Supporting Streaming Data Anonymization with Expressions of User Privacy Preferences

Aderonke Busayo Sakpere(✉) and Anne V.D.M. Kayem

Department of Computer Science, University of Cape Town, Cape Town, South Africa
olfade001@myuct.ac.za, akayem@cs.uct.ac.za

Abstract. Mining crime reports in real-time is useful in improving the response time of law enforcement authorities in addressing crime. However, limitations on computational processing power and in-house mining expertise make this challenging, particularly so for law enforcement agencies in technology constrained environments. Outsourcing crime data mining offers a cost-effective alternative strategy. Yet outsourcing crime data raises the issue of user privacy. Therefore encouraging user participation in crime reporting schemes is conditional on providing strong guarantees of personal data protection. Cryptographic approaches make for time consuming query result generation, so the preferred approach is to anonymize the data. Mining real-time crime data as opposed to static data facilitates fast intervention. To achieve this goal, Sakpere and Kayem presented a preliminary solution based on the notion of buffering. Buffering improves on information loss significantly in comparison with previous solutions. In this paper, we extend the Sakpere and Kayem result to support user privacy expressions. We achieve this by integrating a three-tiered user-defined privacy preference model in data stream process. The three-tiered model offers a simple and generic approach to classifying the data without impacting negatively on information loss. Results from our proof-of-concept implementation indicate that incorporating user privacy preferences reduces the rate of information loss due to misclassification.

Keywords: Data anonymity · Streaming data · Crime reporting · Information loss

1 Introduction

Law enforcement agencies in resource constrained environments.[1] generally lack the "on-the-ground" expertise and resources required to mine crime big data streams. A cost-effective solution is to transfer the task of mining the crime data streams to a third party data miner/analyst. Mining crime reports in real-time as opposed to in static form can be helpful in providing fast interventions in addressing crime. A further advantage is predictions of future crime or disaster occurrences to track the criminals or suspects in a relatively short period.

[1] These are environments that are characterized by low computational and processing resources. Examples emerge in disaster scenarios and remote areas.

O. Camp et al. (Eds.): ICISSP 2015, CCIS 576, pp. 122–136, 2015.
DOI: 10.1007/978-3-319-27668-7_8

1.1 Motivation and Problem Statement

Applying k-anonymity on data streams faces three drawbacks in relation to minimizing delay and incorporating user's privacy preferences during anonymization.

Firstly, existing data stream anonymization algorithms do not take user privacy preferences into consideration. K-anonymity uses the same privacy level (i.e. k-value) for all individuals in the data set. The use of the same privacy level for all users is unrealistic in real-life because individuals tend to have varying privacy protection requirements [12,13]. Furthermore, the use of the same privacy preference level for all users implies that individual privacy needs are misrepresented.

Secondly, existing data stream anonymization algorithms apply a delay constraint on each tuple in the buffered stream [5,6,14]. Buffering incurs high information loss levels in terms of delay in cases of intermittent streaming data flows. This is because anonymization is typically triggered on the basis of the number of records (tuples) in the buffer as opposed to the time-sensitivity of the data.

Thirdly, anonymization of intermittent or slow data streams results in high information loss or suppression as is the case in the crime reporting scenario. However, the focus of many of the existing data stream anonymization algorithms is on fast data streams and as a result overlooking the rate at which data arrives in the stream when determining an optimal buffer size. The buffer size and rate of arrival of the streaming crime data affects information loss with respect to delay and the levels of privacy offered by the anonymization scheme.

1.2 Contribution

In this paper, we offer two contributions. Firstly, we propose an approach to minimizing delay while a record waits to be anonymize in the buffer. Secondly, we augment our streaming data anonymization scheme by supporting anonymization of data stream with user-defined privacy preferences.

In order to minimize delay, we model our buffer as a time-based tumbling sliding window that is constrained by delay as opposed to record count as it is the case in other solutions because of the time-sensitive nature of crime data. Afterwards, we develop a solution to adaptively re-adjust the size of sliding window based on an arrival rate of data that follows a Poisson process.

In order to ensure that privacy controls are enforced in a balanced way i.e. there is no excessive privacy control or insufficient privacy measure, we supported our adaptive buffer resizing scheme with three-tiered user-defined privacy preference (low, neutral (medium) and high) model. In order to see how this can be integrated in real-life, we carried out a survey in a campus setting in a technology resource constrained environment. We modeled and analyzed the data gathered from our survey using association rules in order to automatically deduce features that determine a user's privacy preference in an automated way. Lastly, we came up with an appropriate k-value to be used for a user's anonymization when there is insufficient or excessive privacy enforcement in comparison to the user's need.

Firstly, we carried out a real-life survey in our do main of interest (crime data) in order to determine if the usage and integration of three-tiered user-privacy into k-anonymity is practicable in real-life. Secondly, we came up with association rules in order to determine factors that influence users privacy preference. As a further step, we integrated the association rules into the k-anonymity technique in order to further aid determination of an appropriate k-value to be used for anonymization process.

1.3 Outline

The rest of the paper is structured as follows. In Sect. 2, we present related work highlighting the weaknesses of existing data stream anonymization and user-defined privacy preferences. Section 3, presents a review of our previous work that addressed the buffering problem and we further improve on how to incorporate user privacy preferences. In Sect. 4, we present results from our proof-of-concept implementation and conclude in Sect. 5.

2 Related Work

Sakpere and Kayem [8], presented an adaptive buffer resizing scheme to minimize information loss (delay) during streaming data anonymization was proposed. The buffer is modeled as a time-based sliding window whose size is dynamically re-adjusted based on an arrival rate of data that follows a Poisson process. Information loss in terms of numbers of data records is minimized by selectively suppress data records from a sliding window. Depending on the time sensitivity, such suppressed records are either included in a subsequent sliding window or inserted into a reusable anonymity cluster. Results from our prototype implementation demonstrate that our proposed scheme is privacy preserving and incurs an information loss in terms of delayed records of 1.95 % in comparison to other schemes that incur a 12.7 % rate. However, Sakpere and Kayem's, as well as previous solutions do not consider user privacy preferences during anonymization of streaming data which has the drawback of offering insufficient or excessive protection with respect to user needs.

To address the issue of incorporating user's personal privacy preferences into k-anonymity, Aggrawal and Yu [11], Gedik and Liu [13] allow a user to select an integer, i, (where $1 \leqslant i \leqslant n$) to indicate his/her preferred k-value. A drawback of this is that it might be difficult for users to set a realistic k-value in real-life especially in a Crime Reporting System where users might be under shock. Also, setting a realistic k-value implies that users must understand how k-anonymity works.

An equally novel approach in achieving personalized anonymization using the concept of k-anonymity is the work of Xiao and Yufei [12]. In their work, a user is required to specify the degree of privacy protection for his/her sensitive values. Their solution assumes that each sensitive attribute has a classification tree and each record owner specifies a guarding node in the tree. Guarding nodes depend on user personal privacy preferences and indicate how users want their sensitive values to be represented. A major drawback of their approach is that a guarding node requires that a hierarchy-tree be defined on sensitive attribute. Another major drawback is that in real-life, it is unclear how individual record owners would set their guarding node [11].

We therefore note that the issue of incorporating user privacy preferences to cope with anonymization of streaming data in a manner that is usable in real-life is yet to be studied. This study is necessary in order to generate reliable anonymized reported crime data that meets users need.

3 User-Defined Privacy Preference in Adaptive Buffer Re-sizing Scheme

In this section we present the integration of user-defined privacy preferences into the adaptive buffer resizing scheme. However, we will like to briefly recap how we adaptively reduce buffer size in our previous work [8].

3.1 Adaptive Buffer Resizing Scheme

As mentioned in our introduction section, the buffer size and rate of arrival of the streaming crime data impacts on the accuracy in terms of minimizing information loss and reliability in terms of privacy enforcement of the anonymization scheme. In order to minimize the percentage of information loss (in terms of delay) during the streaming data anonymization process, we use a time-based tumbling sliding window to adjust the size of the buffer dynamically with respect to the arrival rate of the data.

As illustrated in Fig. 1, a “sliding window” or “buffer”, sw_i, is a subset of the data stream, DS where DS = $\{\mathrm{sw}_1, \mathrm{sw}_2, \mathrm{sw}_3,..., \mathrm{sw}_m\}$ implies that the data stream consists of m sliding windows. The sliding windows obey a total ordering such that for $i < j$, sw_j precedes sw_i. 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.

3.2 Streaming Data as a Poisson Process

We model the flow rate of the data stream as a Poisson process because the arrival rate of data in the stream can be viewed as a series of events occurring within a fixed time interval and with a known average rate that is independent of the time of occurrence of the last event [10]. The Poisson distribution is a discrete probability distribution that measures the probability of having a given number of records occurring in the stream within a fixed time and/or space interval provided that these records arrive are each with a known average rate and are each independent of the last event occurrence. So, this occurrence is a good distribution for estimating streaming data reporting rates. In the Poisson distribution, only one parameter needs to be known, namely rate at which the

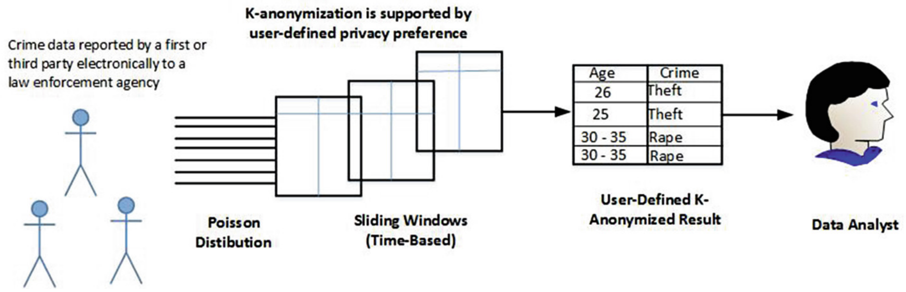

K-anonymization is supported by user-defined privacy preference
Crime data reported by a first or third party electronically to a law enforcement agency
Age
Crime
26
Theft
25
Theft
30 - 35
Rape
30 - 35
Rape
Poisson Distibution
Sliding Windows (Time-Based)
User-Defined K-Anonymized Result
Data Analyst

Fig. 1. Overview of buffer resizing process.

events occur which in our scenario would be the rate at which crime reporting occurs.

Considering that the data stream, DS follows a Poisson process with an arrival rate of events $\lambda > 0$, then we can say that for a sliding window sw_i the probability mass function is given by:

$$f\left(\mathrm{sw}_i, \lambda\right) = \Pr\left(\mathrm{DS} = \mathrm{sw}_i\right) = \frac{\lambda^{\mathrm{sw}_i} e^{-\lambda}}{\mathrm{sw}_i!} \tag{1}$$

where e is the base of the natural logarithm (i.e. $e = 2.71828$) and sw_i is the size of the i_{th} sliding window under evaluation for anonymization.

3.3 Buffer Sizing

Our adaptive buffer re-sizing scheme relies on a streaming data flow rate that obeys the Poisson distribution model described above and works as follows. First we begin by setting the size of the buffer to an initial preset threshold value. Given the time-sensitivity of the data, we use a time value, T that is bounded between t_l and t_u. When the time threshold, T, is attained, the k-anonymization algorithm is applied to the data that was collected during this period. All records that are not anonymizable in the current data set are suppressed and based on the closeness of the suppressed records to their expiry deadlines, we either include the records in the next sliding window or incorporate the records into any of existing reusable clusters that has the smallest distance from the record(s).

In order to decide whether to include a suppressed record in the next sliding window sw_{i+1}, we first compute the average remaining time that the suppressed records have left before they expire. We denote this time as T_E and obtain a value for T_E by subtracting the average time for which the records have been stored in the buffer from the size of the current sliding window sw_i. Next, in order to determine the minimum number of records needed to minimize information loss from expired records in the data stream that will compose the next sliding window sw_{i+1}, we first compute an estimated time-bound for $\mathrm{sw}_{i+1}!$ by adding T_E to the time T_A that was used to anonymize the data in sw_i. We then compute the minimum number of records required adding T_E to T_A and incorporating the value into Eq. 1 to obtain an expected data stream flow rate λ_{i+1} as follows:

$$f\left(\mathrm{sw}_{i+1}, \lambda_{i+1}\right) = \Pr\left(\mathrm{DS} = \mathrm{sw}_{i+1}\right) = \frac{\lambda^{\mathrm{sw}_{i+1}} e^{-\lambda}}{\mathrm{sw}_{i+1}!} \tag{2}$$

where $\mathrm{sw}_{i+1} = T_E + T_A$. From the values of λ_{i+1} and $(T_E + T_A)$ respectively we can easily obtain a value for the minimum number of records n needed for sw_{i+1}.

We must now decide what to do with the suppressed records. As mentioned before, we can either include a suppressed record in the new sliding window sw_{i+1} or based on its distance include the record in a reusable cluster of existing anoymized data. In the first case, for each suppressed record, we compare the remaining time T_R that the record has before it expires. We do this as follows, if $T_R \geq T_E + T_A$ then the affected suppressed record gets included in sw_{i+1}. Otherwise, if $T_R < T_E + T_A$ then the record concerned gets incorporated into an appropriate reusable cluster.

Finally, in order to decide which reusable cluster to include a suppressed record in, we choose the reusable cluster that has the least distance from the suppressed record. When only one reusable cluster exists, we simply add the suppressed value to the cluster. If we have several clusters to select from the one with the lowest distance is chosen. Lastly, when anonymization is not possible and there is no existing reusable cluster into which the suppressed records can be included we create a new reusable cluster.

From the discussion in this section, our framework for the Buffer Re-sizing anonymization of data streams can be summarized as follows [8]:

Algorithm 1. SWET (i,K).

```
for each sliding window $sw_i$, $i$:1 ...$m$ do
    if $((sw_i == 1) || (SuppRec == \phi))$ then
        $sw_{iExistTime} \leftarrow T$
    else
        $sw_{iExistTime} \leftarrow RSWET(T_R, T_A, i, SuppRec)$
    end if
    $T_A \leftarrow$ Anonymization Processing Time
    $SuppRec \leftarrow$ Suppressed Records
    $T_R \leftarrow$ Remaining Time of Suppressed Records
    Update Reusable Cluster (RC)
end for
```

3.4 Integration of User-Defined Privacy Preference into Adaptive Buffer-Resizing

As earlier stated in our introduction that the use of k-anonymity for data streaming anonymization uses a generic approach to enforce privacy preservation for all

Algorithm 2 . RSWET($T_R, T_A, i, SuppRec$).

```
Sort: Sort $T_R$ in ascending order and group by unanonymizable cluster
for j:1 ...|$SuppRec$| do
    if $T_{R_j}$ - $T_A$ < $T_l$ then
        Anonymize $SuppRec_j$ using RC
        Delete $SuppRec_j$
    else
        Calculate arrival rate, $\lambda$, of $SuppRec_j$ in the sliding window, $sw_i$
        Find the Probability, P, of successful anonymization in $sw_i$
    end if
    if $P$ or $\lambda > \delta$ then
        $ExistTime_i \leftarrow T_{R_j} - T_A$
        Add $SuppRec$ to $sw_i$
        break
    else
        anonymize $SuppRec_j$ using RC
        delete $SuppRec_j$ from $SuppRec$
    end if
end for
if $P$ or $\lambda$ for all suppressed records < $\delta$ then
    $ExistTime_i \leftarrow T$
end if
return ExistTime$_i$
```

users without catering for their concrete needs. The consequence of this is that insufficient protection might be provided to a subset of people while excessive privacy control is provided to another subset. In order to ensure a user's data protection meets her need we attempted to support data stream anonymization with user's privacy preference.

Existing literature [7] on personalized-privacy shows that in real-life, users view their privacy preferences as either High, Intermediate and Low. A high privacy level indicates an extreme privacy consciousness whereas a low privacy level depicts a lower privacy consciousness. Therefore, neutral privacy level is intermediate. We also carried out a real-life survey on personalized-privacy and result shows that in real-life, users find it easy to recognize their privacy setting if given three-preferences. The user study approach was used for the survey through questionnaire and interview. All the collected survey data comprised of 26 subjects and eight categorical variables: Sex, Age group, Present education level/Occupation, Highest education qualification (HEQ), Victim of crime, Crime experienced, Preferred privacy level (PPL) and Reason for choice of privacy (RCP). Figure 2 shows a summary of the data obtained during our analysis.

We integrate our three-tier level privacy preference into k-anonymity in the adaptive buffer resizing scheme by starting data streaming anonymization with k-anonymity principle/scheme. K-anonymization schemes classify records into different buckets such that each record in a bucket is indistinguishable from at least *k-1* records [2].

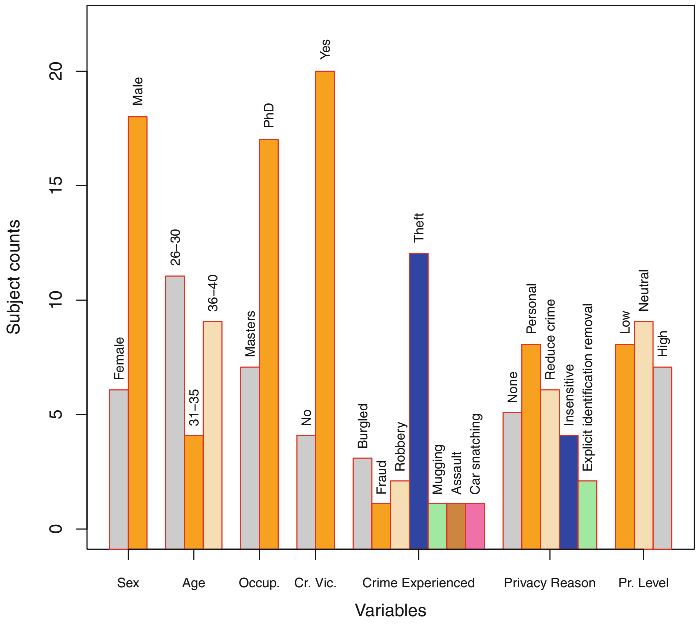


Fig. 2. Histogram illustrating the distribution of subjects over the different categories of variables surveyed in the primary study of privacy level preference.

Our basis for starting anonymization with the principles of k-anonymity schemes is because according to Sweeney, an optimal privacy is reached if a subset contains at least k data set [1,2]. However, it is still possible that a subset can contain records greater than or lesser than k. If a subset is lesser than k, then records in such subsets are not sufficiently protected and if a subset contains greater than k records then excessive privacy control is enforced. Therefore, the integration of user-defined privacy preference into adaptive buffer resizing during k-anonymization is to ensure that no subset contains lesser than or greater than k-records.

To address the shortcomings of a generic privacy enforcement in k-anonymity, we define three different levels of protection for users and incorporate them into adaptive buffer resizing scheme.

We begin by starting anonymization of all tuples using the principles of k-anonymity. Next, we search for subsets that contain lesser than k-records and attempt to anonymize records in the subset using user-defined privacy preference. For example, if a user's privacy preference is medium and his/her record is in a subset that contains lesser than k records, then we attempt to use a mid k-value to carry out anonymization for such records. Finally, we Search for subsets that contain greater than k-records and attempt to anonymize records in the subset using user-defined privacy preference. For example, if a user's privacy preference is low and his/her record is in a subset that contains greater than k records, then we attempt to use a low k-value (obtained by simply removing every explicit identifier) to carry out anonymization for such records.

4 Implementation and Results

We divide this section into two. The first section focuses results on minimizing delay before anonymization while the second section addresses privacy preservation guided by user requirements. Both experiments were performed on an Intel Core i5-3210 2.50Ghz computer with 4GB of physical memory. The operating system used was Ubuntu 12.10.

4.1 Buffering

The proposed adaptive buffering framework was implemented by extending the existing CSE 467[2] k-anonymization implementation. To depict streaming data, we used the file input stream functions in java, that reads data in real-time from an external source/excel file into sliding window. MySQL database storage was used to depict our sliding window. We assume that only a single data is read from the external file into the buffer at each instant in time.

We synthetically generated a realistic crime data set that follows the structure of the Cry-Help App using a random generator software[3]. The CryHelp App is a simple crime reporting application developed for mobile phones running the Android Operating System. Figure 3 shows some screenshots from the CryHelp App. The app was developed in conjunction with the University of Cape Town Campus Protection Service (CPS). The app enables users to send crime reports[4].

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 that reduce information loss with minimum delay [14]. 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 unanonymizable records would be anonymizable during the next sliding window. In our experiment, the proactive-FAANST and passive-FAANST solutions also use the reusable cluster concept as well but do not allow for overlapping of sliding windows, which our implementation does, nor do they model the flow rate of reported crime data as a Poisson process.

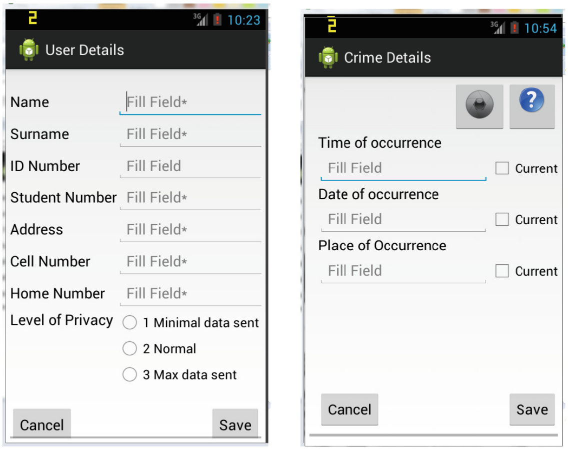
Our experiments were conducted to measure the following:

1. Information loss in terms of delay

[2] http://code.google.com/p/cse467phase3/source%20/browse/trunk/src/Samarati.java?r=64.

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

[4] Further details about the app can be found in http://cryhelp.cs.uct.ac.za/download.



User Details

Name	Fill Field*
Surname	Fill Field*
ID Number	Fill Field
Student Number	Fill Field*
Address	Fill Field*
Cell Number	Fill Field*
Home Number	Fill Field*
Level of Privacy	○ 1 Minimal data sent ○ 2 Normal ○ 3 Max data sent

Cancel Save

Crime Details

Time of occurrence

Fill Field □ Current

Date of occurrence

Fill Field □ Current

Place of Occurrence

Fill Field □ Current

Cancel Save

Fig. 3. Screenshots from CryHelp app.

2. Information gain obtained from modelling the flow rate of the data as a Poisson process

Effect of Privacy Levels (k-anonymity value) on Information Loss (delay). As a heuristic, the choice of $k >= 2$ and $t_l = 2000\,\text{ms}$ and $t_u = 5000\,\text{ms}$, is guided by values that are used in published experimentation results [14]. Figure 4 shows the effect of k-anonymity level on information loss with respect to delay (the number of expired records).

The main goal of our adaptive buffering solution is to reduce information loss (delay) (i.e. to lower the number of expired tuples). Figure 4 depicts that

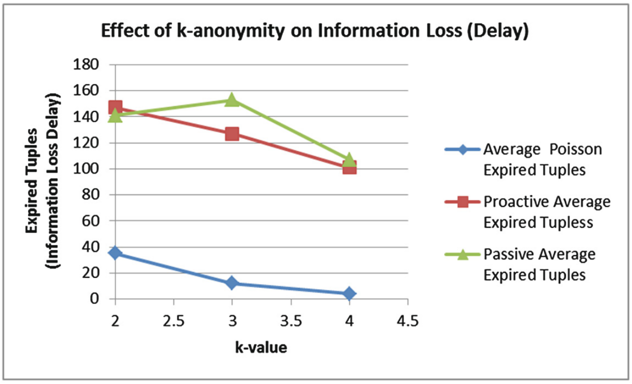


Fig. 4. Performance comparison: information loss with respect to privacy levels (expressed by the K-value).

our solution is successful in achieving its main goal and the information loss (delay) in our solution is lower than passive and proactive solutions. In order to determine the total number of records that expired, a simple query was executed to retrieve all records that have stayed in the buffer longer than the upper limit threshold, t_u. To get the average expired records, we sum up the expired records in all the experiments and divide by the total number of experiments.

Figure 4 shows the effect of the sliding window and k-anonymity level on information loss with respect to delay. In general, our approach shows that there are fewer expired tuples when compared to passive and proactive solutions. This is because before the Poisson probability prediction model transfers suppressed records to another sliding window, it checks for possibility of anonymization of the records. For 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 may be sent to other rounds/sliding windows and eventually expire.

The main goal of our solution is to reduce information loss (delay)(i.e. to lower the number of expired tuples). Figure 4 depicts that our solution is successful in achieving its main goal and the information loss (delay) in our solution is lower than passive and proactive solutions. 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 possibility of its 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 get sent to other rounds/sliding windows and has high tendency to eventually expire.

Effect of Poisson Probability Value δ on Information Loss (Record). To calculate information loss with respect to records i.e. deviation of anonymized data from its initial form, we used the formula in Eq. 3 as it is in [3]. We adopted this metric because it is a benchmark in many data stream anonymization schemes [5,6,14].

$$\text{InfoLoss} = \frac{\mathrm{M}_P - 1}{\text{M - 1}} \tag{3}$$

M_p is number of leaf nodes in the subtree at node P and M is the total number of leaf nodes in the generalization tree. We calculate the information loss of a Sliding Window, SW_i = {R$_1$, R$_2$, R$_3$,..., R$_n$} as follows:

$$\frac{1}{\mathrm{n}}\sum_{i=1}^{n}\text{InfoLoss}(R_i) \tag{4}$$

The total information loss of a data stream is simply calculated by averaging the information loss of all sliding windows in it.

Figure 5 shows the effect of Poisson probability value, δ, on information loss (record). The figure shows that a higher value of δ results in higher information loss. This is because if the probability that a suppressed record will be

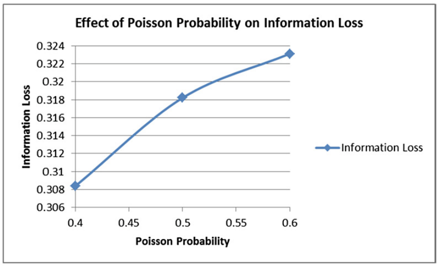


Fig. 5. Effect of poisson probability on information loss (record).

unanonymizable falls below δ such a record will be suppressed if there is no suitable reusable cluster. Suppressed records incur the highest rate of information loss. As a result, the higher the value of δ, the higher the chance of suppressed records and subsequently higher information loss. The result is based on the k-anonymity value of 2 and maximum suppression of 5 per sliding window. Time-based sliding window varies from 2000 ms to 5000 ms.

4.2 User-Defined Privacy Preference

We integrated our three-tier user-defined privacy preference into k-anonymity in data streaming anonymization by starting with a general k-value into a more specific or personalized k-value. As a heuristic, the choice of a general k value is guided by values used in published experimentation results [14].

As earlier stated, k-anonymity uses a generic approach to enforce privacy preservation for all users without catering for their concrete needs. The outcome of this is that insufficient protection might be provided to a subset of people, while excessive privacy control is provided to another subset. Therefore, our experiment is geared towards ensuring that there is a balanced protection by taking user's privacy preference into consideration.

Reduction of Excessive Privacy Control. Results from experiment as shown in Fig. 6, shows that integration of our approach to k-anonymity in comparison to other approaches ensure that excessive privacy control is reduced while at the same time guiding against insufficient protection. Our three-personalised approach has 16.15 % rate of excessive privacy control while Gedik- Personalised model and non-personalised has 63.08 % and 23.08 % rate of excessive privacy control respectively.

The reason our approach performed better than Gedik and non-personalized privacy is because we first used a general k-value and then attempt to personalize when there is excessive privacy control in comparison to user's preference. The result of our three-tier personalized result also shows that the higher the k-value, the higher the rate of excessive privacy control. This is because as k-value increases anonymization and privacy quality increases too. Hence more records

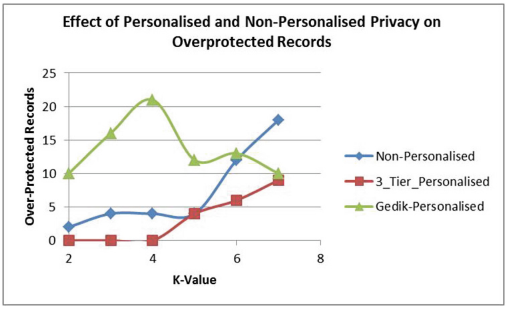


Fig. 6. Effect of personalised and non-personalised privacy on excessive privacy control.

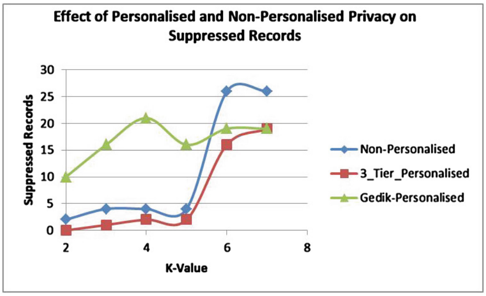


Fig. 7. Impact of the personalized and non-personalized privacy scheme on minimizing number of suppressed records.

have the chance of been suppressed which in-turn leads to excessive privacy control.

Record Suppression. One of the goals of a good anonymization scheme is to ensure that information loss is minimal. Records suppression usually leads to a high information loss. The use of personalized privacy scheme minimizes total number of suppressed records and as a result reduces information loss while the use of non-personalized privacy scheme leads to high number of suppressed records.

Figure 7 shows the effect of personalized and non-personalized privacy on suppressed records. Our result shows that our three-tier personalized privacy has lower rate of suppressed records which is 26 % when compared to Gedik-personalized privacy that has 77.7 % and non-personalized Privacy that has 51 %. This is because our three-tiered personalized privacy model considers suppressed records and attempts to reduce information loss by using user privacy preferences. In order to also measure the effect of k-values on suppressed records, we set k-value to values between 2 and 7. The result also shows that the higher the k-value, the higher the number of suppressed records which is to be expected since high k-values imply a high anonymization degree.

5 Conclusions

We began this paper with an overview of the problem scenario which emerges in developing nations where the lack of data analytics expertise within a law enforcement agency makes the need to have a third party data analytics provider intervene to aid in fast crime report analysis. In addition, we highlighted the fact that the growing need to make the processed information available to field officers requires a mechanism for capturing crime reports in real-time and transferring these reports to the third-party service provider. While solutions in the literature that are hinged on cryptography have been shown to be successful in protecting data in outsourced scenarios from unauthorized access including that of “honest-but-curious” service providers, we noted that querying encrypted streaming data is a time consuming process and that k-anonymization technique is a more practical approach to data privacy preservation in this case.

Anonymizing streaming data in a crime reporting context however, can have strong real-time requirements and therefore information loss can lead to faulty or misguided conclusions on the part of the data analytics service provider. Therefore, streaming data anonymization algorithms (schemes) need to be supported by good buffering mechanisms. Our proposed approach uses the concept of modelling the flow rate of reported crime streaming data as a Poisson process that guides the sizing of a time-based sliding window buffer. The data collected in the buffer is subjected to k-anonymization to ensure privacy of the data. 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. However, the generic paradigm approach to privacy enforcement in the k-anonymity model needs to be refined in order to cater for individual’s need. Therefore, we refined our model to integrate users privacy preference into the k-anonymity model while attempting to reduce delays incurred as a result of buffering. The results show that the use of personalized privacy preferences ensure that protection is enforced in a balanced way by a 23.08 % information loss rate in comparison to non-personalized techniques that have an average balanced protection and incur loss rates of 63.08 %.

References

1. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **10**(05), 557–570 (2002)
2. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **10**(05), 571–588 (2002)

3. Iyengar, V.S.: 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 (2002)
4. Kabir, M.E., Wang, H., Bertino, E.: Efficient systematic clustering method for k-anonymization. Acta Informatica **48**(1), 51–66 (2011)
5. Cao, J., Carminati, B., Ferrari, E., Tan, K.L.: Castle: continuously anonymizing data streams. IEEE Trans. Dependable Secure Comput. **8**(3), 337–352 (2011)
6. Guo, K., Zhang, Q.: Fast clustering-based anonymization approaches with time constraints for data streams. Knowledge-Based Systems, Elsevier (2013, in press)
7. Sakpere, A.B.: User-defined privacy preferences for k-anonymization in electronic crime reporting systems for developing nations. In: Doctoral Consortium, pp. 13–18 (2015). doi:10.5220/0005364700130018
8. Sakpere, A.B., Anne, V.D.M.K., Marchetti-Mercer, M.C.: Adaptive buffer resizing for efficient anonymization of streaming data with minimal information loss. In: Proceedings of the 1st International Conference on Information Systems Security and Privacy, pp. 191–201 (2015). doi:10.5220/0005288901910201
9. Stone, C.: Crime, justice, and growth in South Africa: toward a plausible contribution from criminal justice to economic growth. John F. Kennedy School of Government Working Paper No. RWP06-038(2006)
10. Li, S.: Fuzzy optimization and decision making. Poisson Process with Fuzzy Rates, pp. 289–305. Kluwer Academic Publishers, Hingham (2010)
11. Aggarwal, C.C., Yu, P.S. (eds.): TA General Survey of Privacy-preserving Data Mining Models and Algorithms. Springer, Heidelberg (2008)
12. Xiao, X., Tao, Y.: Personalized privacy preservation. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data. ACM (2006)
13. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: architecture and algorithms. IEEE Trans. Mob. Comput. **7**(1), 1–18 (2008)
14. Zakerzadeh, H., Osborn, S.L.: FAANST: Fast Anonymizing Algorithm for Numerical Streaming DaTa. In: Garcia-Alfaro, J., Navarro-Arribas, G., Cavalli, A., Leneutre, J. (eds.) DPM 2010 and SETOP 2010. LNCS, vol. 6514, pp. 36–50. Springer, Heidelberg (2011)