A Privacy-Preserving System for Enhancing the QoI of Collected Data in a Smart Connected Community

Article

Accepted version

V. Agate, P. Ferraro, G. Lo Re

2024 IEEE Symposium on Computers and Communications (ISCC)

Publisher: IEEE

A Privacy-Preserving System for Enhancing the QoI of Collected Data in a Smart Connected Community

Vincenzo Agate†, Pierluca Ferraro† and Giuseppe Lo Re†
†Department of Engineering, University of Palermo, Palermo, Italy.
Email: vincenzo.agate@unipa.it, pierluca.ferraro@unipa.it, giuseppe.lore@unipa.it

***Abstract*—The Smart Connected Communities paradigm, which synergistically integrates smart technologies with the surrounding environment, has paved the way for a new generation of applications that provide increasingly intelligent services by leveraging information coming from users, and the IoT. While user collaboration is essential to improve the quality of information (QoI), the interest of providers in data can jeopardize the right to privacy by revealing details that users are not willing to share (e.g., habits, health status). In addition, not all involved users consistently exhibit cooperative behavior, and the presence of attackers often undermines the quality of the collected information. In this paper, we propose a system for aggregating and analyzing user data without ever compromising their privacy, whilst improving QoI. The system uses Privacy Preserving Computation techniques, clustering, and an outlier removal step to improve the quality of information. Utilizing a real-world dataset, we tested our system, demonstrating its resilience in a scenario with potential attackers and its superior performance compared to other state-of-the-art systems.**

***Index Terms*—Privacy, Data Aggregation, QoI, Smart Connected Community**

I. Motivations and Related Work

Nowadays, Smart Connected Communities (SSCs) leverage technologies to collect real-time data on various aspects of urban life, such as traffic, energy, waste, public safety, and air quality. This data is then analyzed to make informed decisions and adopt solutions that improve efficiency, sustainability, and quality of life in the community [1].

However, analyzying such data can present some challenges, since participants' responses can be subjective and influenced by factors such as personal experiences and expectations [2]. In addition, responses can be influenced by the wording of the questions themselves, which can be ambiguous or misleading; this can lead to conflicting results from the questionnaires. For example, different residents might have different opinions about the quality of public services based on their individual experiences. However, information gathered from questionnaires can still provide valuable insights into residents' needs and opinions. When analyzed together with other data sources, such as IoT sensors or publicly available government data, they can contribute to a better understanding of the context and drive decisions to improve the community [3], [4].

In cases where there is no single *truth* regarding the questions posed in a survey, the adoption of clustering techniques provides an insight into the diverse perspectives and opinions of population groups [5], without neglecting minorities.

Simple clustering, however, is not enough. Even in the case of small communities, some users may deliberately provide misleading responses to influence questionnaire results and promote their agendas. A system that identifies and eliminates such responses can help in preventing manipulation and preserving the integrity of the decision-making process.

At the same time, users' privacy must be maintained throughout the data collection and analysis process [6], even in traditional services such as recommendation [7] and trust-based applications [8]. When users know that their personal information is protected and treated confidentially, they are more likely to participate in questionnaires and share their opinions and data openly [9], [10]. Maintaining user privacy can also provide protection against abuse or discrimination, as personal information collected during questionnaires could be misused or distorted for discriminatory purposes or to damage people's reputations. However, eliminating misleading responses to improve QoI and safeguarding data privacy can be a significant challenge; in fact, these two goals can often conflict with each other.

In this paper, we propose a privacy-preserving data aggregation system that improves the quality of information of the collected data by discarding outliers. The system always operates on encrypted data throughout the process, so that the secrecy of the information sent by users is never compromised.

In this regard, homomorphic encryption allows a cloud server to perform operations on the encrypted data without having to decrypt it, thus preserving the privacy of the questionnaire answers. Using this technique, responses can be processed as an aggregate and clustered without the need to reveal personal information associated with individual responses.

Various research studies have explored privacy-preserving computation (PPC) techniques and homomorphic encryption [11]–[14]. A data aggregation framework referred to as PPTD, which is one of the first of its kind proposed in the literature in [15], leverages homomorphic encryption techniques. However, this system necessitates users to participate actively in decrypting intermediate aggregate data, imposing significant computational and communication burdens on them. Xu et al. [16] put forward a system that outperforms PPTD, but it does so by employing a symmetric key shared among all users, making it particularly susceptible to exploitation by attackers.

Outliers can negatively affect the QoI of questionnaires by skewing aggregate results; by using homomorphic encryption techniques, these outliers can be identified without revealing

individual responses. However, potential negative effects on diversity of opinion, information bias, information loss and discrimination should be carefully considered. The goal should be to ensure accurate, complete, and fair representation of aggregated information, taking into account potential problems and seeking solutions that minimize risks and maximize benefits to the Smart Connected Community. One of the most common limitations in privacy-preserving systems in the literature is that they require the direct participation of end users, who must remain connected throughout the process. Our system, on the other hand, involves users only in the initial phase, when they answer the questionnaire asynchronously and send their encrypted responses to the system.

The proposed system was tested on a real dataset to verify the effectiveness of the approach and compare it with other state-of-the-art works in a scenario with malicious users trying to attack the systems and alter the aggregate values. The results obtained show that the system performs better than its competitors and is effective in improving QoI while maintaining user privacy. The contribution we make through this paper can be summarized in the following points: 1) A novel privacy-preserving protocol for a data aggregation system that clusters answers coming from users; 2) A method to improve the QoI of collected data by discarding unreliable answers while preserving user privacy; 3) Extensive experiments on a real-world dataset that demonstrate the soundness of our approach and compare our system to other state-of-the-art work in a scenario with malicious users.

The rest of this paper is organized as follows. Section II presents the problem formulation and provides some background on privacy-preserving techniques. Our system is described in detail in Section III. Section IV presents our experimental results and Section V concludes the paper.

II. Problem formulation and Privacy Preserving Computation

In our system, we consider a cloud server (CS) that receives data from questionnaires filled in by users in encrypted form. Although the data is encrypted, the CS is able to aggregate data and remove noisy information without ever violating user privacy, while improving overall QoI.

We treat the CS as a semi-honest party, following the protocol without being able to deviate from it, but at the same time trying to find out as much as possible about its users based on the data it receives. As described by [11], this adversarial model is realistic for many practical scenarios.

In our problem, we consider n users $U = \{u_1, u_2, \ldots, u_n\}$ who are willing to share their knowledge on questions of interest to the community. Typically, users are encouraged to participate when rewarded by an incentive mechanism.

However, users may provide inaccurate answers about facts they do not know in order to obtain greater rewards. More specifically, in our model, users can be malicious and attack the system by providing false answers to the CS.

Given a set of m questions $Q = \{q_1, q_2, \ldots, q_m\}$ that each of the users is asked to answer, a user i will send an answer v_{ij} to the CS for each question j. The CS will then privately aggregate the user's responses. While individual responses may seem innocuous on their own, an analysis of the entire set of a user's responses could reveal information about habits, personal data, and much more, seriously compromising the user's privacy. For this reason, it is essential to work only on encrypted user data, which is unreadable to the CS.

A. Secure Multi-Party Computation

Secure Multi-Party Computation, in contrast to classic cryptosystems [17], exploits some probabilistic homomorphic properties provided by relevant encryption schemes to make multiple parties jointly contribute one or more functions on inputs that will remain private.

The Paillier encryption scheme [18] is an ideal fit for our scenario. Firstly, it is probabilistic, adding an extra layer of security to our system. A probabilistic encryption scheme generates distinct ciphertexts for the same plaintext; this is especially valuable in our application scenario to preserve confidentiality, since the range of potential responses is often limited (such as yes/no questions). Moreover, Paillier's encryption scheme is also additively homomorphic. This feature allows us to perform a series of operations on the ciphertext domain. The goal is that, at the end of the protocol, we can derive aggregate results based on the responses from the users, without compromising their privacy. In an additively homomorphic encryption scheme, given two plaintext messages m_1 and m_2, and their encrypted versions $E(m_1)$ and $E(m_2)$ using the same public key pair (n, g), the following two properties hold:

$$E(m_1) * E(m_2) \ (mod\ n^2) = E(m_1 + m_2 \ (mod\ n)), \quad (1)$$

$$E(m_1)^{m_2} \ (mod\ n^2) = E(m_1 m_2 \ (mod\ n)). \quad (2)$$

Given the multiplication between the ciphertext versions of two messages, the additive property of Equation 1 allows us to obtain the ciphertext version of the sum of the two messages. With Equation 2, we get the multiplication of two messages in the ciphertext domain [19].

The properties described above are not sufficient, by themselves, to perform all the calculations required for clustering answers and removing outliers. For example, while it is possible to compute the product of an encrypted value and a plaintext value, it is not possible to directly compute the product of two encrypted values. Some varieties of homomorphic schemes, called *fully* homomorphic, allow for this kind of operation as well, but are too inefficient in practice. For this reason, the cryptographic system we have chosen, namely Paillier, is *partially* homomorphic and has only the additive homomorphic properties, as described above. Therefore, to perform more complex operations, a third party (also considered semi-honest, as will be explained in the next section) is needed to help the CS.

Suppose our CS wants to send an encrypted user message to a Third Party (TP) that has the private key to decrypt it. In theory, the TP can decrypt the data, perform the operations

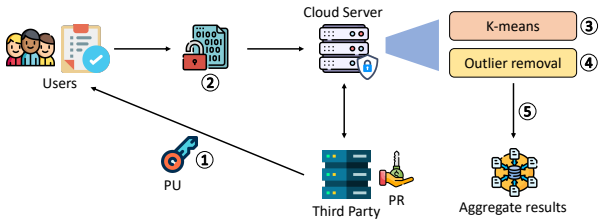

Fig. 1. Architecture of the proposed system.

that the CS could not do on its own, re-encrypt the message and re-send it to the CS. The only problem with this is that the TP would get to know user data in unencrypted form. To avoid that, however, the CS can add noise to the encrypted data before sending it to the TP, by adding a random value r using the additive property. The TP, at this point, will be working (in plaintext) on meaningless data that does not reveal users' personal information, if r has the right random properties. After the TP has made the necessary calculations and re-encrypted the data, the CS can remove the previously added noise, working only on encrypted data due to the homomorphic properties described earlier. This technique is called *blinding*, and will be used in the following to allow the CS to perform complex calculations that would otherwise be impossible.

III. Proposed System

In this section, we present our privacy-preserving data aggregation system to improve the QoI of collected data, utilizing the Paillier homomorphic cryptosystem to ensure secure and efficient processing. The goal of the CS is to cluster data in a privacy-preserving manner and remove outliers, taking advantage of homomorphic encryption to always work on encrypted data. To do this, the CS uses a privacy-preserving version of the K-means, and exploits a threshold mechanism to remove outliers.

We opted for the K-means clustering algorithm as it provides a balanced trade-off between computational efficiency and clustering performance, ensuring that we can achieve high-quality information aggregation while preserving user privacy.

As explained earlier, the CS cannot, by itself, perform all the necessary operations on encrypted data. This is because the CS cannot know the private key used in the protocol; otherwise, it could simply decrypt the secret data sent by users, jeopardizing their privacy.

For this reason, we introduce into our architecture a semi-honest Third Party (TP), which will have two tasks: (1) generate a public/private key pair that will be used to encrypt/decrypt values (communicating only the public key to the CS and to the users) and (2) help the CS by performing some complex calculations, always working on *blinded* data with noise added. It is, of course, crucial that the CS and the TP cannot collude, as stipulated by our adversarial model in which both are considered semi-honest and must necessarily follow the protocol. To ensure that the properties of homomorphic encryption work, it is necessary for all users to use the same public/private key pair. Ciphertexts resulting from encryption with different keys are obviously not combinable with each other. The TP then generates these two keys at the beginning of the protocol and safely stores them.

When the CS requests help from the TP to perform some of the calculations, it will always use the blinding techniques described in Section II-A so as to ensure user privacy. Therefore, the TP will have a supporting role in the calculations and will never learn the data sent by users without alterations.

To summarize, Fig. 1 shows how our system operates, in order to ensure the privacy and secrecy of user data throughout the process and improve QoI: 1) Users receive a public key from the TP to use in the rest of the protocol. 2) Users send questionnaire responses already encrypted with their public key to the CS. 3) The CS aggregates user answers using a privacy-preserving version of the K-means, working only on encrypted data. 4) The CS eliminates outliers and recalculates cluster centroids to improve QoI. 5) The CS sends back aggregated results.

Our protocol starts with the TP generating an asymmetric key pair (PU_k and PR_k), and sending only the public key, PU_k, to users and CS. At this point, each user u_i encrypts its private responses v_{ij} with PU_k, and sends them to the CS. For efficiency reasons, the users also compute and send the encrypted version of each v_{ij}^2 to the CS. This will allow the CS to calculate the distances of each user from the cluster centroids efficiently, interacting with them only to get initial encrypted data.

The CS has the task of computing the K-means in a privacy-preserving manner, by iteratively updating the centroid values until a termination condition occurs (e.g., after a certain number of iterations). For a questionnaire with m questions, the CS must first randomly determine the initial values of k m-dimensional centroids, corresponding to the k clusters of the K-Means. We define these centroids as $C_1, C_2, \ldots, C_k$.

At the beginning of each iteration, the CS must then calculate the distances between the responses sent by each user, $V_i = v_1, v_2, \ldots, v_m$ and the k centroids. We define these distances as $D_i = d_{i1}, d_{i2}, \ldots, d_{ik}$. Specifically, given a user i and a cluster with index l, d_{il} is computed as follows:

$$d_{il} = \sum_{j=1}^{m} (v_{ij} - c_{lj})^2. \tag{3}$$

The CS must calculate this value in encrypted form, $E(d_{il})$, which can be done by using the following equations:

$$\begin{aligned} E(d_{il}) &= E(\sum_{j=1}^{m} (v_{ij} - c_{lj})^2) \\ &= E(\sum_{j=1}^{m} v_{ij}^2) \cdot E(\sum_{j=1}^{m} v_{ij} \cdot (-2c_{lj})) \cdot E(\sum_{j=1}^{m} c_{lj}^2) \\ &= \prod_{j=1}^{m} E(v_{ij}^2) \cdot \prod_{j=1}^{m} E(v_{ij})^{-2c_{lj}} \cdot \prod_{j=1}^{m} E(c_{lj}^2). \end{aligned} \tag{4}$$

Equation 4 can be computed directly by the CS, without requiring the intervention of the TP. This is because the CS

knows the values of the centroids in plaintext (c_{lj}) and can encrypt them with PU_k; moreover, both $E(v_{ij})$ and $E(v_{ij}^2)$ are sent directly by each user u_i. This allows the CS to obtain the encrypted version of d_{il} for each user and each centroid.

The next step is to locate, for each user, the centroid with the minimum distance, namely $\arg\min_l d_{il}$. To identify the $\arg\min$, the idea is to use an iterative process that compares the elements of the D_i vector in pairs keeping only the lower of the two values. At each iteration, the number of minimum candidates is halved. After $\log_2 k$ iterations, the CS gets the encrypted minimum. The necessary comparison operations can be executed with an interactive protocol, which is omitted due to space limitations (please refer to [14] for a full discussion on the subject). The result of the sub-protocol is the encrypted version of an m-dimensional vector A_i, with $a_{il} = 1$ if u_i belongs to the l-th cluster, or 0 otherwise. Recall that the adopted cryptosystem always produces different ciphertexts when encrypting the same value (e.g., 0 or 1) multiple times, as explained in Section II-A; if this were not the case, the CS could easily discover the clusters to which users belong.

At this point, the CS has allocated the user responses to the different clusters for the current iteration (in encrypted form), and can calculate the updated centroids as follows:

$$E(C_l) = E\left(\frac{1}{|U^l|}\sum_{i=1}^{n} V_i \cdot a_{il}\right), \tag{5}$$

where $|U^l|$ is the cardinality of the set of users belonging to the l-th cluster. The CS exploits the blinding techniques presented in section II-A to perform the multiplications within the summation, which involve a vector of encrypted values, V_i, and an encrypted scalar value, a_{il}. With the help of the TP, the CS can thus compute, for each j, $E(v_{ij} \cdot a_{il})$. The summation itself can be computed by repeatedly applying the additive homomorphic property. Finally, the product with the term $1/|U^l|$ can be accomplished by exploiting blinding techniques once again.

Equation 5 is used to compute the encrypted version of the centroids for each cluster. To conclude the iteration, the CS and TP together can decrypt the centroids, leveraging again blinding techniques. The centroids do not represent sensitive data and cannot in any way be traced back to the individual users who submitted the answers, since neither CS nor TP know which users belong to the different clusters. The values of the centroids thus derived are then used by the CS in the next iteration to compute the encrypted distances with the user responses, continuing the K-means algorithm until the termination condition. After the K-means is finished, the last step of the protocol is to remove outliers. This requires the CS to compute one last time the distance of the user responses from all the centroids and to determine for each one the minimum distance, $E(d_{il})$, and the encrypted vector, $E(A_i)$, as defined above. Outliers are removed by exploiting a threshold θ. The CS compares $E(d_{il})$ with θ in a privacy-preserving manner, using the procedure explained in [11]. The result of the comparison, $E(o_i)$, will also be in encrypted form.

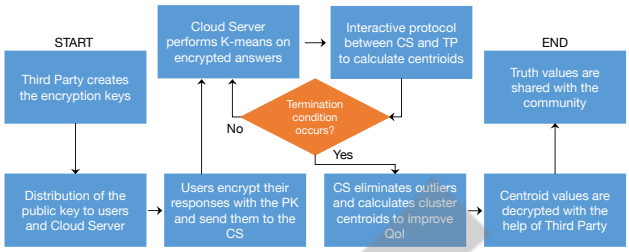

Fig. 2. Flowchart of the operations carried out in our system

The value o_i, in particular, will be 0 if the answers of the i-th user are considered an outlier, 1 otherwise.

The goal of the CS, at this point, is to compute, for each user, a new vector $E(\tilde{A}_i)$ that represents, in encrypted form, whether that particular user's responses are outliers or not. The CS then exploits the values $\tilde{A}_i$ instead of A_i to update the centroids one last time, as described above. The vector $E(\tilde{A}_i)$ for each user i can be computed from o_i and A_i by leveraging blinding techniques, as follows:

$$E(\tilde{A}_i) = E(A_i \cdot o_i). \tag{6}$$

Finally, the CS obtains the k centroid values of the K-means, without outliers. The flow chart in Figure 2 summarizes all the system operation steps described so far.

In some scenarios, it may be necessary to determine a single "truth" value for each task. In such cases, different aggregation functions can be used. For example, the CS could perform a weighted average over the centroids of the different clusters, or select the cluster that is considered *best* according to some metric (e.g., the one with the most users). In other cases, it may make more sense to keep all clusters separate to represent the opinions of different groups of users.

IV. Experimental Results

In this section we evaluate our system, testing it with a real world dataset to verify the effectiveness of our approach and comparing our results with other state-of the art works. In particular, we carried out a series of experiments using the "City Population" dataset[1], which contains 43,071 claims on the number of inhabitants of various cities [20]. These claims have been made by more than 4,000 users over the years. The dataset also includes a ground truth regarding the number of inhabitants of 308 cities, taken from U.S. census data.

If a user makes several claims on the same city, we only consider the last reported value. Also, given that the dataset is quite sparse, we have conducted our experiments on a subset of cities, considering those that have received the most reports from users. Finally, we applied preprocessing steps similar to those indicated by [21]. Considering that the number of inhabitants varies widely from one city to another, all values have been appropriately normalized.

[1]https://cogcomp.seas.upenn.edu/page/resource_view/16

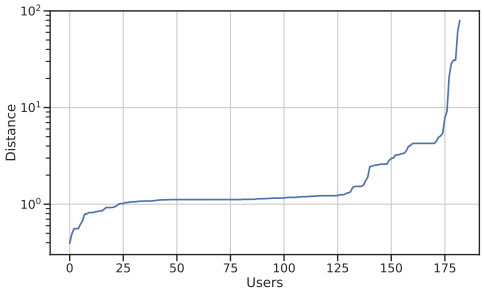

Fig. 3. Users distance from the closest centroid with k = 2.

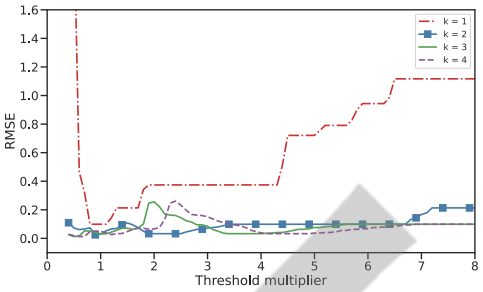

Fig. 4. RMSE obtained with various k and threshold multipliers.

It is important to consider how the QoI obtained by clustering responses and removing outliers can be measured objectively. In the case of questionnaires with correct answers, such as the chosen dataset, the overall accuracy of the results can be assessed. Several commonly used metrics can be used for this purpose. Obviously, such improvements in QoI are more noticeable in the case of direct attacks against the system, such as attacks that aim to increase noise with random responses, or targeted slandering attacks.

In the case of questionnaires with subjective responses, quantifying improvements can be more challenging. However, even in these cases, removing outliers can help reduce noise in the data and improve the consistency of responses. As a result, the aggregate results will be more reliable and representative, which can be considered an objective improvement in terms of QoI. However, as argued above, potential negative effects on information bias and diversity of opinion resulting from the elimination of perceived "inaccurate" responses must be taken into account. In our case, the correct answers are objective, so we can evaluate the system directly.

In the following experiments, the main measure of system accuracy that we considered is the RMSE (root mean square error). Since our K-means randomly selects the initial centroids, the experiments have been repeated 100 times and we present the average of the results obtained. The main parameters to consider are the number of clusters, k, and the threshold to discard outliers. To identify the best value of k, we used one of the most popular approaches regarding K-means, the *elbow method*, which in our case suggested to use 2 clusters. Figure 3 shows the Euclidean distance of the user values to their nearest centroid with 2 clusters, with a logarithmic scale on the y-axis. Unsurprisingly, considering the crowdsourced nature of the data collected, a small number of users send very precise data, most send data with some inaccuracy, and a small number of users make completely unfounded claims.

The choice of the threshold to use for discarding outliers was the focus of another set of experiments. In the following, we define the outlier threshold as a multiplier based on the average distance of the user responses from their respective centroids. For example, a threshold of 2 will discard responses whose distance from their own centroids is more than twice the average. Figure 4 shows the RMSE values achieved by experimenting with both the number of clusters k and the threshold multiplier. Since the RMSE is an error measure, systems with lower RMSE values are the ones with the best performance. As expected, the system with $k = 1$ is the one with higher RMSE values, always proving to be the worst regardless of the threshold. All other systems obtain comparable performance. This in itself proves that the clustering of responses already improves the accuracy of the results. However, removing outliers can further improve the QoI.

Based on the threshold definition, the higher the multiplier in the x-axis, the more responses are included in the results. When the threshold decreases, instead, more outliers are discarded. In general, as the threshold grows the RMSE values tend to increase, flattening out when the threshold is high enough and the system includes all or almost all responses, except for the most obvious outliers which are always discarded unless very high threshold multipliers are used. Figure 4 also shows that, with respect to the analyzed dataset, threshold values around 1 offer optimal performance for all systems. Further experiments performed on synthetic datasets confirm these findings, resulting in optimal thresholds ranging from 1 to 2. Finally, we compared our system with a family of privacy-preserving data aggregation systems. PPTD [15] is inspired by the well-known truth discovery system CRH, and differs from other similar systems in aspects related to the privacy-preserving techniques adopted, while using very similar formulas in terms of improving QoI. Indeed, L-PPTD [22], EPTD [16], LPTD-I [23], and RPTD-I [24] achieve the same results as PPTD in our usage scenario. For the sake of presentation, this family of systems will be represented by PPTD.

To better compare the systems, even in the presence of noisy data or malicious users, we added synthetic attackers to the dataset; these attackers occasionally send random responses to the system, with uniform distribution. Fig. 5 shows the RMSE obtained by two versions of our system (named k1 and k2, respectively, depending on the number of clusters) and by PPTD, as the number of attackers increases from 0 to 200 (50% of the total users) in the x-axis. The threshold multiplier used by both k1 and k2 is 1.0. Regardless of the number of attackers, both of our systems perform better than

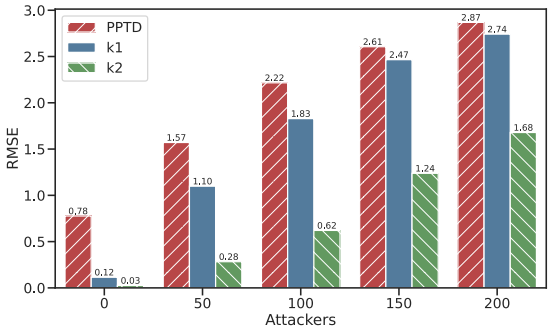

Fig. 5. Comparison with the family of systems represented by PPTD.

PPTD and achieve lower RMSE values than the competitor. In particular, the system with $k = 2$ consistently performs better than the one with $k = 1$. PPTD, on the other hand, achieves an RMSE that increases significantly as the number of attackers grows. As the number of attackers increases, PPTD obtaines an average RMSE of 2.01, compared with 1.652 in our system with $k = 1$ and 0.77 in our system with $k = 2$, which is 62% lower than PPTD.

V. Conclusions

In this work, we have introduced a privacy-preserving data aggregation system that improves the quality of information collected within a Smart Connected Community. To accomplish this task, private data submitted by users is clustered using a modified version of the K-means algorithm, which works directly on encrypted data without the need to decrypt it first. The resulting clustering is then used to discard outliers in a privacy-preserving manner, to improve QoI. Experimental tests conducted on a real-world dataset have demonstrated the effectiveness of our approach, which outperforms other state-of-the-art work. The integration of a reputation management system [25] while preserving user privacy seems to be an interesting direction for future research, as it might improve the quality of the results in the long run.

Acknowledgments

This work is partially funded by the European Union Next-Generation EU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.3 – D.D. 1551.11–10-2022, PE00000004) - MICS (Made in Italy – Circular and Sustainable) Extended Partnership and partially funded by the European Union - PON Ricerca e Innovazione 2014-2020 - DM 1062/2021.

References

[1] A. R. Khamesi, R. Musmeci, S. Silvestri, and D. A. Baker, "Reproducibility of survey results: A new method to quantify similarity of human subject pools," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–7.

[2] M. Kaiser, "Benford's law as an indicator of survey reliability—can we trust our data?" *Journal of Economic Surveys*, vol. 33, no. 5, pp. 1602–1618, 2019.

[3] Y. Sun, H. Song, A. J. Jara, and R. Bie, "Internet of things and big data analytics for smart and connected communities," *IEEE Access*, vol. 4, pp. 766–773, 2016.

[4] D. Gupta, S. Bhatt, M. Gupta, and A. S. Tosun, "Future smart connected communities to fight Covid-19 outbreak," *Internet of Things*, vol. 13, p. 100342, 2021.

[5] Y. Liu, L. Kong, and G. Chen, "Data-oriented mobile crowdsensing: A comprehensive survey," *IEEE communications surveys & tutorials*, vol. 21, no. 3, pp. 2849–2885, 2019.

[6] Y. Cheng, J. Ma, Z. Liu, Z. Li, Y. Wu, C. Dong, and R. Li, "A privacy-preserving and reputation-based truth discovery framework in mobile crowdsensing," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[7] P. Ferraro and G. Lo Re, "Designing ontology-driven recommender systems for tourism," in *Advances onto the Internet of Things*. Springer, 2014, pp. 339–352.

[8] V. Agate, A. De Paola, G. Lo Re, and M. Morana, "Vulnerability evaluation of distributed reputation management systems," in *Proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS'16, 2017, p. 235–242.

[9] A. Kanaan, A. AL-Hawamleh, A. Abulfaraj, H. Al-Kaseasbeh, and A. Alorfi, "The effect of quality, security and privacy factors on trust and intention to use e-government services," *International Journal of Data and Network Science*, vol. 7, no. 1, pp. 185–198, 2023.

[10] V. Agate, A. De Paola, P. Ferraro, G. Lo Re, and M. Morana, "Secureballot: A secure open source e-voting system," *Journal of Network and Computer Applications*, vol. 191, 2021.

[11] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Magazine*, vol. 30, no. 1, 2012.

[12] V. Agate, P. Ferraro, G. Lo Re, and S. K. Das, "Blind: A privacy preserving truth discovery system for mobile crowdsensing," *Journal of Network and Computer Applications*, vol. 223, 2024.

[13] S. Rane and P. T. Boufounos, "Privacy-preserving nearest neighbor methods: Comparing signals without revealing them," *IEEE Signal Processing Magazine*, vol. 30, no. 2, 2013.

[14] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *International symposium on privacy enhancing technologies symposium*. Springer, 2009.

[15] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015.

[16] G. Xu, H. Li, C. Tan, D. Liu, Y. Dai, and K. Yang, "Achieving efficient and privacy-preserving truth discovery in crowd sensing systems," *Computers & Security*, vol. 69, 2017.

[17] V. Agate, F. Concone, A. De Paola, P. Ferraro, G. Lo Re, and M. Morana, "Bayesian modeling for differential cryptanalysis of block ciphers: a des instance," *IEEE Access*, vol. 11, pp. 4809–4820, 2023.

[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999.

[19] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, no. 4, jul 2018.

[20] J. Pasternack and D. Roth, "Knowing what to believe (when you already know something)," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010.

[21] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han, "A confidence-aware approach for truth discovery on long-tail data," *Proceedings of the VLDB Endowment*, vol. 8, no. 4, 2014.

[22] C. Miao, L. Su, W. Jiang, Y. Li, and M. Tian, "A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[23] C. Zhang, L. Zhu, C. Xu, K. Sharif, X. Du, and M. Guizani, "Lptd: Achieving lightweight and privacy-preserving truth discovery in ciot," *Future Generation Computer Systems*, vol. 90, pp. 175–184, 2019.

[24] C. Zhang, L. Zhu, C. Xu, X. Liu, and K. Sharif, "Reliable and privacy-preserving truth discovery for mobile crowdsensing systems," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[25] V. Agate, A. De Paola, G. Lo Re, and M. Morana, "A platform for the evaluation of distributed reputation algorithms," in *2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, 2018, pp. 1–8.