An Agent Based Parallel and Secure Framework to Collect Feedbacks

Rahma Bintey Mufiz Mukta, Mohammad Shamsul Arefin*
Department of Computer Science and Engineering, Chittagong University of Engineering & Technology, Chittagong-4349, Bangladesh.

* Corresponding author. Tel.: +8801716890204; email: sarefin@cuet.ac.bd
Manuscript submitted March 10, 2019; accepted May 20, 2019.
doi: 10.17706/jcp.14.6.404-425

Abstract: Information technology considers a feedback system as a computer program that collects information from the users and guides the target audiences in order to achieve the required outcomes. The feedback systems can be used as a part of an intervention in organizations to increase awareness and improve performance. But unwillingness of employees to provide feedbacks about their organizations for many reasons such as possibility of losing jobs and facing social harassments, insist them to keep some secrets while giving feedbacks. This is the most significant problem of collecting feedbacks from the employees of the organizations. Collecting feedbacks from the employees without disclosing their identities can be a possible solution of this problem. Considering this fact, in this paper we introduce a method to collect feedbacks in a secure environment. In our approach, we use an agent based parallel computation technique that can collect feedbacks from the users without disclosing their given information. In our system, we consider both comments-based feedbacks and numeric feedbacks. In addition, we have considered two different languages for providing feedbacks. For secure collection of feedbacks, we use Java agent Development Framework (JADE). We perform several experiments on both real data and synthetic data to show the effectiveness of our approach. Experimental results show that our system can be used to collect comment-based feedbacks and numerical feedbacks in two languages efficiently without disclosing the feedbacks of individuals' to others.

Key words: Users' feedback, agent based system, secured parallel computation, JADE.

1. Introduction

Feedback is about giving information in a way that encourages the recipient to accept it, reflect on it, learn from it and hopefully make changes for the better. 'Customer feedback' is the buyers' reaction to a firm's products and policies, 'operational feedback' is the internally generated information on a firm's performance and 'employee feedback' is the reaction from employees about the working conditions to the managerial body. Specifically feedback can clarify good performance and help to develop self-assessment.

Many well known e-commerce companies such as Amazon [1], eBay [2], Olx [3], Agoda [4], TripAdvisor [5], Expedia [6], Rakuten [7], Elance [8] sell their products and services via Internet. Besides the e-commerce companies, agencies, government and non-government organizations also provide different types of services via Internet. Most of these companies, agencies and organizations collect feedbacks from the customers to improve the quality of their services. Nowadays feedback system is also common through giving rating for restaurants and visiting places. This also ensures a competitive situation for betterment

among the organizers. In these scenarios, user feedbacks navigate another user for selecting the best one. But in some cases like for collecting actual feedback about the working conditions of any garment industry, we have to ensure the privacy of the workers who are giving feedbacks. In many cases, clothing products of many renowned brands like Nike, H&M, Gap etc are produced in sweatshops in developing countries like Bangladesh. Almost half of the population in Bangladesh lives off of less than a dollar a day. Garment workers in Bangladesh toil day after day under extremely harsh conditions for low wages, sometimes handling dangerous chemicals with their bare hands and inhaling toxic fumes due to poor ventilation in many factories. Moreover, Tazreen fashion factory burnt [9] was the worst garment factory accident in the history of Bangladesh and Rana Plaza collapse [10] in Bangladesh was the worst garment factory accident in the history of world. So now if Bangladesh Government's garment monitoring authority wants to know whether any industry officials are forcing their workers to work in structurally defective buildings and whether there are any cases of physical assault, intimidation and threats, dismissal of union leaders, and false criminal complaints by factory officials or their associates against garment workers, then there is no other best way accept collecting feedback from the workers of this respective factory. Main problem in collecting such feedbacks by these companies, agencies or organizations is that customers/users need to disclose their identities while providing feedbacks. But some problematic situations create reluctance among the users to disclose their identities. Actually workers are forced to continue their job in a poor condition. The same situation is true for almost all companies, organizations and agencies of the developing countries. Here comes the need to develop a framework where customers/users can provide their feedbacks in such a way that it is not necessary to disclose their identities while providing the feedbacks. The main focus of this paper is to develop such a framework. This system will collect feedback from its users in a secured manner. We will also show the efficiency of the system through several experimental analyses.

2. Motivation

Readymade garments sector is the most contributing sector in economic growth of Bangladesh. Continuing the economic success of the Bangladesh garment sector offers benefits for everyone – the retail companies and their consumers, factory owners, and the government. But those gains should not come at the cost of lives and the suffering of garment workers struggling for a better future. In April of 2013, an eight story building in Bangladesh called Rana Plaza collapsed leaving over 100 dead and over 2,000 injured [10]. The poor conditions of the factory itself and the lack of safety precautions taken to ensure its workers' well-being were neglected and therefore led to the collapse. In addition to this incident, there has been a history of factory mishaps over the past couple of years in Bangladesh. In November of 2012, the Tazreen garment factory in Bangladesh caught fire and killed 112 of its workers. Following the disaster, retailers and Bangladesh's government promised widespread reforms. Thousands of factories have since been checked for structural problems with dozens closed and others refurbished, and many more helped to improve working conditions and treatment of employees in initiatives partly paid for by western retailers. However, the report [9] suggests problems remain. Researchers interviewed more than 160 workers from 44 factories in and around Dhaka, including many that supply garments to high streets in North America, Europe and Australia. They heard complaints of physical assault, verbal abuse, forced overtime, unsanitary conditions, denial of paid maternity leave, and failure to pay wages and bonuses on time or in full. The employees of such type of industries are bound to continue their jobs in such horrible situations due to their poverty. They cannot even complain about the poor conditions of the garments to the government's garments monitoring authority due to the fear of leaking their identities that may cause losing their jobs and other social harassments. The situations are almost similar in many industries in the developing

countries.

If we can collect feedbacks from the employees of any industry, the actual scenarios of that industry can be identified. From collected feedbacks of the industry, the monitoring authority can find out the shortcomings of the industry. This will help the monitoring authority to take necessary steps against the poorly performing industries. In addition, it will create a competitive environment among the industries. So a feedback system is very much essential. However, an employee of an industry, in general, does not want to disclose her/his identity while providing feedbacks about the industry. There are many situations where discloser of users' identities can create problems for the users. As for example, if an employee of an organization provides negative feedbacks about the organization, the authority may create financial and social problems for that employee. This is the most significant problem for developing such a feedback system. We can easily overcome this problem by developing a feedback system that will preserve individual's privacy while collecting feedbacks. Therefore, in this proposed method, an agent-based parallel computation framework is proposed to obtain feedbacks from the users. This solves the privacy problems of feedback systems. The proposed system can also ensure efficient performance and accurate results.

3. Related Work

3.1. Privacy Preserving Systems

Data privacy issues can arise in response to information from a wide range of sources. The challenge of data privacy is to utilize data while protecting individual's privacy preferences and their personally identifiable information. Several anonymization techniques have been developed for this purpose. One popular anonymization approach is k-anonymity [11]. With k-anonymity an original data set containing personal health information can be transformed so that it is difficult for an intruder to determine the identity of the individuals in that data set. A *k*-anonymized data set has the property that each record is similar to at least another *k*-1 other records on the potentially identifying variables. Though *k*-anonymity is the widely used privacy preserving technique, but it cannot gives total guarantee in privacy. Homogeneity Attack and Background Knowledge attack show vulnerability in this method. Here comes the necessity of another privacy preserving technique to create diversity in anonymous data. To overcome the limitations of *k*-anonymity, Machanavajjhala *et al*. [12] proposed ℓ-diversity as a stronger notion of privacy. A table is said to have ℓ-diversity if every equivalence class of the table has ℓ-diversity. But ℓ-diversity is insufficient to prevent attribute disclosure due to skewness attack and similarity attack. So another privacy preserving technique called t-closeness has introduced by Ninghui *et al*. [13]. A table is said to have t-closeness if all equivalence classes have t-closeness. But Ninghui *et al*. [13] also noted that *t*-closeness protects against attribute disclosure, but does not deal with identity disclosure.

Recently privacy preserving techniques for data mining is of great concern among the researchers. So many techniques have been evolved for the same. Randomization approach based techniques are used to protect data privacy in the works from [14]-[18]. In work [14] probability distribution is used to construct data mining models as classifiers. Here authors used iterative algorithms to reconstruct data distribution. Authors showed in [15], [16] how to build Naïve Bayesian classifier from perturbed data. The works in [17], [18] consider users' privacy while mining association rules. The main idea of [17] is to maximize the privacy of the users and to maintain a high accuracy in the results obtained with the association rule mining. In [18], authors present a privacy preserving frame-work for mining association rules from randomized data.

3.2. Privacy Preserving Feedback Systems

Although privacy of individual's is an important issue in any computation, till now there is very little

consideration about preserving individual's privacy in feedback systems.

i. In our work [19], we have configured an agent based secured feedback system. In this paper we considered feedback from users through numeric values. Written comments from users were not considered here and the experimental analysis was done with synthetic dataset only. This paper is an extended version of our work of [19] that can generate feedbacks based on numerical values and written comments.

ii. As for the privacy issue, authors in [20] introduce a visualization technique known as *Conversation Votes* to create new backchannels in conversation and augment collocated interaction. In this paper, authors expand the idea of a social mirror to incorporate direct user feedback in the form of anonymous voting.

iii. *MyExperience* [21] is a feedback system that captures both objective and subjective in situ data on mobile computing activities. To preserve the privacy of individual's, *MyExperience* uses strong cryptographic hashing, SHA-1, to map personal information.

iv. Hashem *et al.* [22] propose a theoretical framework for privacy preserving feedbacks collection. In their approach, firstly an user randomly divides each of her/his record's values into several parts and keeps one part for her/him and sends each of the remaining parts to each of remaining users. When all the transactions are completed, the users submit the individual sum of numbers they poses to the servers. Based on the individual sums, the server then computes the average corresponds to each field of the record. However, their approach has several major limitations. First, their system is not scalable well in case of large number of users. As for example, if there are n users and each user's feedback record contains two values then their system needs a transmission of $n(n\text{-}1)$ values among the users. Second, there is no consideration about protection of data during transmission in their system. As a result, there is no way to protect data from third party access and modification. Third, their system is highly vulnerable in presence of dishonest users. If there are some dishonest users in the system, they can easily modify the data send to them. As a result, the feedback system will produce wrong output. However, their system is not robust if there are some dishonest users in the system.

In our work, we provide a framework that can overcome the problems of [22] and also the proposed system is an upgraded version of [19]. Our agent-based computation framework can significantly improve the overall accuracy and computation performance. This framework also can provide accurate feedbacks even in presence of dishonest users. In addition, our system can protect data from third party access and modification. Moreover, due to proper parallelism, the computation time of the propose algorithm is almost independent to the number of users while obtaining feedbacks.

4. Methodology

Our feedback collection framework will be a distributed framework. Users will be able to provide their feedbacks from geographically distant locations both in numerical values and in written form. Corresponding numerical score will be calculated from the written one. Then we will process the individual feedbacks collected from users into a single feedback result in aggregated form. In this whole process users' real feedback data and their identity will not be disclosed.

The system architecture of agent based privacy aware feedback collection framework consists of three main modules: preprocessing module, data integration module and aggregated feedback generation module. The overall system architecture is shown in Fig. 1.

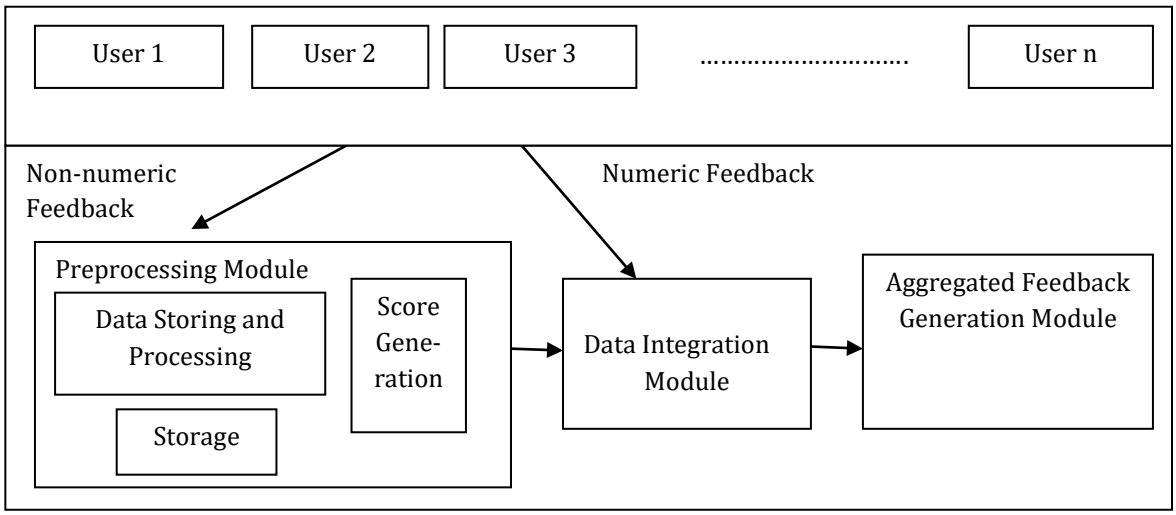

Fig. 1. System architecture for privacy aware feedback collection framework.

Preprocessing module comprises of three sub-modules: Data storing and processing, storage and score generation. This preprocessing module will take the written feedback as input and then its sub-modules will process these feedbacks to obtain the corresponding numeric values of each feedback. Data integration module will merge the processed written feedback values in numeric form of each user with his/her other feedbacks that he/she has given in real numerical values. Finally Aggregated feedback generation module will take all numeric values for each user from data integration module for further processing. This module will process the feedbacks from the system.

4.1. Preprocessing Module

Preprocessing module is necessary to convert the written feedbacks into corresponding numeric values for further processing. The system architecture of the proposed method for generating score from written feedback is depicted in Fig. 2.

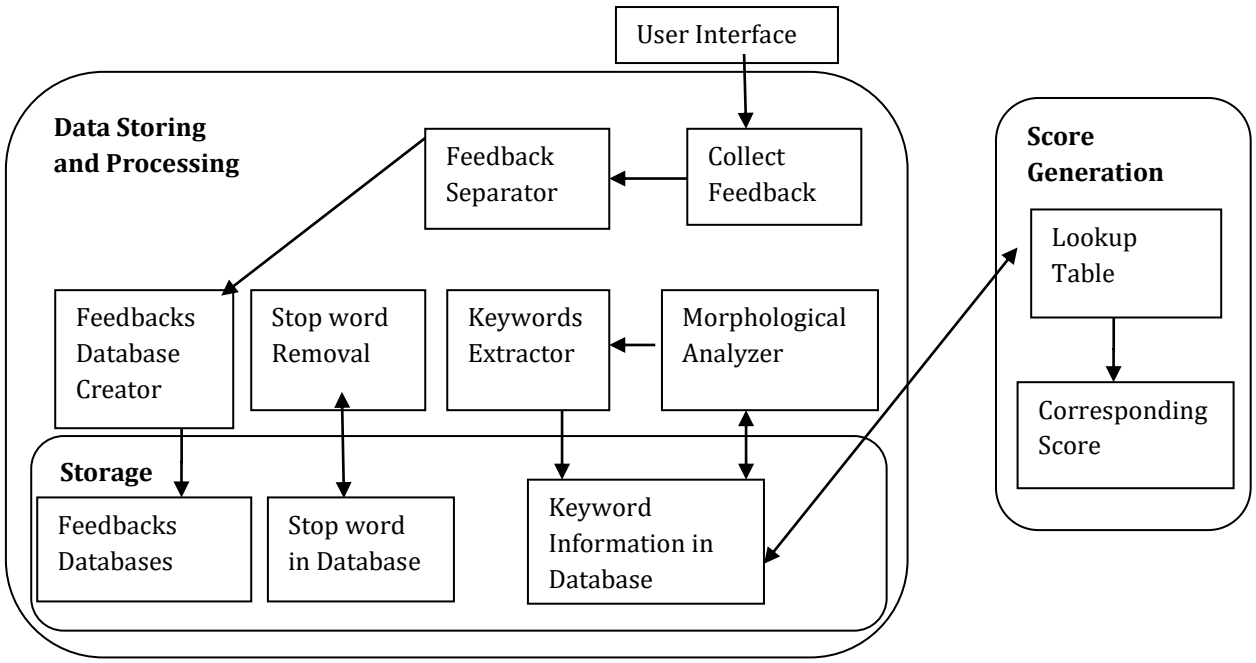

Fig. 2. Score Generation module for written feedback.

This figure has three modules: Data Storing and Processing, Data Storage and Score Generation. The single directional arrows represent the direction of next sub-module to be executed in a module and the double directional arrows represent the relationship of a sub-module with sub-modules from which the sub-module gets help for processing. The modules are described in the following sub-sections.

4.1.1. Data storing and processing

Data storing and processing module consists of the sub-modules: collect feedback, feedbacks separator, feedbacks database creator, stopword removal, keywords extractor and morphological analyzer sub-modules. The relationships among these sub-modules are represented in Fig. 1.

- **Collect Feedback**

  Collect feedback sub-module takes the set of feedbacks F = F_{b1}, F_{b2},. . . . .F_{bn}, F_{e1}, F_{e2},. . . . . ., F_{em} as input from the user. Here, F_{bi},1<*i*<*n* is the list of written feedbacks in one language and F_{ej},1< *j*<*m* is the list of written feedbacks in another language. In our proposed method, we consider that the characters in feedbacks are Unicode supported.

- **Feedbacks Separator**

  Feedbacks separator sub-module separates the feedbacks set F into two subsets F_1 = F_{b1}, F_{b2},. . . . . ., F_{bn} and F_2 = F_{e1}, F_{e2},. . . . . .,F_{em} based on the contents of written feedbacks. The separation is necessary as the analyzing of feedbacks mainly based on the language of the feedback's contents. For separating the feedbacks, we just check some initial sentences of each feedback to determine the language.

- **Feedbacks Database Creator**

  Feedbacks database creator module stores each of the feedback sets F_1 = F_{b1}, F_{b2},. . . . . ., F_{bn} and F_2 = F_{e1}, F_{e2},. . . . . ., F_{em}. In our system, we create two separate file systems to store the feedbacks of each set. One file systems stores the feedbacks in one language and another stores the feedbacks in another language. Each feedback database is created with the information feedback ID, user ID and size for each of the feedbacks. The algorithm for creating and storing the feedbacks in database is given in **Algorithm 1**.

Algorithm 1: *Feedback_DB*
Input: Feedback sets
Require: Storing the feedbacks in the file system

1: **begin**
2: Create two file objects of the directory name where the feedback files exist
3: Create two tables with the filed Feed_ID, User_ID, Size having the data type number for each one
4: **while** counter is not greater than directory length of the director **do**
5: Insert the file name of directory [counter] into User_ID field of the corresponding feedback table
6: Increment the value of counter corresponding to the table by 1
7: **end while**
8: **end**

- **Stop Word Removal**

  Stopwords are auxiliary terms and are used most frequently. The, here, to, a etc are some most frequent words in English and ‡K, †mLv‡b, Gi, GB etc. are some some in Bangla. The list of stopwords

is created from a file where the stopwords are stored manually. We store the stopwords of each language in a separate database table as it requires faster checking to verify whether any word is stopword or not. We remove the stopwords from the feedbacks by a look up at the table. When a new stopword is required to be added, it is just appended in the corresponding table.

- **Keyword Extraction**

  The keyword or term extractor sub-module extracts the root of every word using morphological analyzer. The algorithm for keyword processor takes words as input and gives the root of the words as output. Finding the root of a word is called stemming. To stem the postfixes from the terms of the written comment, the terms were checked against a postfix list. We used Porter Stemming Algorithm [23] to obtain the root words in English documents and the concept of [24] for obtaining the root words in Bangla documents. Bangla words can have many difficult variations by adding the postfixes with the root of the word. To stem the postfixes from the terms of the Bangla written feedback, the morphological analyzer checks the terms against a postfix list shown in [24] where the authors used a list of 143 postfixes for Bangla that are collected from Bangla grammar books and used this postfixes to obtain the keywords. **Algorithm 2** is given below for finding a root of Bangla words. This algorithm takes a word as input and gives the root of the words as output. The character sequences of postfixes are adopted from [24].

Algorithm 2: Finding a root of Bangla words
Input: List of terms T = t_1 , t_2 , . . . , t_x and list of postfixes P = p_1 , p_2 , . . . , p_y in a language
Require: Root words
1: **begin**
2: **for** each term t_i, $1 \leq i \leq x$ **do**
3: Check each character of t_i from last with the last character of p_j, $1 \leq j \leq y$
4: **if** a match is found then
5: Check next character of *ti* with the character at same position of p_j until a complete matching with p_j is found
6: Discard the matched part from *ti* and return remaining part as root
7: **else**
8: Discard the scanning of the corresponding term and return *ti* as root word r_i
9: **end if**
10: **end for**
11: **end**

4.1.2. Storage

Storage module stores information processed by database storing and processing module. Three storages are there, feedbacks databases will store all written feedbacks, stopword database will keep the list of stopwords and extracted keywords from written feedbacks are stored in keyword database. This total storage is required for the next score generation module to process the written feedbacks.

4.1.3. Score generation

This module generates score for retrieved feedback corresponding to each user ID. The keywords we

selected were categorized into several categories for generating score for each feedback.

Table 1. Keywords for Different Category

S. N.	Categorization	Keywords	Score
1.	Excellent	Attractive, excellent, best, wonderful, easy আকর্ষণীয়, অসাধারণ, ভালো, সহজ, বেশি ভালো, বেশি সহজ	5
2.	Very Good	Friendly, happy, nice, appreciate সমব্যাথী, খুশি, প্রশংসা, উৎসাহ	4
3.	Average	Difficult, less কঠিন, রূঢ়, মাঝে মাঝে, কম	3
4.	Poor	Risky, lack, unsafe, poor, unhealthy, overcrowd, improper ঝুঁকিপূর্ণ, অভাব, অপ্রতুল, অনিরাপদ, নিম্নমান, অস্বাস্থ্যকর, ভিড়,	2
5.	Terrible	Block, lock, force, danger বাধা, তালাবদ্ধ, বাধ্য, বিপদ	1

Table 1 shows some of the categories and their corresponding scores. The keywords from each feedback are matched with the categories of this table. When matched keywords in a users' feedback are found, we assume that feedback into the defined category and assign the corresponding score as shown in Table 2.

Table 2. Example of Score Generation

User ID	Feedbacks	Generated Score
U_1	*It's **risky** work. We often toil away in **unsafe** buildings where the exits and windows are often **block**ed.*	4
U_2	*Despite some progress, children are still involved in the production of textiles, where the working environment is **unsafe** for them.*	2
U_3	*Conditions are very **poor** to survive*. Demanding higher wages can prove deadly. One labor leader who did just that was found murdered last year.	2
U_4	*It's **difficult** to access the factories for safety purpose.* It took firefighters all night to put out a fire in a factory last year that killed more than 100 people, because the access road to the factory was difficult to traverse.	3
U_5	We are trapped inside our factory for hours. *We often are **lock**ed inside the building for our entire shift, sometimes longer if we work overtime.*	1
U_6	*Women work mainly as helpers, machinists and **less** frequently, as line supervisors and quality controllers.* There are no female cutting masters. Men dominate the administrative and management level jobs. Women are discriminated against in terms of access to higher-paid white collar and management positions.	3

U_7	*Taking the advantages of workers' poverty and ignorance the owners **forced** them to work in **unsafe** and **unhealthy** work place **overcrowded** with workers beyond capacity of the factory floor and **improper** ventilation*. It leads to a destruction that causes death to the workers. They also violate the safety code in order to gain the huge profit in view of owners.	2
U_8	The current pay scale given by the government is relatively very low in comparison to living standard. *But the environment is quite **friendly** as managers want to know our demands frequently.*	4
U_9	কারখানা শ্রমিকের কাজের পরিবেশ ঝুঁকিপূর্ণ। আমাদেরকে ধুলোবালি, ধোঁয়া, আগুন, গ্যাস, উচ্চ শব্দ ও বিপজ্জনক সরঞ্জামের সাথে অনেক উচ্চতায় অথবা ভূগর্ভেও কাজ করতে হয়।	2
U_{10}	যাতায়াতের পথ কারখানার কাঁচামাল ও যন্ত্রপাতি দিয়ে বন্ধ। জরুরী প্রয়োজনে নিরাপদ স্থানে যাওয়ার জন্য বের হওয়ার পথের অভাব রয়েছে।যে কয়েকটা সিঁড়ি আছে, তাও ঠিকমত রক্ষণাবেক্ষণ করা হয় না।	2

We have considered different criteria for evaluating a factory like sanitary condition, working environment, job security, owner behavior, promotion policy etc. Table III shows the extracted root words of the written feedback after stopword removal for the user ID U_1 to U_3. Proposed system will search each comment based on user id for matching the evaluating criteria. For example, the comment of user id U_1 contains the words work in the first sentence and environment in the second sentence. So we can define the comment about working condition of the factory. Now this extracted root words are matched with the selected keywords with score from Table 1 and the corresponding score is generated for each sentence. Table III shows the procedure of generating scores for user IDs of Table 2. For example first sentence of U_1 contains the word risk whose score is 2 and the second sentence contains the words unsafe (score=2) and block (score=1). As the second sentence have two matched keywords, so we will take the average score of these two words. Finally the overall score will be the summation of the scores applied for two sentences individually. Again the comment of user id U_3 contains two different criteria. First sentence is about poor working condition of the factory and the second sentence is indicating low salary structure of the same factory. So separate score will be generated for each criterion. In this proposed system, we have selected some evaluating criteria for rendering feedback through multiple choice and the others are left for written feedback. As we have selected salary structure for multiple choice section, so we will not consider it again in comment section. So, our overall generated score for the comment from user id U_3 is applied by considering only his first sentence (underlined one) about working environment. As the extracted root words contain the keyword poor whose score is 2 in Table 1, so the generated score is 2 there for user id U_3 as shown in Table 3.

Table 3. Score Generation Procedure

User ID	Extracted Root Words	Score
U_1	*Risky*, work. Toil, *unsafe*, build, exit, window, *block*.	Risky=2, unsafe+block=2+1=3 Overall= 2+3/2=2+2(round off)=4
U_2	Some, progress, children, involve, product, textil, work, environment, *unsafe.*	Unsafe=2

U_3	Condition, very, *poor*, survive. Demand, high, wage, prove, dead. One, labor, leader, found, murder, last, year.	Poor=2

Algorithm 3 is the procedure of generating the score that are shown in Table 3.

Algorithm 3: Score Generation for Written Feedback
Input: DB of extracted root words and DB of keywords with corresponding score
Require: Generated score for each feedback
1: **begin**
2: **for** each root word of a sentence
3: Check the word with the DB of keywords
4: **if** a match is found in a sentence
5: Determine the corresponding score of the matched word
6: **else if** another match is found in the same sentence
7: Determine the corresponding score of the next matched word
8: Compute the average of these scores of a single sentence
9: Continue until the end of sentence
7: **else** return null
8: **end for**
9: **end**

f_2... f_k. Numeric features are directly passed to this phase. Non-numeric feedbacks on any features are converted to numeric digits. To preserve the privacy of individual's, instead of publishing the exact feedbacks of each user, we have to publish the feedback results in such a way that the feedbacks information will be accurate while privacy of individual's is preserved. Table 4 shows the information of ten users where the feedbacks of features $f_1$and f_2 are given in numeric digits and f_3 are converted to numeric digits from written comment. Corresponding written comments of feature f_3 are shown in Table 2. Instead of publishing the information of Table 2, we utilize a framework to publish aggregated information that will preserve individual's privacy.

Table 4. Users' Feedback

User ID	f_1	f_2	f_3
U_1	10	5	4
U_2	15	3	2
U_3	22	4	2
U_4	36	1	3
U_5	11	4	1
U_6	13	3	3
U_7	20	2	2
U_8	18	2	4

U_9	40	3	2
U_{10}	30	5	2

The **algorithm 4** is given below for finding an integrated table for the feedbacks given in a language. The algorithm collects the values of the feedback that are given through numeric values (step 2-4) and then again collects the values from preprocessing module where written feedbacks are processed to convert into respective numeric values. Then the both collected values are merged into a single table for further processing (step 5-8).

Algorithm 4: Data Integration
Input: List of feedback values in numeric $N = n_1, n_2, \ldots, n_x$ and list of feedback values extracted from written comment $W = w_1, w_2, \ldots, w_x$ in a language for each user.
Require: Aggregated values correspond to each user
1: **begin**
2: **for** each term n_i, $1 \leq i \leq x$ **do**
3: Collect numeric feedback values for an user against the user ID i
4: **end for**
5: **for** each term w_j, $1 \leq j \leq x$ **do**
6: Collect numeric feedback values extracted from written feedback in preprocessing module for an user against the user ID j
7: Merge the values in a single table with collected values for numeric feedback for each user ID
8: **end for**
9: **end**

4.3. Aggregated Feedback Generation

For secure feedback processing data values of all features are divided into some portions, where the sum of these portions is actually the real data values. Table 5 shows the division of the data values that are listed in Table 4.

Table 5. Division of Data Values

User ID	Division of f_1					Division of f_2					Division of f_3				
	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{31}	f_{32}	f_{33}	f_{34}	f_{35}
U_1	3	8	-3	4	-2	1	2	-1	3	0	3	-2	7	0	-4
U_2	3	5	2	4	1	5	0	-2	1	-1	4	3	-4	-2	1
U_3	12	-4	6	7	1	1	2	-3	3	1	0	2	-5	1	4
U_4	7	13	5	-3	14	2	3	-2	-1	-1	5	-6	-1	7	-2
U_5	4	1	2	3	1	6	-4	1	2	-1	9	-6	2	-8	4
U_6	2	1	5	3	2	2	-3	0	3	1	4	-1	-7	2	5
U_7	8	-3	10	3	2	1	3	-4	-2	-1	3	0	8	-5	-4

U_8	5	3	-3	8	5	4	7	-8	0	2	2	-3	6	7	-8
U_9	10	10	-5	15	10	9	4	-7	-4	1	5	-11	-1	7	2
U_{10}	10	2	3	8	7	1	3	-4	6	-1	0	9	-2	-4	-1

We assume there is a coordinator who is responsible for calculating the feedbacks by divide-and-conquer strategy. The coordinator first asks each user within the system to divide each of its data values into *s* parts in such a way that the sum of *s* parts is equal to the data value. Each user then randomly divides each of its data values into distinctive *s* parts. As for example, each data value of Table 4 has been divided into five parts as shown in Table 5. Based on the number of users involve in feedbacks computation, the coordinator then creates a number of groups and assign *s* agents to each group. In general, if there are more users in the system more groups are created.

For example, the users' of system has been divided into two groups and there are five agents for each group as shown in Fig. 3. Later, the coordinator assigns token, a unique identifier for each agent within a group. It then sends the agents to the users of the groups. Upon arrival of an agent to a user in a group, the agent asks for data values. The user then provides a single part of each data value to the agent. The agent then goes to the next user of the group and asks the user for data values. The user also provides a single part of each data value to the agent. At this moment, the agent adds the new values with the values already in the agent. After completing the traversal of all users within the group, the agent contains "local sum" in its data structure and goes back to the coordinator. For each token, the coordinator then computes the "global sum" considering the "local sum" of the groups. Based on this "global sum", the coordinator computes the average of the users' feedbacks. During the process, agents are used to preserve privacy of users' data. Note that all the groups' computations are performed simultaneously.

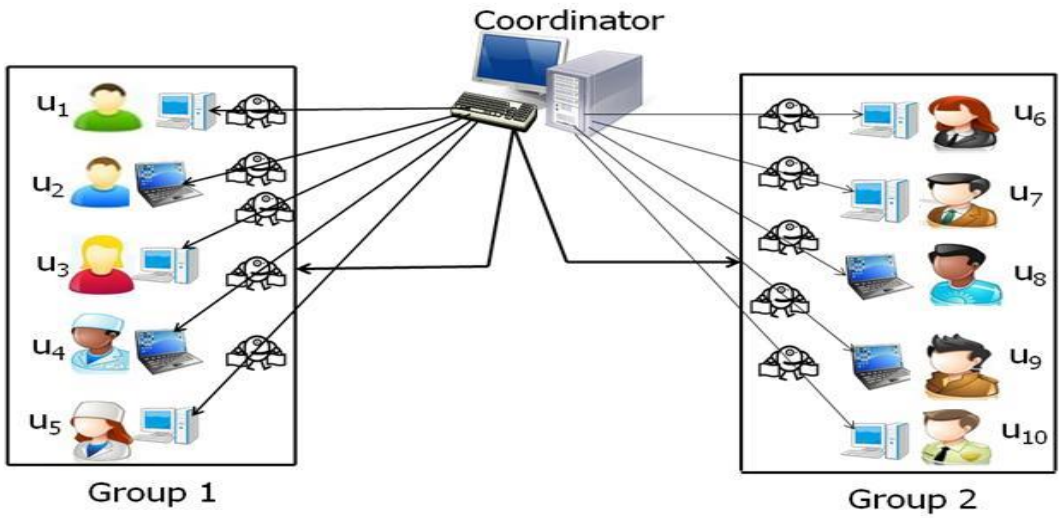

Fig. 3. Example of divide-and-conquer computation.

Considering the secure computation of feedbacks from users of Table 4, for each group the coordinator creates five agents and assigns a separate token *T*1, *T*2, *T*3, *T*4, and *T*5 to each agent of the group. Each agent has an "array data structure" to keep parts of data values. Initially the array contains 0 in each of its index. Each agent travels the users in a circular way pre-defined by the coordinator. Fig. 4 shows the computation process in group 1 with token *T*1. When an agent arrives at a user of a group, it asks for a part

of each of data values from the user. The user then push a part of each data value to the agent. The agent then adds the values with the values already in the array. The agent then moves to the next user and performs same tasks. Note that during this process, the user cannot see the contents of the array of the agent. In the example of Fig. 4, it is observed that agent with token *T1* visits the users in the order (u_1> u_2> u_3> u_4> u_5). From Fig. 3, we can see that u_1 pushes (f_{11}; f_{21}; f_{31}) = (3, 1, 3) to the agent. The agent then adds these values with the initial contents of the array. Next, the agent goes to user u_2. User u_2 then pushes (f_{11}; f_{21}; f_{31}) = (3, 5, 4) to the agent. Here, the contents of the array updates as ((3 + 3) = 6), ((1 + 5) = 6) and ((3+4) =7). The agent then traverses to users'u_3, u_4 and $u_5$in a sequential order as defined by the coordinator. During the traversal of any user, the agent performs the same tasks like u_1 and u_2. After visiting all the users in the group, agent with token *T1* contains (f_1; f_2; f_3) = (29, 15, 21) in its array and returns back to the coordinator.

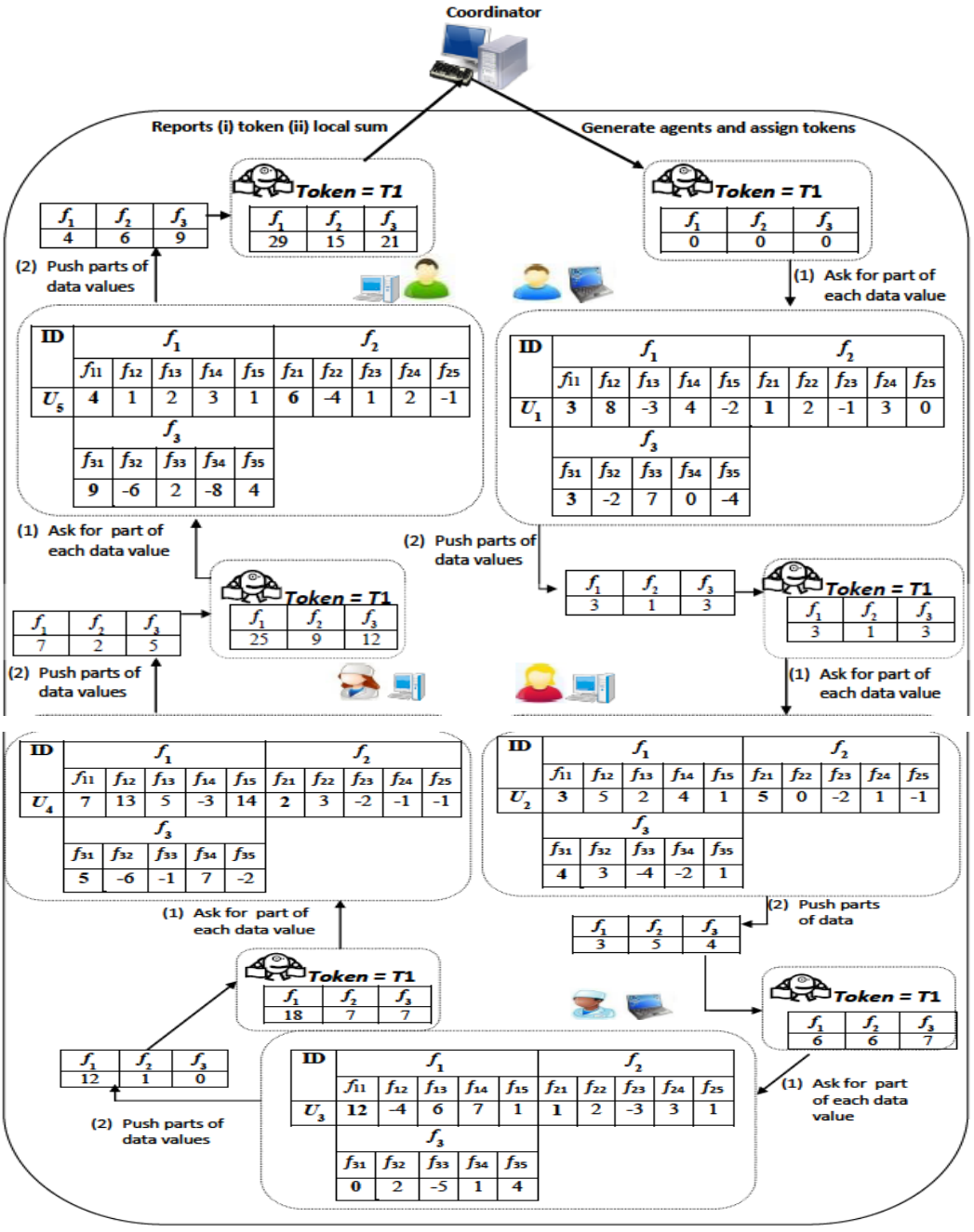

Fig. 4. Computation in group 1 with token *T1.*

Fig. 5 shows similar computation in the group 1 with token *T2*. Here, in order to minimize idle time, the traversals order of the agent in different. It starts its traversal from user u_2 and follows the order (u_2> u_3> u_4> u_5> u_1). After the computation, the agent contains (f_1; f_2; f_3) = (23, 3, -9) in its array and goes to the coordinator and reports the results. The agents with tokens *T*3, *T*4, and *T*5 perform similar computations with traversal order (u_3> u_4> u_5> u_1>u_2), (u_4> u_5> u_1> u_2> u_3), and (u_5> u_1> u_2> u_3> u_4) and reports results (f_1; f_2; f_3) = (12, -7, -1), (f_1; f_2) = (15, 8, -2), and (f_1; f_2; f_3) = (15, -2, 3) respectively to the coordinator. These are the "local sum" for group 1. During these processes, "local sum" in group 2 is computed simultaneously.

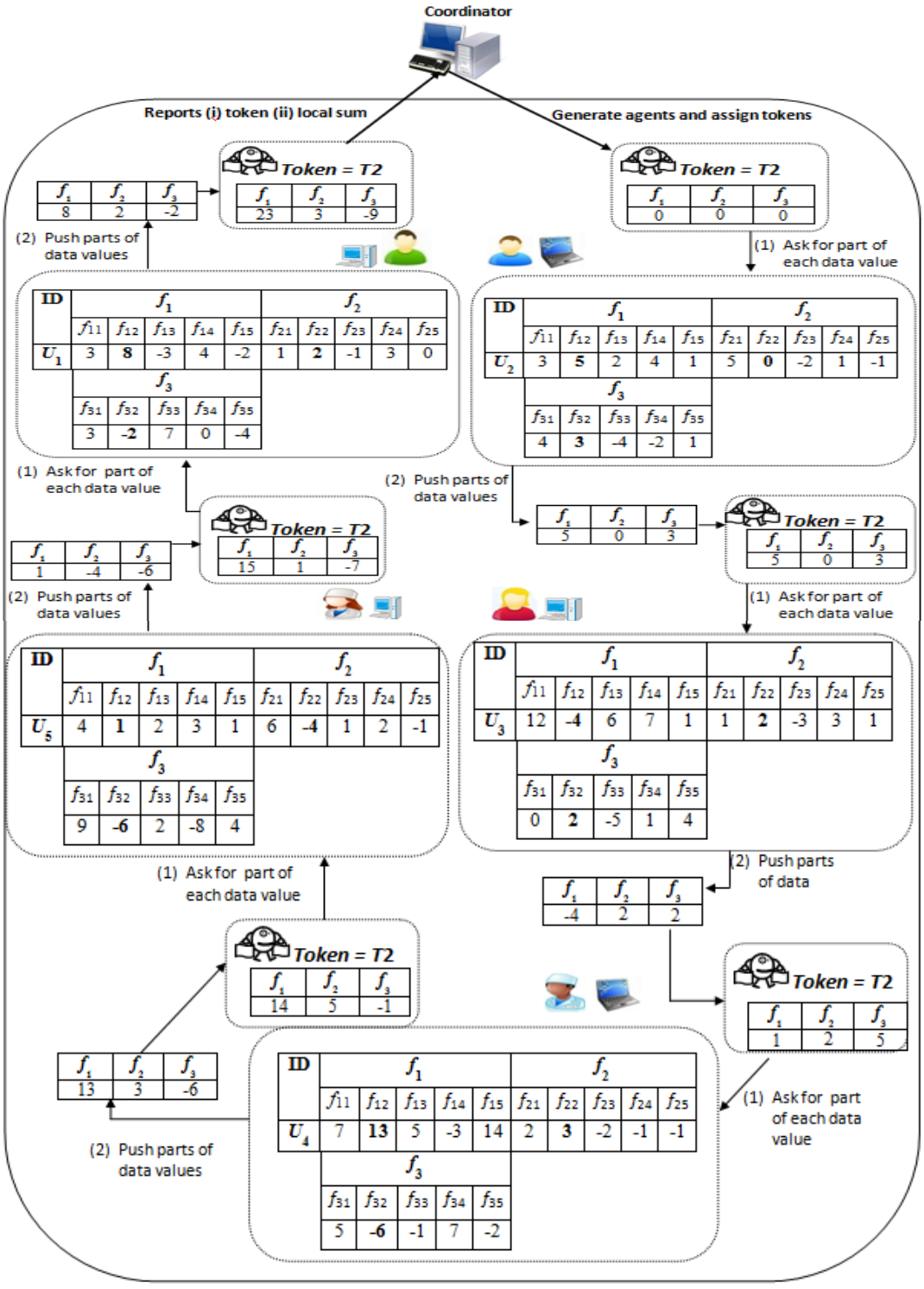

Fig. 5. Computation in group 1 with token *T2*.

Algorithm 5 is given below for the computation within a single group. Each token traverse the users of the group and compute the local sums for all features based on the divided parts of each data values of the features (step 4-6). And at step 7 this local sum is reports to the coordinator.

Algorithm 5: Computation within Group
Input: Divided data values for each features, **F = f₁ , f₂ , . . . , f x**
Require: Local sum for each token
1: **begin**
2: Let *ag(αt), (1 ≤ t ≤ x)* be the agents
3: Assign unique identifiers (token) for each agent
4: **for** each token **do**
5: *ag(αt)* travels users in the group and compute local sums for each feature collecting a single divided part from the features
6: **end for**
7: Coordinator receives local sums of *ag(αt) (1 ≤ t ≤ x)* for each token in the group
8: **end**

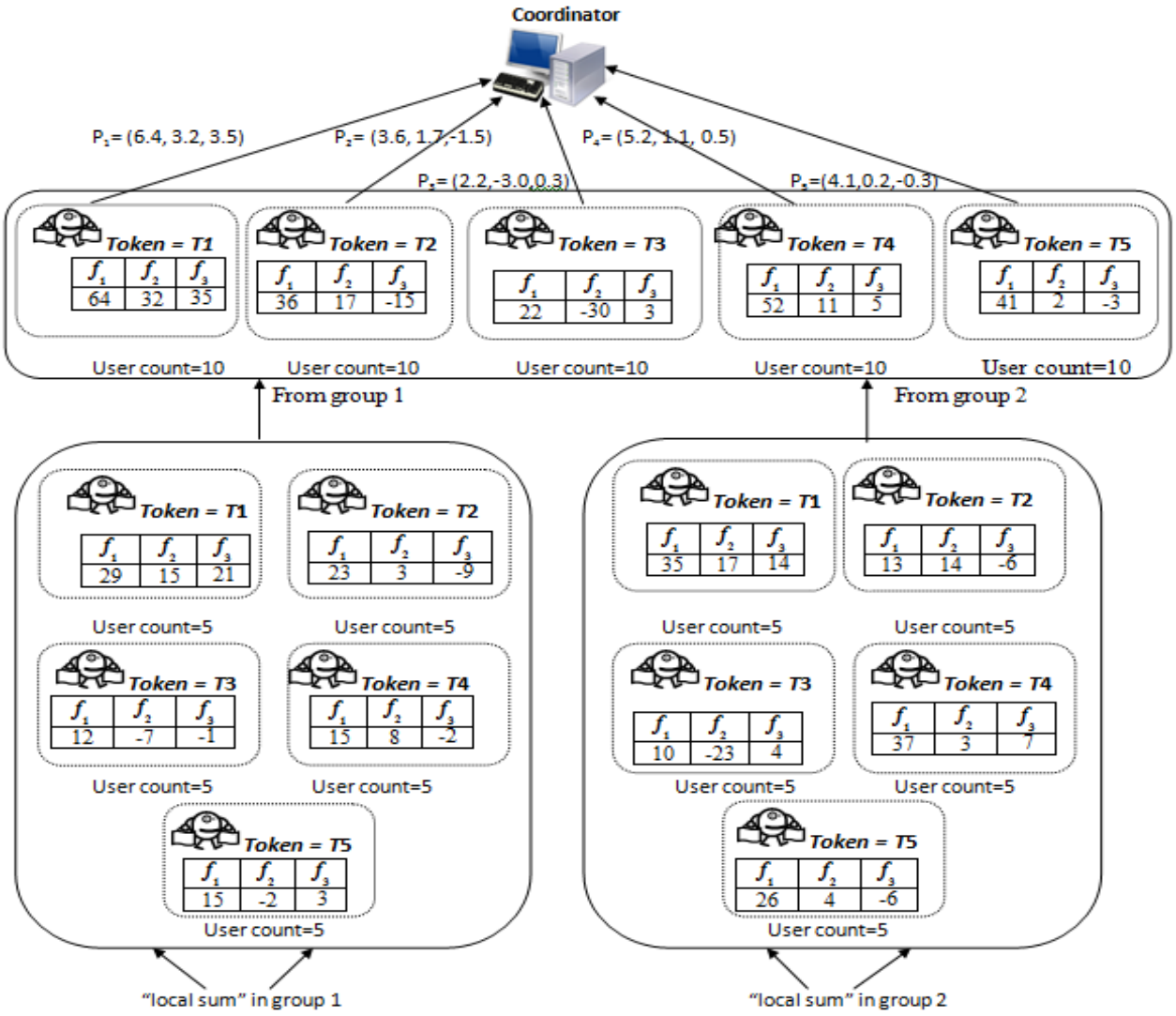

Fig. 6. Merge process at the coordinator.

After the agent-based computation in two groups, the coordinator first computes the "global sum" for each token. Then, the agents return the average values based on the "global sum" to the coordinator. Finally, the coordinator performs addition operation among the average values received from the agents to obtain the final feedback results in aggregated form. Fig. 6 shows the final computation at the coordinator.

From the example of Fig. 6, we get average values (6.4, 3.2, 3.5), (3.6, 1.7, -1.5), (2.2, -3.0, 0.3), (5.2, 1.1, 0.5), and (4.1, 0.2, -0.3) for tokens $T1$, $T2$, $T3$, $T4$, and $T5$ respectively. After adding these values, we obtain (21.5, 3.2, 2.5) as final feedbacks in aggregated form. Note that the obtained results in our system are same as the results obtained by averaging the values of each attribute of Table 4 separately. However, in our system there is no discloser of users' feedbacks information. **Algorithm 6** is for the merging process at the coordinator. Global sums are computed for each token that have collected the local sums by traversing on all users of different groups in the system (step 2-4). Then the average of these global sums for each token is calculated by dividing the summed values by the total number of users from whom the token has collected data values and returns the average values to the coordinator (step 5-8).

Algorithm 6: Global Merging
Input: Local sums of $ag(\alpha_t)$ $(1 \le t \le x)$ for each token within different groups and number of users within each group
Require: Aggregated values correspond to each $ag(\alpha_t)$, $(1 \le t \le x)$
1: **begin**
2: **for** each token **t (t= 1 to x) do**
3: Compute global sum of $ag(\alpha_t)$ from local sum for each group
4: **end for**
5: **for each** $ag(\alpha_t)$, $(1 \le t \le x)$ do
6: Compute the average of the global sum of each features
7: **return** the aggregated values to the coordinator
8: **end for**
9: **end**

Note that our system is well scalable in case of large number of users. This is because we have divided users in groups and computation in groups is carried out in parallel that minimizes the time of computation. More over our system protects data from third party access and modification. This due to the fact in our approach we utilize an agent-based computation and our agents can communicate using TLS/SSL that provides strong security while agents moves from one user to another user. In addition, our system is well secure from the modification of feedback information by dishonest users. This is because in our system, we do not need to send parts of data values of a user to other users.

5. Experimental Analysis

We have implemented our proposed privacy aware feedback system using Java Agent Development Framework. We have performed the experiment in a simulation environment of a PC running on windows OS having an Intel(R) Core i5, 2.5 GHz CPU, and 4 GB main memory.

5.1. Analysis of Processing Written Feedback

We developed a corpus with 500 feedbacks where 300 feedbacks are in English and rests 200 are in Bangla. These feedbacks were collected from different survey reports on garment factories in Bangladesh, different websites where interview of garments workers are uploaded like [25] and some other websites like

amazon.com [1], where customers provide their reviews on products and about the website itself. We assume that all texts are unicode supported.

We performed stemming on the total dataset initially with forty feedbacks and incrementally added words from more forty feedbacks in the system. Table 6 and 7 depict the number of words that requires stemming and the required time for stemming of different number of words from each feedback written in Bangla and English respectively. By analyzing the stemming performance for both English and Bangla words the number of stemming required increases with the increase of feedback number and also the stemming time increases almost linearly with the increase of document number. But the average required time for English word stemming is little bit lower than Bangla word stemming as the Bangla word processing requires to handle composite letters.

Table 6. Stemming Results for Bangla Words

Feedback Number	Total No. of words	Words stemmed	Required time (m-sec)
40	2213	1490	743
80	4521	1987	854
120	6486	2549	1103
160	9325	3361	1390
200	11933	4201	1815

Table 7. Stemming Results for English Words

Feedback number	Total No. of words	Words stemmed	Required time (m-sec)
40	2569	1592	668
80	4911	2701	1134
120	7386	4031	1661
160	9747	5248	2204
200	12146	7287	3060
240	14521	8712	3659
280	16932	9497	3988

5.2. Analysis of Aggregated Feedback Generation Procedure

5.2.1. Experiments with synthetic data

Total analysis of the proposed system requires huge amount of data. Due to the lack of enough real data,

we evaluate this portion of our proposed algorithm using synthetic datasets only.

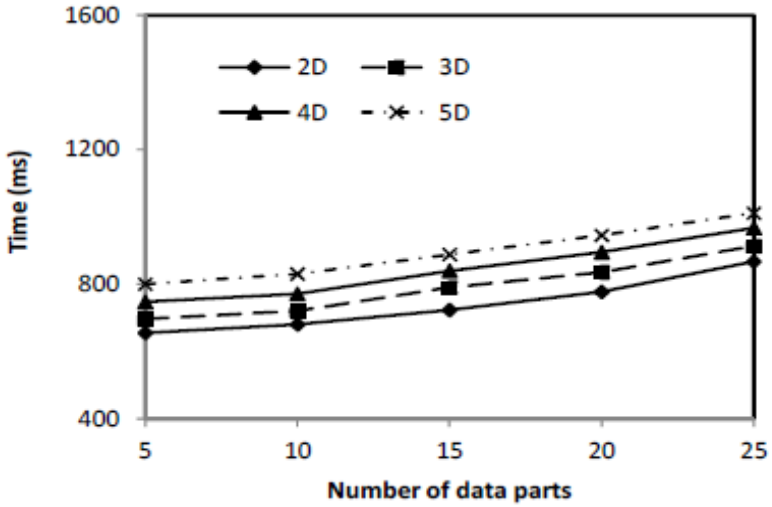

Fig. 7. Time varying number of data parts.

We first evaluate the effect of data parts *s*. Fig. 7 shows the results when we consider two (2D), three (3D), four (4D), and five (5D) features while distributing 40000 users among twenty groups and each group contains around two thousand users. We observe that with the increases of *s*, there is very slight increase in computation time. This is because during the computation process each agent in a group follows a different execution order that increase parallelism in computation. In addition, the computation among the groups is also performed in parallel. We can also observe that computation time gradually increases if the number of features increases.

In the next experiment, we evaluate the effect of the number of users involve in the system. In this experiment, we considered 20000, 40000, 60000, 80000, and 100000 users. Same as the previous experiment, users were distributed among twenty groups. In case of 20000 users, each group contains around 1000 users. Similarly, for 20000, 40000, 60000, 80000, and 100000 users, each group contains around 2000, 3000, 4000, and 5000 users. In this experiment, we set *s* to 10. Fig. 8 shows the results. In this experiment, it is observed that in case of fixed number of groups, response time increases with the increase of the number of users. Also, note that there is an increase of response time with the increase in the number of features.

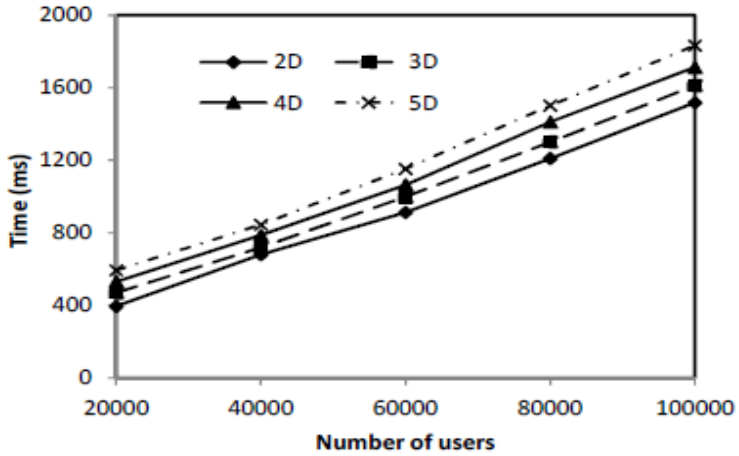

Fig. 8. Time varying number of users.

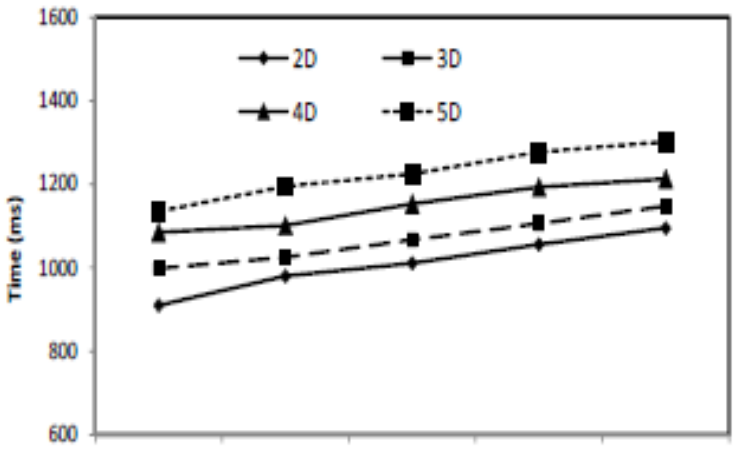

Fig. 9. Time varying number of users and number of groups.

Finally, we conduct the experiment to examine the effects while number of users and number of groups both increases. In this experiment, we distribute 20000, 40000, 60000, 80000, 100000 users among 5, 10, 15, 20, 25 groups, respectively. In this experiment, we set *s* to 10 and examine 2D, 3D, 4D, and 5D cases. Fig. 9 shows the results. From the results, we can find that in case of more users, if we create more groups, there is almost no performance degradation.

5.2.2. Experiments using real data

We have considered 500 written comments that are collected from different websites. Here in the experimental process, time required from preprocessing to final aggregated feedback generation is evaluated.

We have analyzed the effect of changing number of data parts *s*. Fig. 10 shows the result for written corpus where five features are considered. 500 users are distributed among 5 groups and each group contains around 100 users. The result states that with the increase of data parts *s*, there is little increase in computation time. Parallelism in final feedback computation process has lowered the total computation time.

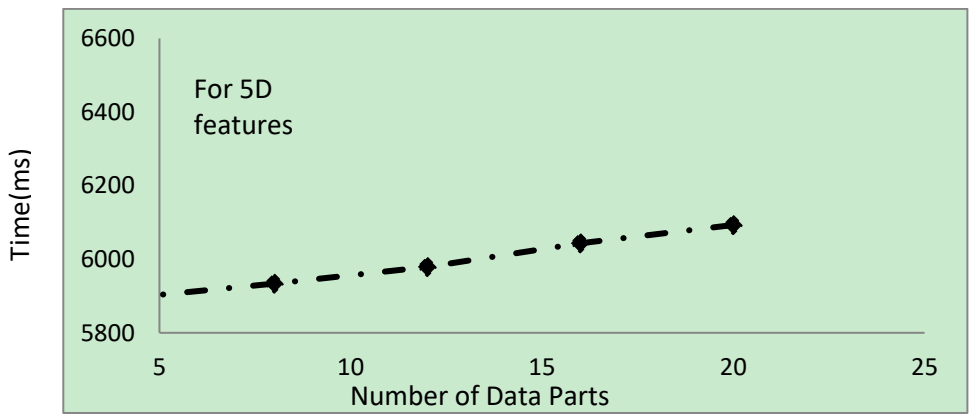

Fig. 10. Time varying number of data parts for written corpus.

In the next experiment, we evaluate the implemented system by changing number of users in the system.

Users were distributed among 5 groups. So, for 100, 200, 300, 400 and 500 users, each group contains around 20, 40, 60, 80 and 100 users respectively. In this experiment we set *s* to 5. Fig. 11 shows that for fixed number of data parts and groups, computation time increases with the increase of number of users.

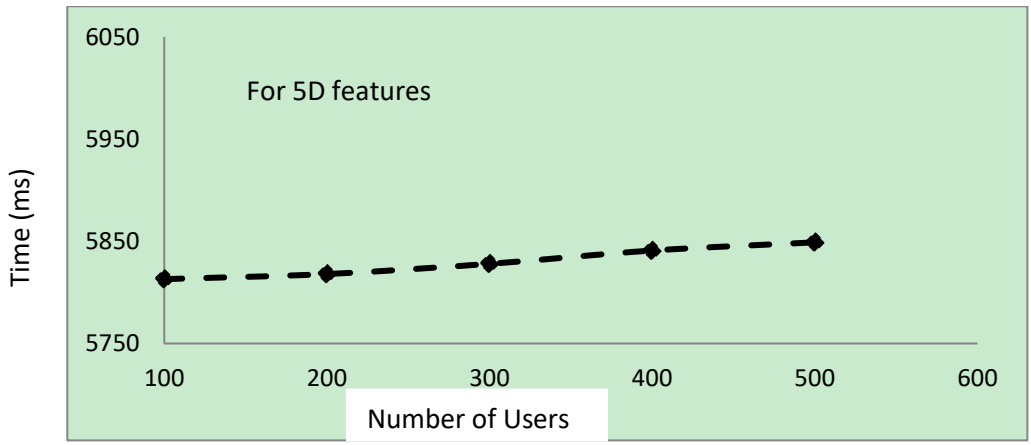

Fig. 11. Time varying number of users for written corpus.

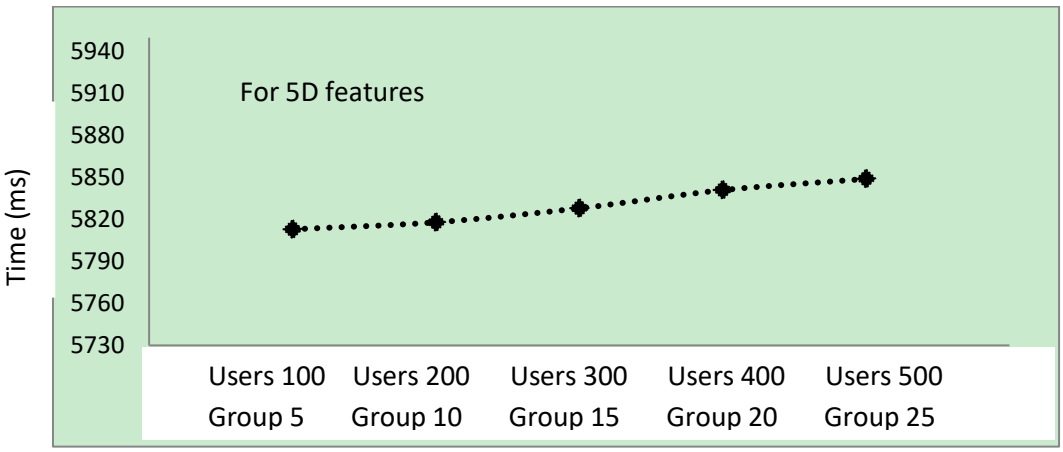

Fig. 12. Time varying number of users and number of groups for written corpus.

Finally, we conduct the next experiment to observe the effect when number of users and number of groups both increases. Here, we distribute 100, 200, 300, 400 and 500 users among 5, 10, 15, 20 and 25 groups respectively. In this experiment we set *s* to 5 and examine 5D cases for written corpus. Fig. 12 shows that the result is same like the synthetic data. If we create more groups for more users, almost no performance degradation is observed here.

6. Conclusion

With the rapid growth of network infrastructure, collecting feedbacks form the users via Internet are becoming popular. In privacy aware environment, users do not want to disclose their identities due to several factors. Therefore, we proposed an agent-based algorithm for computing feedbacks in a parallel manner from the users. The proposed algorithm can efficiently collect users' feedback while preserving individual's privacy. Our system can handle written feedbacks as well as numerical feedbacks. Our system can handle written feedbacks of two different languages: English and Bangla. We have conducted experiments on real data and synthetic data. Experimental results demonstrate that the proposed algorithm for feedbacks collection is scalable enough to handle large number of users. Although we have

considered, two different languages for written feedbacks, the system can be expanded to adapt other languages with a slight modification of our system.

References

[1] Amazon E-commerce Company. (2017). Retrieved from http://www.amazon.com

[2] eBay Inc. (2015). Retrieved from http://www.ebay.com

[3] Olex. (2015). Retrieved from http://www.olx.com

[4] Agoda Company Pte Ltd. (2015). Retrieved from http://www.agoda.com

[5] TripAdvisor. (2015). Retrieved from http://www.tripadvisor.com

[6] Expedia. (2015). Retrieved from http://www.expedia.com

[7] Rakuten. (2015). Retrieved from http://www.rakuten.com

[8] Elance. (2015). Retrieved from https://www.elance.com

[9] Bangladesh garment workers suffer poor conditions two years after reform vows. (2017). Retrieved from https://www.theguardian.com/world/2015/apr/22/garment-workers-in-bangladesh-still-suffering-two-years-after-factory-collapse

[10] 2013 Savar building collapse. (2015). Retrieved from http://en.wikipedia.org/wiki/2013

[11] Emam, K. E., & Dankar, F. K. (2008). Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association, 15(5)*, 627-637.

[12] Machanavajjhala, A., Gehrke, J., Kifer, D., & Venkitasubramaniam, M. (2006) *l*-diversity: Privacy beyond *k*- anonymity. *Proceedings of the 22nd International Conference on Data Engineering* (p. 24).

[13] Li, N., Li, T., & Venkatasubramanian, S. (2007). *t*-closeness: Privacy beyond *k*-anonymity and *l*-diversity. *Proceedings of the 23nd International Conference on Data Engineering* (pp. 106-115).

[14] Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (pp. 439-450).

[15] Zhan, J. Z., Matwin, S., & Chang, L. (2005). Privacy-preserving collaborative association rule mining. *Proceedings of the 28th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy* (pp. 153-165).

[16] Zhang, P., Tong, Y., Tang, S., & Yang, D. (2005). Privacy preserving naive bayes classification. *ADMA, 3584,* 744-752, Springer.

[17] Rizvi, S., & Haritsa, J. R. (2002). Maintaining data privacy in association rule mining. *Proceedings of the 28th International Conference on Very Large Data Bases* (pp. 682-693).

[18] Evfimievski, A. V., Srikant, R., Agrawal, R., & Gehrke, J. (2002). Privacy preserving mining of association rules. *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 217-228).

[19] Arefin, M. S., Mukta, R. B. M., & Morimoto, Y. (2014). Agent-based privacy aware feedback system. *Lecture Notes in Computer Science, LNCS (LNAI), 8933,* 725-738.

[20] Bergstrom, T., & Karahalios, K. (2007). Conversation votes: Enabling anonymous cues. *Proceedings of Computer / Human Interaction 2007* (pp. 2279-2284).

[21] Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., & Landay, J. A. (2007). My experience: A system for in situ tracing and capturing of user feedback on mobile phones. *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services* (pp. 57-70).

[22] Hashem, T. (2014). Privacy preserving feedback and monitoring system. *Proceedings of Workshop on Women Empowerment through ICT: Higher Studies, Research and Career* (pp. 30-31).

[23] The Porter Stemming Algorithm. (2017). Retrieved from http://snowball.tartarus.org/algorithms/porter/stemmer.html

[24] Arefin, M. S., Sharif, M. A., & Morimoto, Y. (2013). BAENPD: A bilingual plagiarism detector. *Journal of Computers, 8(5),* 1145-1156.

[25] Bangladesh garment workers hold largest wage protest yet. (2017). Retrieved from http://www.huffingtonpost.com/2013/09/21/bangladesh-wageprotest_n_3967390.html

Rahma Bintey Mufiz Mukta received her B.Sc. engineering and M.Sc. engineering in computer science and engineering from Chittagong University of Engineering and Technology (CUET), Bangladesh in 2013 and 2017 respectively. She is currently working as an assistant professor in the Department of Computer Science and Engineering, Chittagong University of Engineering and Technology (CUET), Chittagong, Bangladesh. Her research interest includes privacy preserving data mining, multilingual data management and machine learning.

Mohammad Shamsul Arefin received his B.Sc. engineering in computer science and engineering from Khulna University, Khulna, Bangladesh in 2002, and completed his M.Sc. engineering in computer science and engineering in 2008 from Bangladesh University of Engineering and Technology (BUET), Bangladesh. He has completed Ph.D from Hiroshima University, Japan. He is currently working a professor and head in the Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh. His research interest includes data privacy and data mining, cloud computing, big data, IT in agriculture, and OO system development.