On Anonymizing Streaming Crime Data: A Solution Approach for Resource Constrained Environments

Aderonke Busayo Sakpere, Anne Kayem

On Anonymizing Streaming Crime Data: A Solution Approach for Resource Constrained Environments

Aderonke Busayo Sakpere[1], Anne V.D.M. kayem[2]

[1]Department of Computer Science, University of Cape Town, Cape Town, South Africa, olfade001@myuct.ac.za
[2]Internet Technologies and Systems Group, Hasso-Plattner-Institute, Potsdam, Germany, anne@mykayem.org

Abstract. A typical resource constrained environment is restrained in terms of availability of resources such as skilled personnel, equipments, power and Internet connectivity. Designing privacy-based service-oriented architectures therefore requires re-adapting existing solutions to cope with the constraints of the environment. In this paper, we consider the case of mobile crime-reporting systems that have emerged as an effective and efficient data collection method in developing countries. Analyzing the data, can be helpful in addressing crime but, law enforcement agencies in resource-constrained contexts typically do not have the expertise required to handle these tasks. A possible cost-effective strategy is thus to outsource the data analytics operations to third-party service providers. However, the sensitivity of the data makes privacy an important consideration. In this paper we propose a two-pronged approach to addressing the issue of privacy in outsourcing crime data in resource constrained contexts. We build on this in the second step to propose a streaming data anonymization algorithm to analyse reported data based on occurrence rate rather than at a preset time on a static repository. Results from our prototype implementation and usability tests indicate that having a usable and covet crime-reporting application encourages users to declare crime occurrences and anonymizing streaming data contributes to faster crime resolution times.

1 Introduction

While organizations generate data that can contribute to improving performance daily, many of these organizations do not have the in-house expertise required to analyse the data. The lack of expertise is prominent in resource constrained environments manifested in rural/remote developing world regions, for instance, where constraints on resources such as access to computational power, reliable electricity, and the Internet pose a further challenge. A cost-effective solution is to outsource the data to a professional third-party data analytics service provider.

A study [1] carried out in technologically resource-constrained environments has revealed that collected crime data are usually not studied or analysed to

support crime resolution. A possible reason for this is the lack of the necessary in-house expertise, both in terms of human capital and computational processing power [15, 5, 25]. This deprives policy makers in these regions of the benefits that could have been derived through data analytics. A possible solution to this is to involve a third-party data analytics service provider [1, 2]. However, because of the sensitive nature of crime data it makes sense to ensure that the outsourced data are protected from all unauthorized access including that of an honest-but-curious data mining service provider. Therefore, this paper focuses on developing a test bed framework to preserve privacy during real-time information sharing using the crime domain as an application scenario. However, it is important to stress that the ideas and approaches considered in this study are applicable to other areas or domains as well.

2 Related Work

A naive approach to preserve privacy or anonymity in data is to exclude explicit identifiers such as name and/or identification number. However linking attacks aimed at data deanonymisation, can be provoked successfully by combining non-explicit identifiers (such as date of birth, address and sex) with external or publicly available data [3, 26, 27]. To illustrate how a linking attack can be provoked, let us consider Figure 1, which shows two compartments (or storage) that contains data. The upper compartment contains a portion of a publicly available table in which name is an explicit identifier attribute and the lower compartment shows a portion of a data stream that has been sanitized to exclude explicit identifiers (name) in order to disguise the identities of the individuals associated with the data. However, when a joining operation is performed on both compartments using attributes common to both compartments, the supposedly anonymized individual is re-identified successfully as Ade who lives at 10 Pope Street and also revealing her sensitive information that she has been a victim of rape.

According to Sweeney [18, 19] 87% of the population in the United States were uniquely identified by the combination of non-explicit identifiers such as gender, zip code and date of birth from the 1990 census dataset using linking attack. Therefore, Sweeney et al. came up with a better approach named k-anonymity to anonymize data in a manner that linking attack is minimized.

K-anonymity ensures privacy is preserved by hiding each individual in a cluster which contains at least k individuals such that an adversary finds it difficult to get additional individual information, but rather information about a group of k individuals [18, 20]. To understand how k-anonymity works, let us assume an attacker tries to identify a friend in a k-anonymized table, but the only information he has is her birth date and gender. K-anonymity ensures that the adversary finds it difficult to identify the individual by guaranteeing that at least k people have the same date of birth and gender. Thus minimizing the rate of linking attack to at least $1/k$. K-anonymity algorithms can generally be grouped into two categories, namely hierarchy-based generalization and hierarchy-free gener-

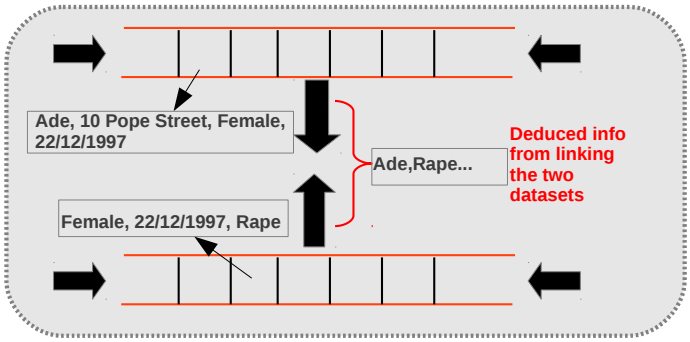

Fig. 1: Illustration of Linking Attack

alization[3]. In the hierarchy-based generalization, data anonymization requires a generalization tree as an input to aid in the anonymization process while in the hierarchy-free generalization, generalization tree is not required as an input rather the algorithm makes use of clustering concepts and some heuristics.

The evolution of k-anonymity has led to the birth of newer privacy models to address its inherent limitation. Some of the popular and newer privacy models that extend k-anonymity are ℓ-diversity and t-closeness [30, 31]. The main essence of ℓ-diversity is to address homogeneity attack to which k-anonymity is vulnerable and it does this by requiring that each cluster in a k-anonymized table has at least ℓ distinct sensitive values [30]. T-closeness further complements ℓ-diversity by ensuring that distribution of sensitive values in a cluster is similar to that of the entire anonymized table[31].

An equally fast-growing data preservation technique is differential privacy. Differential privacy achieves anonymization by altering the data (i.e. unanonymized data) with the addition of mathematical noise [13]. In other words, differential privacy preserves privacy through the difference between the data supplied and the noise added to it. Interestingly, recent research [17] [16] has shown that the use of t-closeness with k-anonymity can yield similar privacy result as those of differential privacy. In this research we focus on k-anonymity and its complementary techniques because of the simplicity [30], effectiveness [19] and high utility [32] offered, especially when compared to an evolving counterpart such as differential privacy. In addition, recent research [32] [33] has shown that differential privacy is achieved as long as a dataset is anonymized using k-anonymity and t-closeness. Therefore this paper focuses on the use of k-anonymity, ℓ-diversity and t-closeness to achieve anonymization.

The adaptation of k-anonymity & its complementaries to data stream (real-time data) has led current research to integrate the concept of a buffering (or sliding window) mechanism and delay constraint into data stream anonymization

[3, 26, 27, 4, 22, 28]. The buffer is designed to hold a portion of the data stream at every instant of time, after which an anonymization algorithm can be applied to data in the buffer. Delay constraints is required to put a check on each tuple so that it does not stay in the buffer beyond a pre-defined deadline. Inspite of this, many of the existing algorithms adapted for anonymization of data streams face the following challenges:

- First, buffering according to delay constraints, can result in certain records being held in the buffer for long periods [3, 23, 8]. When such records are time-sensitive or need to be processed in real time, occurrence of delay usually results in high levels of information loss. Since a key requirement of a good anonymization scheme is high data utility, high levels of information loss due to expired tuples or dropped (or suppressed/unanonymizable) records are undesirable.
- Second, building on the first problem, we note that many of the existing data stream anonymization schemes based on k-anonymity and its derivatives do not take distribution of future data streams into consideration during anonymization [4]. An implication of this is that a record that is likely to offer better anonymization at a lower rate of information loss in a future sliding window or data stream can be anonymized with such a future sliding window rather than the current sliding window or data stream. Therefore, there is a need to have a model that can predict the best sliding window or stream with which a record should be anonymized.

Therefore, the focus in this paper is to present a data-stream anonymization framework that addresses the aforementioned challenges inherent in existing framework. More detailed literature review can be found in [14].

3 Data Stream Anonymization Framework

Figure 2 presents an overview of our Data Stream Anonymization Framework using the crime domain as an application scenario. Users make crime reports electronically and the reports are anonymized in real time at the anonymization layer. The results from the anonymization layer are transferred to third party for data mining process at the application layer.

3.1 Users Layer: Crime Reporting Layer

As noted in previous sections, this research considers the crime domain as an application scenario for achieving data stream anonymization. However, the ideas in this research extend to any other domain that requires real-time anonymization of sensitive data. Thus, to enable us to create an application that allow people to report crime in a secured and covert way, we converted the existing paper-based crime reporting system of a University Campus Setting in South Africa into a digitized Crime Reporting System. We chose to use mobile device

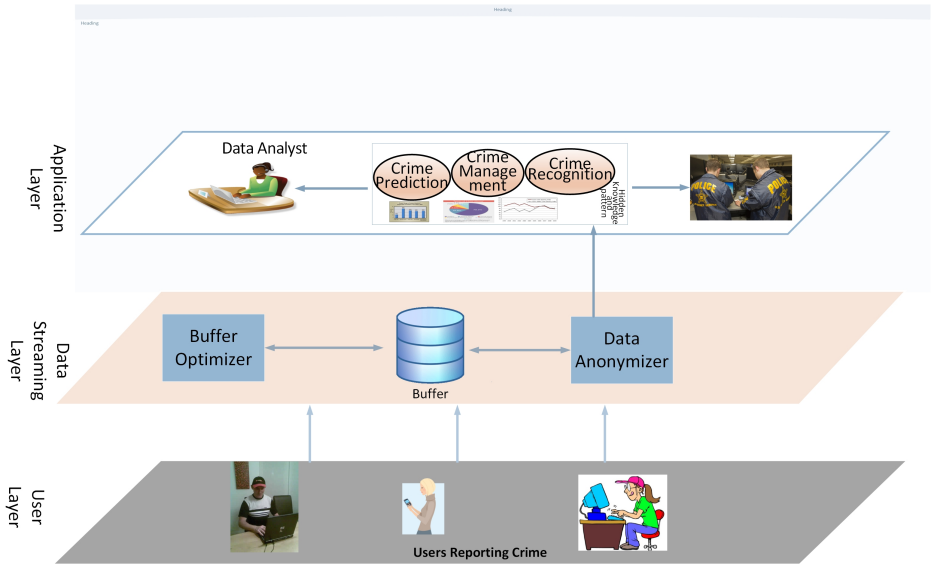

Fig. 2: System Overview

as platform for crime reporting because the use of mobile devices provides a good security platform for crime report [11] [24].

In order to ensure that our mobile crime reporting system is acceptable and usable in real-life we applied an iterative user-centered design methodology. To this effect we interviewed different key stakeholders within law enforcement agencies and crime victims. We had different iterations in our design until we came up with a final prototype acceptable to all stakeholders. Figure 3 presents the screenshots of our final prototype. More details about the research on the development and deployment of the crime reporting application,CryHelp, can be found in [5].

3.2 Anonymization Layer: Data Stream Anonymization

In our proposed crime reporting application scenario, data arrives in form of streams and contains information that is analyzed for statistical or data mining predictions. These data are temporarily stored in a buffer in order for anonymization to take place. A buffer is used to hold portions of the continuous data stream based on delay constraints that specify the duration for which tuples can remain in the buffer just before anonymization takes place. As illustrated in Figure 4, the buffer optimizer uses time-based sliding window and Poisson probability to monitor the data in the buffer, ensuring that tuples are anonymised before the expiry time threshold is reached while Figure 5 illustrates the data anonymizer which uses k-anonymity, ℓ-diversity and t-closeness for data privacy preservation. We opted for Poisson distribution because it is concerned with the number

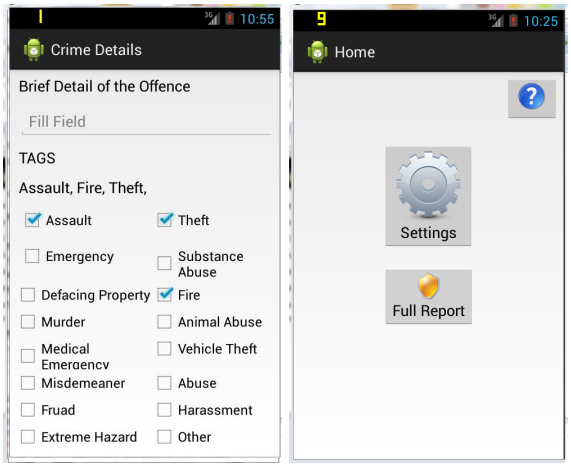

Fig. 3: Screenshot of our Crime Reporting Application

of success that occurs within a given unit of measure. This property of the Poisson distribution makes viewing the arrival rate of the reported crime data as a series of events occurring within a fixed time interval at an average rate that is independent of occurrence of the time of the last event [6]. Only one parameter needs to be known, the rate at which the events occur which in our case is the rate at which crime reporting occurs.

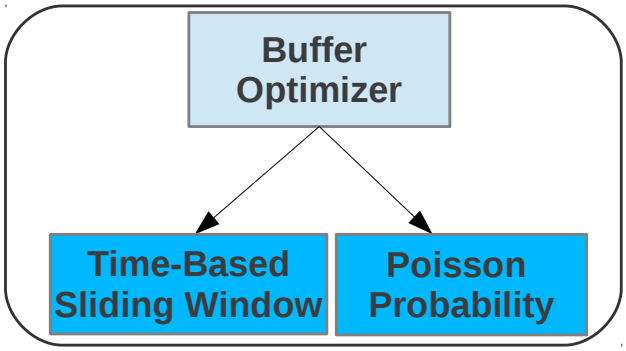

Fig. 4: Details of Anonymization Layer: Buffer Optimizer

4 Adaptive Buffer Re-Sizing Scheme

As illustrated in Figure 6, a "sliding window (buffer)", sw_i, is a subset of the data stream, DS, where DS = $\{\text{sw}_1, \text{sw}_2, \text{sw}_3, ..., \text{sw}_m\}$ implies that the data stream consists of m sliding windows. The sliding windows obey a total ordering such that for $i < j$, sw_i precedes sw_j. Each sliding window, sw_i only exists for a specific period of time T and consists of a finite and varying number of records, n, so that $\text{sw}_i = R_0, ..., R_{n-1}$.

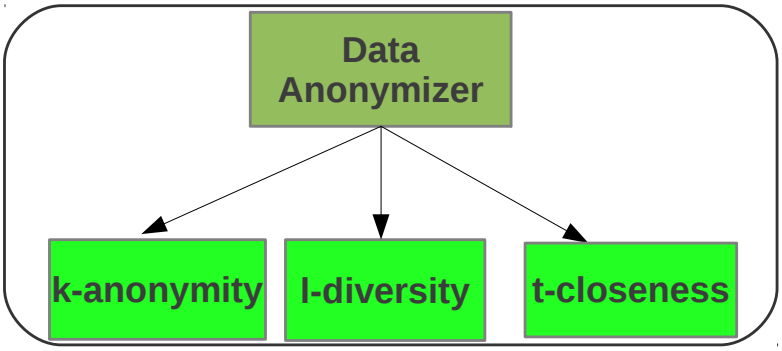

Fig. 5: Details of Anonymization Layer: Data Anonymizer

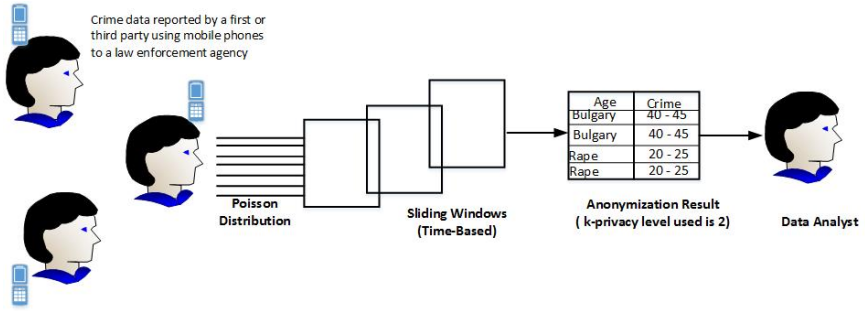

Fig. 6: Overview of Dynamic Buffer Sizing Process

Our adaptive buffer sizing scheme as illustrated in figure 7 is categorized into 6 phases and detailed explanation about how each of these phases work is in our earlier publication [12]. We summarize these details as follows. First we begin by setting the size of the buffer to some initial threshold value. Given the time-sensitivity of the data, we set the size of the sliding window, sw_i, to a value, T. T is a time value that is bounded by a lower bound value, t_l, and an upper bound value, t_u. The anonymization algorithm is applied to the data that was collected in the sliding window sw_i during the period T. So, essentially $\text{sw}_i = T$. All records that are not anonymizable from the data collected in sw_i are either included in a subsequent sliding window, say sw_{i+1} or incorporated into already anonymized clusters of data that are similar in content wise.

In order to determine whether or not an unanonymizable record can be included in a subsequent sliding window, say sw_{i+1}, we compute its expiry time T_E and compare its values to the bounds for acceptable sliding window sizes $[t_l, t_u]$. We compute T_E as follows:

$$T_E = \text{sw}_i - T_S - T_A \quad ...(1)$$

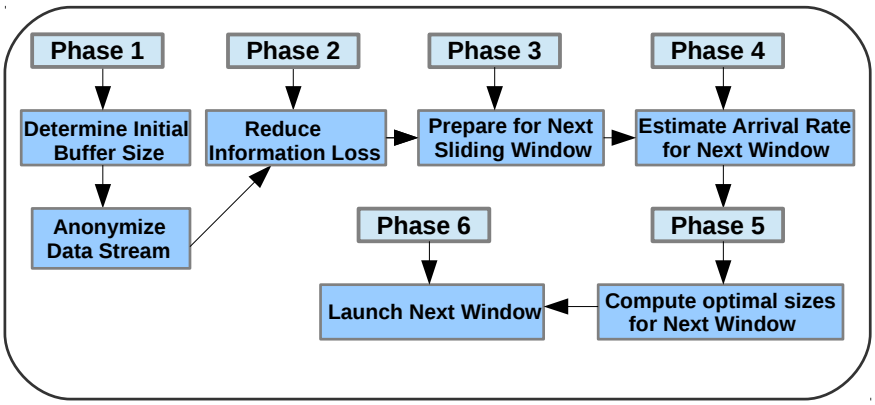

Fig. 7: Phases of Adaptive Buffer Re-sizing Scheme

where T_E is the time to expiry of a record , T_S is the time for which the record was stored in sw_i, and T_A is the time required to anonymize the data in sw_i.

Starting with the unanonymizable record, R_i, that has the lowest T_E and whose value falls within the acceptable bound, $[t_l, t_u]$, we check for other unanonymizable records, n, that belong to the same data anonymization group as R_i. We then proceed to find the rate of arrival, λ, of data in that anonymization group. We compute the arrival rate of records required to anonymize R_i within time T_E as follows:

$$\lambda = \frac{\mathrm{n}}{\mathrm{sw}_1} \times \mathrm{T}_E \quad ...(2)$$

The expected arrival rate, λ, is used to determine the probability of arrival of at least the number of records needed to guarantee that delaying anonymizing the unanonymizable record, R_i, to the next sliding window sw_{i+1} will not adversely increase information loss. We next determine the minimum number of records, m, required to guarantee anonymization of R_i in the next sliding window, sw_{i+1}. When the decision is to include the R_i into the next sliding window sw_{i+1}, we need to then compute the optimal size for sw_{i+1} in order to minimize information loss from record expiry. We achieve this by finding the probability that m records will actually arrive in the data stream within time, T_E, in order to anonymize the unanonymizable record, R_i. We use equation 3 to compute the probability of having i = 0...m records arrive in the stream within T_E

$$f\left(\mathrm{sw}_{i+1}, \lambda\right) = \Pr\left(\mathrm{i} = 0 \ldots. \mathrm{n}\right) = \frac{\lambda^i e^{-\lambda}}{\mathrm{i!}} \quad ...(3)$$

where λ is the expected arrival rate, e is the base of the natural logarithm (i.e. $e = 2.71828$) and i is the number of records under observation.

Therefore the probability of having greater that m or more records arrive in the stream within time T_E is

$$1 - \sum_{i=0}^{m-1} pr \quad ...(4)$$

where pr is the probability outcome of equation 3.

If the result of equation 4 is greater than a preset probability threshold, δ, we set the size of the subsequent sliding window, sw_{i+1}, to the expiry time of the unanonymizable record under consideration. We then mark the unanonymizable record for inclusion in the subsequent sliding window along with other unanonymizable records that have their T_E within bounds for acceptable sliding window sizes $[t_l, t_u]$. In the event that the probability of all unanonymizable records is less than the preset probability threshold, we set the subsequent sliding window size to a random number within the time bound, $[t_l, t_u]$. More detailed explanation of the adaptive buffer scheme can be found in our previous work [12].

5 Experiments and Results

This section presents the implementation and results of the crime-reporting Application, CryApp, and the adaptive buffering scheme algorithm.

5.1 CryHelp Application Evaluation

In order to evaluate the usability of our mobile crime reporting application, CryHelp, we developed a questionnaire. The questionnaire was based on IBM CSUQ [24]. The advantage of the IBM CSUQ [24] is that it allows questionnaires to be divided into scores and specific categories. These categories are: System Overall, System Usefulness, Information Quality and Interface Quality. These categories allow evaluation of each individual component of the system to gauge which aspects perform well or poorly on average. These results directly address the issue of whether a mobile device can be used to effectively and securely send a crime report.

Figure 8 shows the result of each component of the system. From the figure, it can be seen that overall the system was well received with an overall system score of 77.06%, this suggests the users found the system very usable with a standard deviation of 0.05 for contributing scores System Use, Information Quality and Interface Quality. It is not surprising to find that the interface quality (78.33%), though marginally, is the most appreciated aspect of the system as the design process was centered on the users. These results bode very well for the feasibility of a mobile solution for crime reporting.

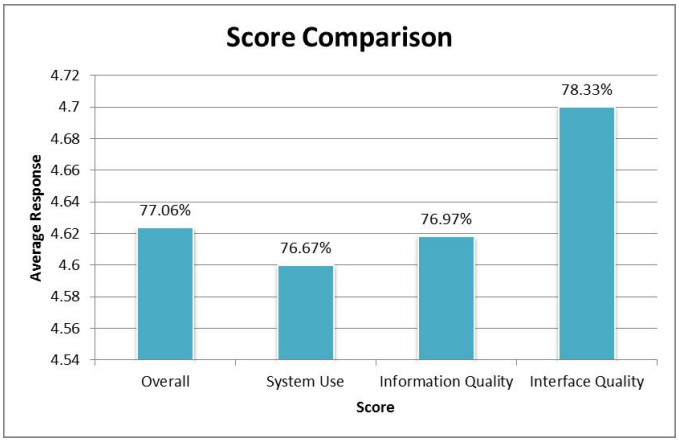

Fig. 8: The chart of the questionnaire score breakdown, with standard deviation 0.05

5.2 Experiments on Anonymization

Our feasibility study and experiment conducted on the prototype crime data collection application, CryHelp [5], informed the generation of more datasets for the second phase of experiment. The generation of more crime data was done using a random generator software [1] and pseudo-random algorithm based on a Gaussian distribution to populate the crime data-stream based on ground-truth provided by the users, UCT Campus Protection Service and the South African Police Service. Our data are in two sets, which contain 1000 and 10 000 records respectively, the first set contains 1000 records while the second set contains 10000 records, this is a reasonable bound for daily average crime report rates in South Africa [5].

Therefore, this section discusses the gains obtained using Poisson probability distribution to predict the time a sliding window should exist, while ensuring that records do not expire, the number of unanonymizable (or suppressed) records is minimal and privacy is maintained using k-anonymity, ℓ-diversity and t-closeness. The gains obtained are explained in the following sub-sections:

Recovered Unanonymizable Tuples: During anonymization there is usually a trade-off between the rate of IL, suppression and generalization. Usually if an equivalence class (cluster) is unable to satisfy the privacy requirement, such a class is either merged with another class or all its records are suppressed. A higher suppression rate implies that vital information is likely to be concealed from the recipient of the anonymized table, while merging of classes implies an increase in IL, which has the drawback of offering lower data utility. In order to curb this, Poisson probability distribution predicts the chances of such unanonymizable

[1] http://www.mockaroo.com

(suppressed) records undergoing anonymization in the next sliding window in a manner that preserves privacy and maximizes data utility with the goal of minimizing delay or expiration of records.

Figures 9 and 10 show the rate at which unanonymizable records were anonymized again, going by the predictions of Poisson probability distribution. It is evident from the figure that many unanonymizable records were recovered and allowed to go for anonymization again. It was also observed that the probability threshold influenced the number of unanonymizable records recovered. This leads to the conclusion that the higher the probability threshold, the lower the probability of unanonymizable records being given a chance for anonymization re-consideration in subsequent sliding window(s). The implication of this is that more records are likely not to be given the chance of another round of anonymization if higher threshold values are used. Another observation is that if a higher threshold value is used, then there are fewer changes or movements in records between sliding windows.

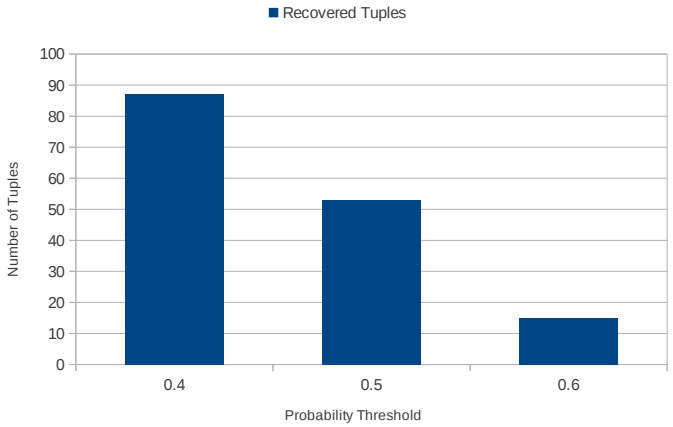

Fig. 9: Poisson Probability Threshold Versus Recovered Tuples for Dataset 1

Privacy Value/Level versus Recovered Unanonymizable Tuples: "Privacy level" simply means the degree of anonymity offered, while unanonymizable tuples are those tuples that belong to an equivalence class whose size is less than k. For the purpose of sliding windows that start with a small number of tuples, the minimum privacy level threshold was set as k=2 and the maximum at k= 15; the ℓ-diversity value, ℓ, was varied between values 3 and 5 and finally the t-closeness value, t, was alternated between values 0.1 and 0.15 for the two datasets.

As illustrated in Figures 11 and 12, it was observed that as the privacy value/level increases, the possible number of unanonymizable records that can be recovered using Poisson probability prediction is reduced. The main reason for this is that as the privacy level or degree increases, it is expected that the rate or possibility of achieving anonymization will become increasingly challenging. This definitely also influences the expectation of higher chances of anonymization rate

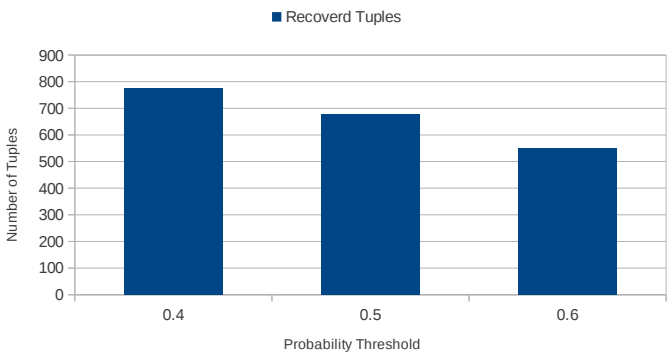

Fig. 10: Poisson Probability Threshold Versus Recovered Tuples for Dataset 2

for unanonymizable records. To understand the reason for the decline in the rate of recovered unanonymizable records better, let us assume the k privacy level is set to three and an equivalence class, EC_i, has two records; this implies that we are looking for at least one more record to make EC_i satisfy k-anonymity. In essence, using Poisson probability, the adaptive buffer resizing model attempts to predict the chance of at least one record in EC_i arrive within time, t, in the next sliding window. If k is set to four, this will mean the chance of at least two records arriving in the next sliding window. An implication of this is that the chances of having at least two records is more difficult or demanding compared to the chances of just one record. Thus, this explains why the model has a drop in recovered unanonymizable records as privacy level increases. Therefore, the conclusion is that the rate at which unanonymizable records in a current sliding window can be anonymized in a subsequent window is mainly dependent on the privacy value.

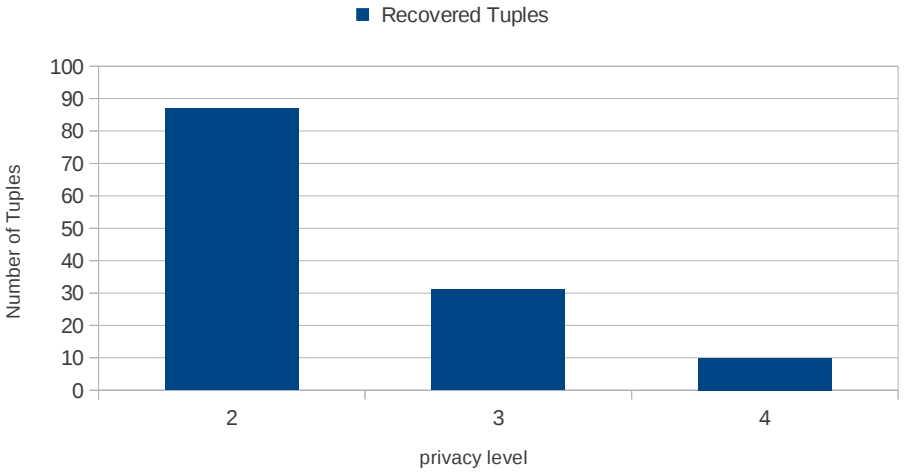

Fig. 11: Relationship Between Privacy Level and Recovered Tuples for Dataset1

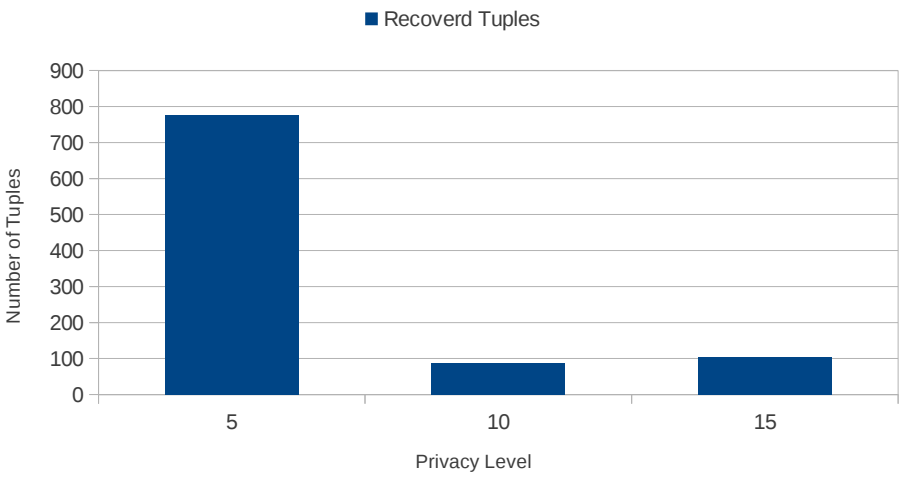

Fig. 12: Relationship Between Privacy Level and Recovered Tuples for Dataset2

5.3 Benchmarking: Poisson Solution Comparison with Non-Poisson Solution

As a baseline case, for evaluating our proposed adaptive buffering scheme we implemented the proactive-FAANST and passive-FAANST. These algorithms are a good comparison benchmark because they are the current state-of-the-art streaming data anonymization and reduce IL with minimum delay and expired tuples [3]. The proactive-FAANST decides if an unanonymizable record will expire if included in the next sliding window, while passive-FAANST searches for unanonymizable records that have expired. A major drawback of these two variants is that there is no way of deciding whether or not such unanonymizable records would be anonymizable during the next sliding window. This is necessary to avoid repeatedly cycling a tuple that has a low chance of anonymization in subsequent sliding window(s). Moreover, these algorithms do not consider the fact that the flow or speed of a data stream could change. These weaknesses of proactive-FAANST and passive-FAANST are what we attempt to address by using Poisson probability distribution to predict if such tuples would be anonymizable in subsequent sliding window(s) by taking into consideration the arrival rate of records, success rate of anonymization per sliding window, time a tuple can exist and rate of suppressed records.

Expired Tuples and Information Loss in Delay: A tuple expires when it remains in the system for longer than a pre-specified threshold called delay [3, 23]. In order to decide whether a tuple has exceeded its time-delay constraint, additional attributes such as arrival time, expected waiting time and entry time were included. As a heuristic, the choice of delay values, $t_l = 2000$ ms and $t_u = 5000$ ms, is guided by values of delay that are used in published experimentation results [3].

In general, our approach shows that there are fewer expired tuples when compared to passive-FAANST and proactive-FAANST solutions. This is because before our Poisson prediction transfers suppressed records to another sliding window, it checks for the possibility of their anonymization. In other solutions, there is no mechanism in place to check the likelihood of the anonymizability of a suppressed record before allowing it to go to the next sliding window/round. As a result, such tuples are sent to the next sliding window and have high a tendency to expire eventually. Our solution also shows that the lower a k-value, the higher the number of expired tuples. This is because the outcome of Poisson prediction is lower for higher k-values. As a result, there are fewer changes of sliding windows as the k-value increases and this means there is a lower possibility of expired tuples.

One of the main goals of our solution is to reduce IL in delay (i.e. to lower the number of expired tuples). Figure 13 depicts that our solution is successful in achieving its main goal, and the IL (delay) in our solution is lower than in passive and proactive solutions. In order to determine the total number of records that expired, a simple count function was used to retrieve all records that had remain in the buffer longer than the upper limit threshold, t_u. To determine the average expired records, we sum up the expired records in all the experiments and divide the result by the total number of experiments.

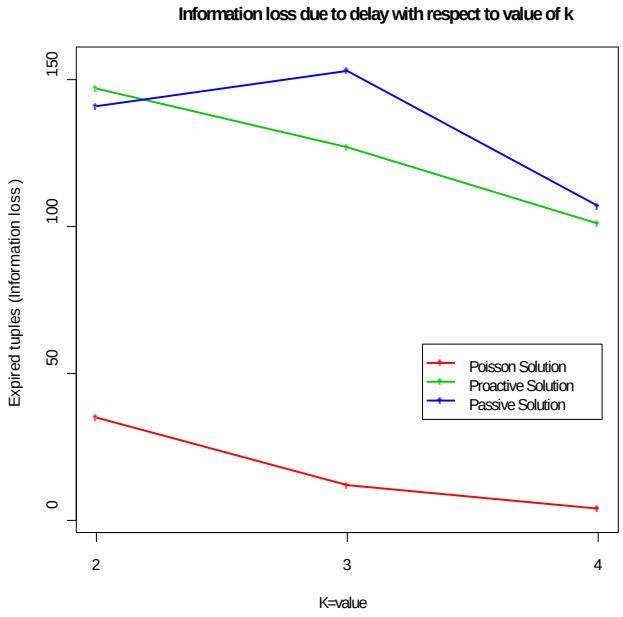

Fig. 13: Privacy Level Versus Expired Tuples for Poisson Solution, Passive-FAANST and Proactive-FAANST

Data Utility and Information Loss in Record: An important factor that is considered in anonymization is the degree of usability of anonymized data for data analysis or data mining tasks [29]. Therefore, we compared the degree of IL in records of our solution with that of Passive-FAANST and proactive-FAANST. Our result, as illustrated in Figure 14, shows that at the minimal level of privacy enforcement, the information loss of our solution is on par with the other two schemes, while at the maximal level our solution has better data utility.

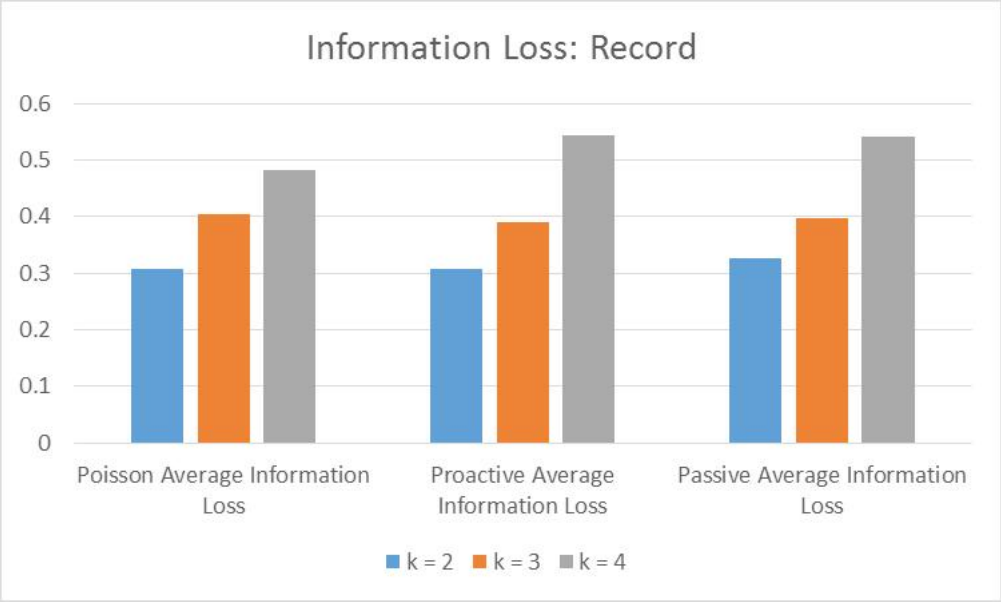

Fig. 14: Privacy Level Versus Information Loss for Poisson Solution, Passive-FAANST and Proactive-FAANST

6 Conclusions

We started this paper on the note that resource constrained environments lack data analytic expertise that can analyze and mine crime data in real-time. This anonymization process is important in order to provide intervention that can carry out this analysis in timely fashion. We adopted k-anonymity, ℓ-diversity and t-closeness as our anonymization scheme due to their simplicity, efficiency and applicability in real-life. However current literature on integration of these techniques to data stream has issues in terms of performance and privacy. The performance issue deals with information loss in terms of delay and running cost.

To address the challenge of ensuring that delay is optimal during anonymization process, we adaptively resized the buffer to handle intermittent flows of

crime reporting traffic optimally by using Poisson Distribution. Results from our prototype implementation demonstrate that in addition to ensuring privacy of the data, our proposed scheme outperforms other with an information loss rate of 1.95% in comparison to 12.7% on varying the privacy level of crime report data records.

References

1. Isafiade O. E. and Bagula A. B. (2013). "Citisafe: Adaptive spatial pattern knowledge using fp-growth algorithm for crime situation recognition." In Ubiquitous Intelligence and Computing, IEEE 10th International Conference on Autonomic and Trusted Computing (UIC/ATC) (pp. 551-556)."
2. Qiu L., Li Y. and X. Wu(2008). "Protecting business intelligence and customer privacy while outsourcing data mining tasks." "Knowledge and Information Systems, 17(1):99120".
3. Zakerzadeh, H. and Osborn, S. L. (2013). "Delay-sensitive approaches for anonymizing numerical streaming data." "International Journal of Information Security, 1-15, Springer."
4. Guo, K. and Zhang, Q. (2013) "Fast clustering-based anonymization approaches with time constraints for data streams." "Knowledge-Based Systems, Elsevier."
5. Sakpere, A.B., Kayem, A. V.D.M. and Ndlovu T. (2015) "A Usable and Secure Crime Reporting System for Technology Resource Constrained Contexts. In Proceedings of the 29th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA 2015), Gwangju, Korea March 24-27, 2015.
6. Li, S. (2006) "Poisson process with fuzzy rates. In Fuzzy Optimization and Decision Making, 9(3), pp. 289-305.
7. Sakpere A. B. and Kayem Anne V.D.M. (2014). "A state of the art review of data stream anonymisation schemes." "IGI Global, PA, USA."
8. Sakpere A. B. and Kayem Anne V.D.M. (2015). "Adaptive buffer resizing for efficient anonymization of streaming data with minimal information loss." In Proceedings of 1st International Conference on Information Systems Security and Privacy (ICISSP), pages 191 - 201.
9. Sakpere A.B. (2015). "User-Defined Privacy Preferences for k-Anonymization in Electronic Crime Reporting Systems for Developing Nations." "In proceedings of the 1st International Doctoral Symposium on Security and Privacy".
10. Mohammadian, E., Noferesti, M. and Jalili, R. (2014). "FAST: Fast Anonymization of Big Data Streams.", In Proceedings of the 2014 International Conference on Big Data Science and Computing (p. 23). ACM.
11. Lasley, J.R. and Palombo, B.J. (1995). "When crime reporting goes high-tech: An experimental test of computerized citizen response to crime.",Journal of Criminal Justice, 23(6), pp. 519-529.
12. Sakpere, A.B. and Kayem, Anne V.D.M. (2015). "Adaptive Buffer Resizing for Efficient Anonymization of Streaming Data with Minimal Information Loss." n Proceedings of the 1st International Conference on Information Systems Security and Privacy,ISBN 978-989-758-081-9, pages 191-201. DOI: 10.5220/0005288901910201.
13. Dwork, C., (2006) "Differential privacy." In Automata, languages and programming (pp. 1-12)., Springer Berlin Heidelberg.

14. Sakpere A.B., and Kayem Anne V.D.M. (2014). "A state of the art review of data stream anonymisation schemes." Information Security in Diverse Computing Environments, IGI Global, PA, USA.
15. Jensen, K. L., Iipito, H. N., Onwordi, M. U. and Mukumbira, S. (2012). "Toward an mPolicing solution for Namibia: leveraging emerging mobile platforms and crime mapping." In Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (pp. 196-205). ACM.
16. J. Domingo-Ferrer and J. Soria-Comas (2015). "From t-closeness to differential privacy and vice versa in data anonymization." *Knowledge-Based Systems, 74:151158, 2015.*
17. J. Soria-Comas and J. Domingo-Ferrer (2013). "Differential privacy via t-closeness in data publishing." *Proceedings of the 11th Annual Conference on Privacy, Security and Trust (PST), pages 2735.IEEE, 2013.*
18. Sweeney, L. (2002). "k-anonymity: A model for protecting privacy." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10 (05), 557-570.
19. Sweeney, L. (2002). "Achieving k-anonymity privacy protection using generalization and suppression." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05), 571-588.
20. Sweeney, L. (2001). "Computational Disclosure Control: A Primer on Data Privacy Protection." Thesis (PhD), Massachusetts Institute of Technology, Cambridge, MA, 2001. http://www.swiss.ai.mit.edu/6805/articles/ privacy/sweeney-thesis-draft.pdf
21. Iyengar, V. S. (2002). "Transforming data to satisfy privacy constraints." In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 279-288). ACM.
22. Cao, J., Carminati, B., Ferrari, E. and Tan, K. L. (2011) "Castle: Continuously anonymizing data streams." Dependable and Secure Computing, IEEE Transactions on, 8(3), 337-352.
23. Mohammadian, E., Noferesti, M. and Jalili, R. (2014) "Fast: fast anonymization of big data streams." In Proceedings of the 2014 International Conference on Big Data Science and Computing, page 23.ACM, 2014.
24. Lewis, J.R. (1995) "IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use" International Journal of Human-Computer Interaction, 7(1), pp.57-78.
25. Burke, M.J. (2013) "Enabling anonymous crime reporting on mobile phones in the developing world." Masters Dissertation, University of Cape Town.
26. Wang, W., Li, J., Ai, C. and Li, Y. (2007) "Privacy protection on sliding window of data streams." In Collaborative Computing: Networking, Applications and Worksharing, 2007. IEEE.
27. Li, J., Ooi, B. C. and Wang, W. (2008) "Anonymizing streaming data for privacy protection." In Data Engineering, 2008. IEEE 24th International Conference on (pp. 1367-1369). IEEE.
28. Zhang, J., Yang, J., Zhang, J. and Yuan, Y. (2010). "KIDS: K-anonymization data stream base on sliding window."In Future Computer and Communication (ICFCC), 2010 2nd International Conference on (Vol. 2, pp. V2-311). IEEE.
29. Li, T. and N. Li (2009). "On the tradeoff between privacy and utility in data publishing." Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,pages 517-526,ACM.
30. Machanavajjhala, A., Kifer D. and Johannes G. (2007)."l-diversity: Privacy beyond k-anonymity." "ACM Transactions on Knowledge Discovery from Data (TKDD), 1(1), 3."

31. Li, N., Li T. and Venkatasubramanian S. (2007).“t-closeness: Privacy beyond k-anonymity and l-diversity.” ”International Conference on Data Engineering (ICDE), (3), 106115.”
32. J. Soria-Comas and J. Domingo-Ferrer (2013).“Differential privacy via t-closeness in data publishing.” ”Proceedings of the 11th Annual Conference on Privacy, Security and Trust (PST), IEEE.”
33. J. Soria-Comas and J. Domingo-Ferrer (2015).“Differential privacy via t-closeness in data publishing.” ”Knowledge-Based Systems), Elsevier.”