takao, Kento Sugiura, and Yoshiharu Ishikawa^(✉)

Graduate School of Informatics, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
takao@db.is.i.nagoya-u.ac.jp, {sugiura,ishikawa}@i.nagoya-u.ac.jp

Abstract. With the development of IoT, edge computing has been attracting attention in recent years. In edge computing, simple data processing, such as aggregation and filtering, can be performed at network edges to reduce the amount of data communication and distribute the processing load. In edge computing applications, it is important to guarantee low latency, high reliability, and fault tolerance. We are working on the solution of this problem in the context of environmental sensing applications. In this paper, we outline our approach. In the proposed method, the aggregate value of each device is calculated approximately and the fault tolerance is also guaranteed approximately even when the input data is missing due to sensor device failure or communication failure. In addition, the proposed method reduces the delay by outputting the processing result when the error guarantee satisfies the user's requirement.

Keywords: Stream processing · Edge computing · Approximate processing · Fault tolerance

1 Introduction

With the spread of *Internet of Things (IoT)*, collection and aggregation of information via networks is becoming increasingly important. The approach of transferring a large amount of information obtained by sensing to a server and processing it all at once by the server has a large communication overhead and greatly deteriorates the processing speed. Therefore, *edge computing*, which performs filtering and aggregation at the edge of the network as shown in Fig. 1, is attracting much attention to solve this problem.

Distributed stream processing that introduces processing at edges has advantages in terms of efficiency, but it also has difficulties. In particular, *fault-tolerance* is an important issue, especially how to process when a network failure occurs. One solution is to multiplex resources, but in edge environments, unlike cluster environments where resources are plentiful, sensing data management and stream processing must be performed by a small number of poorly equipped machines, and fault-tolerance based on the premise of resource saving is required.

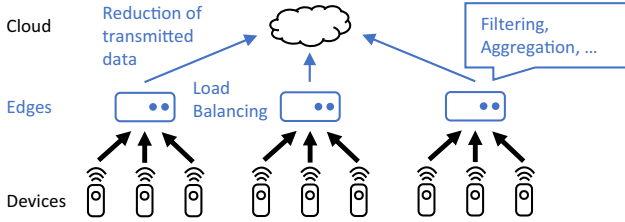


Fig. 1. Overview of edge computing

In this study, we focus on *approximate stream processing*. The basic idea is to reduce the resource burden of stream processing by allowing a small amount of error, instead of the considerable overhead incurred when trying to do it rigorously. Since the information obtained from sensors and other devices contains errors by nature, a certain amount of error is considered acceptable in many cases. Approximate stream processing has been the subject of much theoretical and systematic research, and approaches to processing with error guarantees have been proposed [5, 9]. In our research, we aim to develop these ideas and establish a method that is superior in terms of quality and efficiency.

In this paper, we outline our quality-assured approximate stream processing approach for environmental sensing applications such as temperature. In environmental sensing, the acquired values of each sensor are often correlated, and efficiency improvement using correlation between sensors has been proposed in the context of sensor networks [4]. We extend that approach by approximating the aggregate value of each sensor to provide a theoretical error guarantee for approximate processing results, which can be applied to fault-tolerance and efficient processing.

The conceptual of the proposed method is shown in Fig. 2. The system receives the required confidence and tolerance values from the user and estimates the aggregate values within the range that satisfies the user’s requirements, and reduce the processing delay. As an example, we consider a sensor stream with missing data as shown in Fig. 3. Note that in this example, the data from device X_2 has not reached the edge since time 4. Therefore, time 3 is the latest point in time (*watermark*) for the entire data source, and data processing cannot proceed after time 4. Therefore, approximate aggregate values are calculated by estimating the data of device X_2 after time 4 based on the statistical model that models the correlation and the measured values of other devices. In addition, the loss of internal state due to node failure can be regarded as data loss of all devices, such as the situation at time 5 and 6. Therefore, by estimating the missing values using a statistical model, we can approximate and guarantee the fault-tolerance of stream processing.

2 Preliminaries

In this section, we define the terms and concepts needed for the rest of this paper.

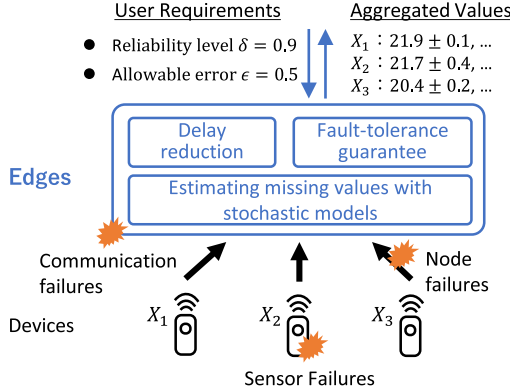


Fig. 2. Overview of the proposed method

device	time step						
	1	2	3	4	5	6	7 ...
X_1	21	22	22	22	-	-	23
X_2	21	21	22	-	-	-	...
X_3	-	20	20	20	-	-	21

Fig. 3. Data stream with missing items

2.1 Definition of Data Streams

First, we define a *sensor data stream*.

Definition 1. Let $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ be the set of all n devices. All devices are assumed to output data periodically and synchronously, and the true values of devices at each time $t \in \mathbb{N}^+$ is denoted by $\mathbf{x}^t = \langle x_1^t, x_2^t, \dots, x_n^t \rangle$. The sensor data stream that started to be measured at a certain time t is represented as an infinite series of true values of the devices at each time by the following formula:

$$\mathbf{x}^{[t, \infty)} = \langle \mathbf{x}^t, \mathbf{x}^{t+1}, \dots \rangle \tag{1}$$

However, in an edge computing environment, data from all devices may not be available at each time. Therefore, we define a *data stream with missing items* as a data stream that is actually observed.

Definition 2. Let the observed values at time t be $\mathbf{o}^t = \langle o_1^t, o_2^t, \dots, o_n^t \rangle$. For a data stream $\mathbf{x}^{[t, \infty)}$, we define a data stream with missing items by the following formula:

$$\mathbf{o}^{[t, \infty)} = \langle \mathbf{o}^t, \mathbf{o}^{t+1}, \dots \rangle, \tag{2}$$

where it is assumed that the observed values are always equal to the true values.

In this paper, we assume device failures, communication failures, and node failures as missing scenarios.

Figure 3 shows a data stream with missing items. For example, at time 1, the value of device X_3 is missing. On the other hand, since the values of X_1 and X_2 are observed, $o_1^1 = x_1^1 = 21$ and $o_2^1 = x_2^1 = 21$ hold.

Next, we define partitioning of a data stream based on *sliding windows*, assuming an aggregation process with a certain time window length.

Definition 3. *Given a data stream $\mathbf{x}^{(t,\infty)}$, a window width w , and a sliding width l , using the sliding window concept, the data stream is partitioned based on the following formula:*

$$\left\{ \mathbf{x}^{[t',t'+w)} \mid \forall i \in \mathbb{N}^0, t' = t + i \cdot l \right\}. \tag{3}$$

For example, if we partition the data stream in Fig. 3 with the window width $w = 3$ and sliding width $l = 2$, we obtain the following set:

$$\left\{ \langle \mathbf{o}^1, \mathbf{o}^2, \mathbf{o}^3 \rangle, \langle \mathbf{o}^3, \mathbf{o}^4, \mathbf{o}^5 \rangle, \dots \right\}. \tag{4}$$

In this paper, we focus on the processing of a observed finite stream $\mathbf{o}^{[t,t+w)}$ partitioned into time windows.

2.2 Estimation of Missing Values Based on Multidimensional Gaussian Distribution

In this research, the correlation between devices \mathbf{X} is represented using a *multi-dimensional (multivariate) Gaussian distribution* $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Note that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent the mean vector and variance-covariance matrix of \mathbf{X} , respectively. In the following, the missing value estimation by range integration and the improvement of the estimation accuracy by using the posterior probability distribution (correlation between devices) are explained.

When the value of the device $X \in \mathbf{X}$ follows a Gaussian distribution with expected value μ and variance σ , the true value can be estimated probabilistically by range integration even if the value of X is missing. For example, the probability that the value of X is x' can be obtained using a certain error upper bound e as follows:

$$P(X \in [x' - e, x' + e]) = \int_{x'-e}^{x'+e} \mathcal{N}(x \mid \mu_X, \sigma_X) dx. \tag{5}$$

When using the same error upper bound, since the probability is largest when $x' = \mu$, we can simply use the expected value as an estimate to recover the missing value probabilistically.

There is a correlation between devices in the real world, for example, the measured values of temperature sensors located geographically close to each other have similar values. In [4], this correlation is modeled by a multidimensional Gaussian distribution, and a method to improve the estimation accuracy by using the posterior probability distribution has been proposed. When the observed

value \mathbf{o} is obtained for some device $\mathbf{O} \subseteq \mathbf{X}$, the Gaussian distribution $\mathcal{N}(x | \mu_{X|\mathbf{o}}, \sigma_{X\mathbf{o}}, \sigma_{X|\mathbf{o}})$ can be obtained from the following equation [3]:

$$\mu_{X|\mathbf{o}} = \mu_X + \Sigma_{X\mathbf{O}} \Sigma_{\mathbf{O}\mathbf{O}}^{-1} (\mathbf{o} - \mu_{\mathbf{O}}) \quad (6)$$

$$\sigma_{X|\mathbf{o}} = \Sigma_{XX} - \Sigma_{X\mathbf{O}} \Sigma_{\mathbf{O}\mathbf{O}}^{-1} \Sigma_{\mathbf{O}X}, \quad (7)$$

where the subscript of each symbol indicates that the dimension corresponding to the subscript has been extracted from the vector or matrix. For example, μ_X is the value of the dimension corresponding to X extracted from the mean vector $\boldsymbol{\mu}$ (i.e., the expected value of device X), and $\Sigma_{X\mathbf{O}}$ is the columns corresponding to \mathbf{O} extracted from the X rows of the variance-covariance matrix Σ (i.e., the covariance vector of device X and the set of devices \mathbf{O} from which the data was obtained). Since the variance of the posterior probability distribution is reduced compared to that of the prior distribution, more accurate estimation is possible.

3 Estimation of Time Window Aggregation Based on Probabilistic Models

In [4], Deshpande et al. proposed an approximate aggregate query method using a multidimensional Gaussian distribution. The method targets the aggregation among all devices \mathbf{X} , i.e., the processing for a specific time. However, stream processing in this paper generally involves aggregation over time windows, which requires processing over multiple time periods. We extend the method of Deshpande et al. to aggregation over time windows and derive the estimated confidence of the aggregation process for data streams with missing items [12]. Since the proposed method uses the reproducibility of the Gaussian distribution, it covers both mean and total aggregates. However, due to space limitations, only the mean aggregate will be presented below.

3.1 Derivation of Gaussian Distribution for Aggregate Values

Since the Gaussian distribution is reproducible, the mean value of the device $X \in \mathbf{X}$ in a given time window $\mathbf{x}^{[t, t+w)}$, $Y_X = (\sum_{t' \in [t, t+w)} X^{t'})/w$, also follows a Gaussian distribution. By replacing the expression for the expected value and variance of the Gaussian distribution for the aggregate value at a particular time in [4] with Y_X , we can derive the expected value and variance of Y_X as follows [12]:

$$E[Y_X] = \frac{1}{w} \sum_{t' \in [t, t+w)} E[X^{t'}] \quad (8)$$

$$E[(Y_X - \mu_{Y_X})^2] \approx \frac{1}{w^2} \sum_{t' \in [t, t+w)} E[(X^{t'} - \mu_{X^{t'}})^2]. \quad (9)$$

In the approximate derivation of Eq. (9), it is assumed that the same probability model $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is used for all times in the time window.

As shown in Eq. (8) and (9), the average aggregate for the time window $\mathbf{x}^{[t,t+w]}$ is obtained from the expectation and variance of the device at each time. Therefore, from the partial observations \mathbf{o}^t obtained at each time, the expectation and variance of the missing values are calculated based on Eq. (6) and (7), and the mean value for the entire time window is estimated. If there is an observed value o^t at time t for the device $X \in \mathbf{X}$, we calculate the expected value as $\mu_X^t = o^t$ and the variance as $\sigma_X^t = 0$.

3.2 Confidence of Estimation in Time Window Aggregation

Since the results of the time window aggregation also follow a Gaussian distribution as described in the previous section, we can calculate the confidence of the estimation in a similar way to Sect. 2.2. The estimated confidence of the mean value Y_X of the device $X \in \mathbf{X}$ in a given time window is given by the following equation:

$$P(Y_X \in [\mu_{Y_X} - e, \mu_{Y_X} + e]) = \int_{\mu_{Y_X} - e}^{\mu_{Y_X} + e} \mathcal{N}(y \mid \mu_{Y_X}, \sigma_{Y_X}) dy, \quad (10)$$

where the expected value μ_{Y_X} is treated as an estimate, and the upper bound on the error is denoted by e .

Note here that the results of the aggregate query are probabilistic because they contain missing values. Therefore, in order to calculate the estimated confidence level, the user needs to specify the parameters for either the error upper bound or the required confidence level. For example, suppose that the user specifies the required confidence level δ as a parameter. In this case, by solving $P(Y_X \in [\mu_{Y_X} - e, \mu_{Y_X} + e]) = \delta$, we can calculate the minimum estimation error e that satisfies the required confidence level. Note that if the number of missing values is extremely large, the estimation error e also becomes large, and the range of possible values of the aggregate $\mu_{Y_X} \pm e$ also becomes wide.

4 Delay Reduction Based on Probabilistic Estimation

The approach of approximate data stream processing can be applied to efficient processing as well as fault handling. In this section, we discuss delay reduction in time window aggregation based on the output of estimates. The idea here is to reduce the delay in the time window aggregation by outputting the results when the user-specified estimation accuracy is met.

In this study, we use the *required confidence level* δ and *acceptable error* ϵ as thresholds to determine the estimation accuracy. In other words, we require that the aggregate value of each device in a given time window satisfies the following formula:

$$\forall X \in \mathbf{X}, P(Y_X \in [\mu_{Y_X} - \epsilon, \mu_{Y_X} + \epsilon]) \geq \delta. \quad (11)$$

Note, however, that this equation may not be satisfied depending on the degree of deficiency, as described in Sect. 3.2.

If we express the required confidence level of the estimation by Eq. (11), it can be seen that there are cases where not all observations in the entire time window are required for the output of the estimate. To take an extreme example, if the required confidence level is extremely small, such as $\delta = 0.01$, it is possible that Eq. (11) can be satisfied by the prior distribution alone without using any observations. Therefore, in this paper, the estimated aggregate value is output when it is determined that Eq. (11) is satisfied, even if it is in the middle of processing a time window.

In order to efficiently determine whether or not to output an estimate, we first describe the variance thresholds necessary for sufficiently accurate estimation. We then describe the computation of the estimation accuracy at the midpoint of the time window, and propose an output decision for the estimate based on stream processing with incremental Gaussian updates.

4.1 Threshold Value for Output Judgment of Estimated Value

In order to efficiently determine the validity of Eq. (11), the decision of the size of the confidence level is attributed to the decision of the size of the variance. Equation (11) determines whether each confidence level satisfies the required confidence level δ , but the calculation of the confidence level requires a range integral over a Gaussian distribution. Namely, it is inefficient to simply calculate the range integral for each device at each time.

Here, we note that the integration result does not depend on the expected value when using an integration range centered on the expected value in the Gaussian distribution [4]. In this case, Eq. (11) can be rewritten as follows:

$$\forall X \in \mathbf{X}, P(Y_X \in [-\epsilon, \epsilon]) \geq \delta \leftrightarrow \forall X \in \mathbf{X}, \int_{-\epsilon}^{\epsilon} \mathcal{N}(y \mid 0, \sigma_{Y_X}) dy \geq \delta. \tag{12}$$

This allows us to solve the following equation for the given required confidence level δ and error upper bound ϵ to find the *variance threshold* σ_θ required to meet the required estimation accuracy:

$$\int_{-\epsilon}^{\epsilon} \mathcal{N}(y \mid 0, \sigma_{Y_X}) dy = \delta. \tag{13}$$

Since it is a range integral centered on the expected value of the Gaussian distribution, the smaller the value of the variance, the larger the value of the confidence level. In other words, the following equation and Eq. (11) are equivalent:

$$\forall X \in \mathbf{X}, \sigma_{Y_X} \leq \sigma_\theta. \tag{14}$$

As described below, the variance of the aggregate value can be calculated incrementally by basic arithmetic operations, and thus the timing of the output can be determined more efficiently than a simple decision using range integration.

4.2 Incremental Update of Probabilistic Estimation

Using the variance thresholds described in the previous section, we can calculate the estimates and determine their output by determining the expected value and variance of the Gaussian distribution that the aggregate follows at each time in the time window. That is, for each sensor $X \in \mathbf{X}$, when the variance of the aggregate σ_{Y_X} is less than or equal to the threshold σ_θ , the expected value μ_{Y_X} is output as the estimated value. If we simply calculate the expected value and variance, duplicate calculations will be performed at each time. Thus, in the following, we describe an incremental update method suitable for stream processing.

First, we show the expected value and variance at any time $t^* \in [t, t+w)$ in a time window $\mathbf{x}^{[t, t+w)}$. Since $\mathbf{o}^{[t, t^*)}$ is obtained as the observed value, the value of the device at each time can be estimated using the posterior probability distribution up to time t^* and the prior probability distribution at times after t^* . That is, from Eq. (8) and (9), the expected value of the aggregate $\mu_{Y_X}^{t^*}$ and variance $\sigma_{Y_X}^{t^*}$ at time t^* can be obtained by the following equation:

$$\mu_{Y_X}^{t^*} = \frac{1}{w} \left(\sum_{t' \in [t, t^*)} \mu_{X t' | \mathbf{o}^{t'}} + \sum_{t' \in (t^*, t+w)} \mu_X \right) \quad (15)$$

$$\sigma_{Y_X}^{t^*} = \frac{1}{w^2} \left(\sum_{t' \in [t, t^*)} \sigma_{X t' | \mathbf{o}^{t'}} + \sum_{t' \in (t^*, t+w)} \sigma_X \right). \quad (16)$$

Note that the prior distribution in the time window is constant.

We derive an asymptotic equation to incrementally update the expected value and variance. The following equations can be derived by transforming $\mu_{Y_X}^{t^*} - \mu_{Y_X}^{t^*-1}$ for the expectation value and $\sigma_{Y_X}^{t^*} - \sigma_{Y_X}^{t^*-1}$ for the variance:

$$\mu_{Y_X}^{t^*} - \mu_{Y_X}^{t^*-1} = \frac{\mu_X - \mu_{X t^* | \mathbf{o}^{t^*}}}{w} \quad (17)$$

$$\sigma_{Y_X}^{t^*} - \sigma_{Y_X}^{t^*-1} = \frac{\sigma_X - \sigma_{X t^* | \mathbf{o}^{t^*}}}{w^2}. \quad (18)$$

In other words, the Gaussian distribution of aggregate values can be updated by calculating the posterior probability distribution based on the new observed values \mathbf{o}^{t^*} obtained at each time t^* and applying the difference.

5 Application to Fault Tolerance Assurance

In this section, we consider the estimation of aggregate values based on the probabilistic model described so far as a guarantee of fault tolerance in edge computing environments. Unlike existing distributed parallel stream processing systems that take fault tolerance into account, in an edge computing environment, there may be a situation in which communication with each device is disrupted due

to a fault, making it impossible to retrieve and resend data observed during the fault. Namely, there may be cases where all the input data for a certain period of time is missing, such as time 5 and 6 in Fig. 3.

Even if all data for a certain period of time is missing due to a failure, the aggregate value can be estimated in the same way using our method. Although missing data due to a failure has a large impact, it is not theoretically different from partial missing data due to communication failure or specific missing data due to device failure. In other words, by estimating all data for all missing periods based on prior probabilities, we can calculate the estimated aggregate value and the confidence level reduced by the missing data.

Specifically, we use *checkpointing* together with the incremental probabilistic estimation described in Sect. 4. Checkpointing is a common method used in existing distributed parallel stream processing systems. The internal state of the process, i.e., the expected value and variance of the aggregate result, is periodically stored in a persistent storage area as a checkpoint. Then, when a failure occurs, the expected value $\mu_Y^{t_c}$ and variance $\sigma_Y^{t_c}$ of the latest checkpoint acquisition time t_c are restored to main memory and processing is restarted. However, as mentioned earlier, our method does not allow retransmission of lost data from the data source after recovery. All input data $\mathbf{o}^{(t_c, t_d)}$ up to the recovery time t_d are assumed to have been lost and the process is restarted. If no failure occurs after that, the expected value μ_{Y_X} and variance σ_{Y_X} of the Gaussian distribution followed by the aggregate Y_X for sensor $X \in \mathbf{X}$ can be obtained as follows:

$$\mu_{Y_X} = \frac{1}{w} \left(\sum_{t' \in [t, t_c]} \mu_{X^{t'} | \mathbf{o}^{t'}} + \sum_{t' \in (t_c, t_d)} \mu_X + \sum_{t' \in [t_d, t+w)} \mu_{X^{t'} | \mathbf{o}^{t'}} \right) \tag{19}$$

$$\sigma_{Y_X} = \frac{1}{w^2} \left(\sum_{t' \in [t, t_c]} \sigma_{X^{t'} | \mathbf{o}^{t'}} + \sum_{t' \in (t_c, t_d)} \sigma_X + \sum_{t' \in [t_d, t+w)} \sigma_{X^{t'} | \mathbf{o}^{t'}} \right). \tag{20}$$

In the proposed method, there is no need to perform any special processing other than restoring the checkpoint as a failure recovery process. As shown in Eq. (15) and the second term in parentheses of (16), for any time $t^* \in [t, t+w)$ in a time window, the prior distribution is used to calculate the expectation and variance for times after t^* . Thus, at the time of the latest checkpoint acquisition t_c , we can say that the computation of the expected value and variance estimates for all missing data intervals (t_c, t_d) until the completion of the recovery has been completed. Therefore, after the failure is recovered, the approximate aggregation process with error guarantee can be resumed by simply restoring the latest checkpoint.

6 Related Work

We discuss the problem of missing data in time series data and related research on fault-tolerance guarantees for stream processing.

6.1 Missing Data in Time Series

The problem of missing data in time series has long attracted attention, and many studies have been reported. The basic approaches to the missing value problem are listwise and pairwise methods, which remove tuples and cases containing missing values [10]. Although these methods are widely used due to their simplicity, they have a problem of processing accuracy because the number of data to be processed is small and the number of data per attribute can be biased. Then, many methods have been proposed to estimate missing values with high accuracy, including interpolation by regression [6, 11, 13] and association rule mining [8].

For example, *stochastic regression imputation* takes into account the distribution of the data and randomly scatters the values from the regression line as the estimated values, and unlike conventional regression methods, it can provide more accurate estimates without underestimating the variance [6]. While this method targets the accurate estimation of each missing value, the proposed method aims to efficiently obtain an approximate aggregate result of the data stream containing the missing values. The proposed method takes into account the variance of the estimates through an error evaluation framework, and also allows for early output based on error guarantees and fault tolerance guarantees using checkpointing.

In addition, existing methods have various innovations to improve the processing accuracy, such as *multiple imputation* that combines multiple estimation results to solve the problems of underestimation of variance and estimation uncertainty [6]. However, there are some issues in low latency and fault tolerance, which are important in edge computing environments.

6.2 Fault Tolerance Assurance for Stream Processing Systems

For the problem of data loss due to system failures, many stream processing systems, including Flink [1] and Spark Streaming [2], guarantee robust fault tolerance without errors. These systems, which assume a parallel processing environment in the cloud, can guarantee the output of the result of processing the input data once without excess or deficiency, regardless of the occurrence of a failure, by backing up the internal state by checkpointing and restoring the input data based on retransmission and reprocessing. However, especially in a single-node environment, a failure at the network edge may cause a breakdown in communication with each device, making it impossible to acquire and resend data observed during the failure. The redundant configuration of the system is essential for retransmission of missing data, but the redundant configuration for each edge machine is expensive and impractical.

In order to reduce the cost of fault tolerance guarantee, Huang et al. [7] proposed an approximate fault tolerance method that guarantees the reliability of the processing results. This method reduces the amount of data to be backed up and the frequency of backups by backing up only when the threshold values for the number of unprocessed data and the amount of change in the internal state

are exceeded, thereby improving the throughput. However, this method requires a detailed preliminary analysis that takes into account the actual node configuration in order to set a threshold value suitable for a Service Level Agreement (SLA), which poses many difficulties in actual operation.

In addition, all of these methods target only missing data due to stream processing system failures, and cannot handle missing input data due to sensor device failures or communication failures.

7 Conclusions

In this paper, we outlined our research on approximate aggregate processing methods that improve processing low latency, high reliability, and fault tolerance for edge computing environmental sensing applications. The proposed method uses a probabilistic model to estimate missing data due to sensor device failures or communication failures, and theoretically guarantees an upper bound on the error for aggregated results. Furthermore, we reduce the delay by outputting an approximate aggregate value that is incrementally computed at the shortest time when the error guarantee of our method satisfies the user requirement. In addition, our method can efficiently recover from failures even in situations where acquisition and retransmission of input data are impossible due to failures.

One of the future challenges is to follow the changes in correlation over time by updating the model dynamically. Dynamic model updating is expected to improve the accuracy of the process, since it will be able to take into account changes in correlations by time of day and season. However, in order to enable dynamic model updating, there are many issues that need to be considered in terms of consistency of processing results and fault tolerance, such as the adequacy of guaranteeing errors in aggregate results in response to model changes, and considering the effects of model update information loss due to node failures. For example, incremental processing and internal state management will be necessary to ensure that the theoretical error guarantee does not break down in response to changes in the model, including cases where the models are different before and after failure recovery. Adaptive checkpointing can also be considered, taking into account the trade-off between the degradation of processing performance due to the loss of model update information and the checkpointing cost, but it should also be balanced among multiple time windows with partial overlap, such as a sliding window.

Acknowledgement. This study is based on results obtained from a project, JPNP16007, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). In addition, this work was supported partly by KAKENHI (16H01722, 20K19804, and 21H03555).

References

1. Apache Flink: Stateful Computations over Data Streams. <https://flink.apache.org/>
2. Spark Streaming | Apache Spark. <http://spark.apache.org/streaming/>
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Boston (2006). <https://doi.org/10.1007/978-1-4615-7566-5>
4. Deshpande, A., Guestrin, C., Madden, S.R., Hellerstein, J.M., Hong, W.: Model-based approximate querying in sensor networks. *VLDB J.* **14**(4), 417–443 (2005)
5. Duffield, N., Xu, Y., Xia, L., Ahmed, N.K., Yu, M.: Stream aggregation through order sampling. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM). pp. 909–918 (2017)
6. Enders, C.K.: Applied Missing Data Analysis. Guilford Press, New York (2010)
7. Huang, Q., Lee, P.P.C.: Toward high-performance distributed stream processing via approximate fault tolerance. *Proc. VLDB (PVLDB)* **10**(3), 73–84 (2016)
8. Jiang, N., Gruenwald, L.: Estimating missing data in data streams. In: Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA). pp. 981–987 (2007)
9. Johnson, T., Muthukrishnan, S., Rozenbaum, I.: Sampling algorithms in a stream operator. In: Proceedings of ACM SIGMOD. pp. 1–12 (2005)
10. Raymond, M.R., Roberts, D.M.: A comparison of methods for treating incomplete data in selection research. *Educ. Psychol. Measur.* **47**(1), 13–26 (1987)
11. Ren, X., Sug, H., Lee, H.: A new estimation model for wireless sensor networks based on the spatial-temporal correlation analysis. *J. Inf. Commun. Converg. Eng.* **13**(2), 105–112 (2015)
12. Takao, D., Sugiura, K., Ishikawa, Y.: Approximate streaming aggregation with low-latency and high-reliability for edge computing. *IEICE Trans. Inf. Syst.* **J104-D**(5), 463–475 (2021). (in Japanese)
13. Yi, X., Zheng, Y., Zhang, J., Li, T.: ST-MVL: Filling missing values in geo-sensory time series data. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), pp. 2704–2710 (2016)