Efficient Approximation and Privacy Preservation Algorithms for real time online Evolving Data Streams

Rahul Patil (patil.rahul3068@gmail.com)
Dr. D. Y. Patil Institute of Technology
Pramod Patil
Dr. D. Y. Patil Institute of Technology

Research Article

Keywords: Approximation, data streaming, clustering, k-anonymization, l-diversity, privacy preservation.

Posted Date: October 5th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2112560/v1

Abstract

Mining real-time streaming data is a more difficult research challenge than mining static data due to the processing of continuous unstructured massive streams of data. As sensitive data is incorporated into the streaming data, the issue of privacy continues. In recent years, there has been significant progress in research on the anonymization of static data. For the anonymization of quasi-identifiers, two typical strategies are generalization and suppression. But the high dynamicity and potential infinite properties of the streaming data make it a challenging task. To end this, we propose a novel Efficient Approximation and Privacy Preservation Algorithms (EAPPA) framework in this paper to achieve efficient data pre-processing from the live streaming and its privacy preservation with minimum Information Loss (IL) and computational requirements. As the existing privacy preservation solutions for streaming data suffered from the challenges of redundant data, we first proposed the efficient technique of data approximation with data pre-processing. We design the Flajolet Martin (FM) algorithm for robust and efficient approximation of unique elements in the data stream with a data cleaning mechanism. We fed the periodically approximated and pre-processed streaming data to the anonymization algorithm. We propose novel k-anonymization and l-diversity privacy principles for data streams using adaptive clustering. The proposed approach scans a stream to detect and reuse clusters that fulfill the k-anonymity and l-diversity criteria for reducing anonymization time and IL. The experimental results reveal the efficiency of the EAPPA framework compared to state-of-art methods.

1. Introduction

Today, digital transformation is transforming the face of organizations all over the world, and it is increasingly necessary for firms to incorporate data into their operations than developing and investing in new technology. According to recent research, the global data sphere will raise from 33 Zettabytes (ZB) in 2018 to 175 Zettabytes (ZB) by 2025 [1]. Almost everyone nowadays interacts with data daily. Businesses are attempting to acquire this data to process it further to extract interesting patterns that can be utilized to boost profitability. Many data mining approaches have been proposed in the literature by various academics, most of which are relevant to batch processing [2]. These strategies may be used to extract knowledge that can then be utilized to make decisions. According to different polls connected to digitization undertaken by international organizations such as IDC, it is expected that because all consumers interact with data in their everyday lives, real-time data would account for 30% of worldwide data available in 2025 [3]. Businesses want to take advantage of this possibility to process data in real-time to increase profits. By 2025, 75% of the world's population will be dealing with data daily, and the number of individuals interacting with data will continue to rise as billions of Internet of Things (IoT) devices are linked throughout the world [4–6]. Smartphones, numerous sensors, the web, online transactions, IoT devices, and other sources of data are constantly generating data. This real-time data contains a wealth of information and wisdom. Unlike batch processing of static data, continuously generated real-time data (also known as data streams) necessitates different processing methods to extract useful patterns and knowledge, because data streams arriving at high speeds must be processed

as soon as possible, and delayed processing renders data streams invalid. Businesses aim to acquire data and process it to extract hidden information that will benefit their operations, i.e., make the most of data [7]. Sensitive and private data, such as financial status, medical issues, location, and so on, are managed by diverse stakeholders throughout the whole cycle, from data collection through knowledge extraction. The open problem in data streams mining is to fine-tune the balance between maximal data utilization to find relevant patterns and privacy preservation [8].

Privacy preservation is a vital requirement for both static and streaming data. Hospitals and other organizations frequently need to disseminate microdata for scientific study and knowledge-based decision-making [9], such as illness analysis and prediction. Such data is frequently recorded in the form of table D. The recorded data D consists of different types of attributes such as sensitive attributes, explicit identifiers (ID), quasi-identifiers, and others. "The explicit identifier (ID) can identify individuals (e.g. name and social security number); quasi-identifier (QI) is a set of attributes that can potentially identify an individual, such as zip code, date of birth, and gender (in general, we assume that QI is background knowledge possessed by attackers); and sensitive attributes (e.g. visit date, if it is not contained in QI)". To avoid the leaking of personal information, clear identifying information must be deleted when microdata is provided. Individual privacy might potentially be jeopardized if additional public data is linked to QI [10]. As a result, adequate mechanisms for privacy-protected data posting are necessary. Techniques for preserving privacy, such as k-anonymity, l-diversity, t-closeness, and so on, are appropriate for anonymizing static datasets just once in an offline setting. But when anonymization is required regularly for data streams and execution speed is critical, these techniques may be deemed ineffective. Existing privacy preservation approaches for streaming data failed to provide the required privacy level, resulting in a greater IL. On the other hand, the time and space requirements for performing the privacy preservation of streaming data become a difficult research topic. As a result, an effective solution is necessary to accomplish the streaming data's complete privacy protection principles.

However, before applying the privacy preservation and knowledge discovery, another vital challenge is the data approximation and pre-processing of periodically streamed data [11]. There is a lack of effective techniques to perform the data approximation. Every day, more than 2.5 quintillion bytes of data are produced on the internet. Big data refers to data that is growing in terms of volume, diversity, and velocity. To examine this data, one must first gather it, store it in a secure location, clean it, and analyze it. Dealing with useless or redundant data is one of the most difficult difficulties that big data developers confront [11]. It takes a lot of time and resources to keep and analyze all of this extra data, yet it's all for naught in the end. As a result, removing duplicate data becomes critical for lowering analysis costs and reducing duplication. Data cleaning may be done using a variety of ways [12–15], but first, you must determine how much usable data is available in the dataset. As a result, before removing duplicate data from a data stream or database, it's vital to identify what data is different or unique. Apart from this, the online streaming data may contain noisy data which can result in incorrect knowledge discovery and decision making. Therefore, before performing the streaming data mining (knowledge discovery and decision making), the appropriate mechanisms are required to approximate and pre-process the online streaming data.

To end this, we proposed a novel framework EAPPA in this paper to achieve an efficient knowledge discovery mechanism from the input streaming data. The EAPPA framework consists of two vital phases such as (1) streaming data approximation and pre-processing, and (2) dynamic privacy preservation. The periodically collected incoming streaming data contains redundant information and different noises around all attributes. The FM algorithm is designed in this paper to approximate the number of unique attributes in the input data stream. The highlight of the FM algorithm is that it employs less time and memory while running. After approximation of the periodically received streaming data, we applied the lightweight mechanism of data cleaning using Natural Language Processing (NLP). The clustering techniques play the significant role in privacy preservation [16–19]. The structured input of the first phase is then processed for privacy preservation where we have designed the adaptive clustering-based k-anonymization. We extend the adaptive k-anonymized clustering by l-diversity to prevent attribute disclosure. The remainder of the sections of this paper is ahead. In section 2, we present a brief study of various privacy preservation techniques. The design and methodology of the proposed EAPPA framework are presented in section 3. The experimental results and comparative analysis is presented in section 4. Conclusion and future works are disclosed in section 5.

2. Related Works

As the main focus of this paper is on the effective processing of streaming data mining concerning privacy preservation, we review different approaches to privacy preservation in this section. We begin with a discussion of privacy concepts and anonymous techniques for static datasets with one-time anonymization. Then, we review the state-of-art developments in privacy protection for publishing dynamic datasets and data streams. Finally, we demonstrate the key research issues of building privacy preservation approaches for data streams and the contributions of the suggested model.

A. Privacy Preservation for Static Datasets

Recently privacy preservation in static datasets like Online Social Networks (OSNs) becomes an essential requirement due to various cyber threats. Several techniques [20–29] have been proposed to achieve privacy preservation over the static OSN datasets recently. To understand the functionality of static dataset privacy preservation methods, we reviewed recent techniques with single and multiple sensitive attributes. A hybrid OSN privacy protection approach had proposed in [20]. They looked at identity and location privacy leaks and resilience. The game-based Markov decision process system had designed to maximize data value while retaining anonymity. For OSN publication, the authors provide a local differential privacy strategy to keep community structure data in [21]. This model's published versions produced synthetic social network information using the edge probability reconstruction structural constraint. Unanonymization of social networks using structural information had demonstrated in [22]. An efficient node matching mechanism had also designed. The clustering method was developed in [23] to achieve k-anonymity in OSNs. Initially, the author employed Particle Swarm Optimization (PSO) to construct the clustering strategy to reduce IL. The high processing cost of PSO-based clustering led to the development of a hybrid Genetic Method (GA)/PSO-based algorithm (PSO-GA). Recent research [24] used

OSN to demonstrate the impact of user attributes on de-anonymization accuracy. Their multipartite graph included user attribute diversity and essential characteristics. The multipartite graph had divided into groups. Using clustering to secure OSN privacy was proposed in [25]. They recommend clustering to achieve privacy for nodes, linkages, and properties in social networks. To ensure k-anonymity, OSN nodes were clustered by similarity. To achieve l-diversity privacy, k-anonymity was enhanced. The feature learning approach for privacy-preserving poisoning prevention had recently developed in [26]. They used feature learning to infer social relationships between users before building an inferred social graph. The privacy-preserving principles assumed a social network. Message obfuscation using message replication and sensitive attributes replacement approach had presented in [27]. They assessed each user's privacy trustworthiness in OSN based on their social behaviors. To maintain anonymity across all graph portions, the authors proposed in [28] combining distinct series. These are the dK-1 series for degree frequency, the dK-2 series for joint degree frequency, and the dK-3 series for edge connecting information. The "Customizable Reliable Differential Privacy (CRDP)" technique had suggested in [29]. They used the social distance to change the privacy protection settings for the shortest link between two nodes.

The recent privacy preservation methods [20–29] for static datasets revealed that applying k-anonymization is not sufficient to address all the privacy requirements. Therefore, k-anonymization had achieved using clustering with l-diversity or t-closeness methods. Such methods delivered effective privacy solutions for the static dataset, but cannot be suitable for the dynamic and streaming data. They failed to address the unstructured, dynamic, and high-volume real-time streaming data.

B. Privacy Preservation for Data Streams

Privacy preservation for data streams is not easier tasks due to reasons disclosed earlier in this paper; therefore, rare attempts were made on achieving the effective privacy preservation on dynamic or data streams recently. In this section, we reviewed some standard works for privacy preservation of data streams and highlight their limitations.

The authors of [30] suggested an anonymization approach with the generalization that allows for continuous data dissemination while maintaining privacy as new records are entered. It ensures that each release meets separate l-diversity requirements and that a new anonymized table issued does not introduce any inference channels with regard to previously released tables. However, this approach only allows for insertions. The authors of [31] suggested an m-invariance privacy model and a generalized anonymization mechanism to address both record insertions and deletions. The authors of [32] expanded m-invariance to m-distinct to address both external ("the dataset has updated with record insertions and/or deletions") and internal modifications ("the attribute values of each record are dynamically updated"). But methods in [30–32] cannot be directly applied to data streams. Anonymizing data streams varies from anonymizing dynamic datasets because the inferences that may be drawn while anonymizing data streams differ. Anonymizing a dynamic dataset necessitates duplicated table releases. Because each record is anonymized only once, this inference is impossible in an anonymizing data stream. An attacker can analyze the output anonymized tuples to make conclusions in the anonymizing

data stream. Because data streams may only be scanned in one pass and executed in a pipeline, strict guarantees on the maximum permissible latency between entering input and matching anonymized output are necessary. Thus, efficiency is critical when anonymizing data streams.

The authors of [33] introduced k-anonymity for publishing privacy-preserving data streams in the case of a person with numerous records, as well as a clustering method to anonymize data streams and assure the anonymized data freshness through fulfilling given latency limits. They do, however, assign different equivalence classes to various recordings of the same person. As a result, they lose the link between the weights of a susceptible property that belongs to the same person. In [34], the authors have proposed data stream anonymization built on clustering to speed the anonymization process and prevent information loss by taking into account temporal limits on tuple publishing and cluster reuse. They named this approach as “Fast clustering-based k-Anonymization approach for Data Streams (FADS)”. However, methods in [33] and [34] failed to address the data redundancy and attribute disclosure problems. In [35], authors have suggested two novel privacy techniques: “improved identify-reserved (α, β)-anonymity and l-diversity”. They created the DAnonyIR model using a clustering approach that uses several decision functions to reduce IL due to generalization. The authors of [36] initially looked at the privacy issue of broadcasting transactional data streams using a sliding window. Then, to anonymize a sliding window in real-time, they suggested two dynamic techniques with generalization and suppression. Information of the same user or tuple may be distributed among different windows. Another recent mechanism called IDEA (“Incomplete Data strEam Anonymization”) had been proposed in [37] for the continuous data stream. They addressed the incomplete data streams challenges in IDEA with clustering-based anonymization. A slide-window-based processing architecture was implemented in IDEA to continually anonymize data streams, with each tuple being produced with anonymized clusters. However, the sliding window-based approach in [36] and [37] to anonymize the data streams suffered from data redundancy and sensitive data loss problems.

C. Contributions

Fewer studies [33–37] proposed so far on handling privacy preservation requirements for data streams. Each method [33–37] has suffered from serious challenges for the privacy preservation of data streams. The key challenges such as IL due to sliding window technique, redundant data anonymization, lack of data stream cleaning, incomplete privacy preservation, etc. It motivates us to propose a novel framework called EAPPA for continuous data streaming using effective algorithms. The contributions of the EAPPA framework are described below.

- We propose the integrated stream data mining framework that consists of stream data approximation, data cleaning, adaptive clustering-based k-anonymization, and l-diversity.
- The efficient FM algorithm is designed in EAPPA to perform the approximation using a suitable hash function of the currently received data stream in one pass with minimum time and memory requirements. The approximated data streams are pre-processed to remove the unwanted noise contains without loss of sensitive and original information.

- We propose the clustering-based improve k-anonymization algorithm to anonymize the periodically received pre-processed data streams within time constraints using the similarity measure technique. This approach ensures the k-anonymized clusters with at-least k-anonymized data tuples. To prevent attribute disclosure in k-anonymized clusters, we introduced the l-diversity approach in the EAPPA model.
- By changing the number of data stream tuples and clusters, we investigate the performance of the EAPPA framework with similar techniques on a real-world dataset.

3. Proposed Model

Figure 1 shows the overall functionality of the proposed model of processing the data streams. As discussed above, the phases of the EAPPA model are two-fold. The first contribution is called Data Approximation and Pre-processing (DAP) and the second contribution is called Adaptive Clustering-based Privacy Preservation (ACPP). Figure 1 shows the processing of both phases step-by-step using the delay constraint approach. The functionality is mainly derived from the concept of periodical data processing of incoming streaming data. Before acquiring the data streams, we first initialized the timer t. The acquired data streams are then processed using the DAP. We applied the FM algorithm to reduce the redundant data streams followed by the pre-processing algorithm to filter out the noisy contains. The sequentially received data streams are first checked for duplication, pre-processed, and then stored into output matrix D. The functionality of DAP continues until the timer reaches a pre-defined threshold value λ. Once the delay constraint is satisfied, EAPPA launches the ACPP phase which takes the D as input. In the ACPP phase, we first compute the k-number of centroids using a basic k-means algorithm. Then estimate the similarities of each data stream tuple with its corresponding centroid tuple. All the estimated similarities are recorded and sorted in descending order. Finally, the k-anonymized clusters are formed as per the sorted order to ensure the k-anonymity. As the k-anonymization failed to prevent the attribute disclosure, we applied the l-diversity privacy notion on each cluster. The clusters ensuring the k-anonymity and l-diversity are then published before taking the next periodic data stream. We explore the design of both phases is explored in the below section.

A. DAP

The functionality of DAP is consists of FM-based data streams approximation and pre-processing of each received tuple without losing the sensitive information. Figure 2 shows the methodology of the DAP phase in detail. The input data stream S holds the 1 or more tuples. If the number of tuples in S is more than 1, then we initiate the FM algorithm and pre-processing algorithm. The main aim of the FM algorithm is to estimate the total number of distinct data streams, but we explored the FM algorithm to extract the distinct data streams and discard the redundant data streams. This process continues for each incoming tuple. Algorithm 1 shows the modified FM algorithm integrated pre-processing algorithm 2.

As shown in algorithm 1, we have effectively utilized the FM algorithm to approximate the periodic data stream. The advantage of estimating the number of unique tuples using the FM algorithm is explored in this paper to identify the redundant or duplicate incoming tuple. The core functionality of the FM algorithm is belonging to the steps such as defining the hash function (step 9), computing that the hash function of each attribute belongs to each stream (step 11), binary conversion of each hash value (step 12), counting trailing zeros of a binary number (step 13), and computing total distinct streams in S using step 15 and 16. After discovering the number of distinct elements, we utilized that parameter to discover whether the current data stream s is unique or redundant and accordingly we take the actions as shown in steps 17–24 in algorithm 1. From algorithm 1, we called algorithm 2 for the pre-processing of the input data stream and stored the final pre-processed streams in D. The core part of this algorithm is the manual discovery of the unique attribute of each streaming tuple and defining hash function. For this work, we defined the hash function shown in Eq. (1).

$$h\left(x\right) \leftarrow \left(a.\,x + b\right) mod c$$

1

Where, we set the x represents the attribute value of current stream. We set $a = 1, b = 6 \& c = 32$ to compute the hash values.

Algorithm 1: DAP

Input

$S : inputdatastream$

$\lambda : pre - definedtimeconstraint$

$j : uniquesenstiveattribute$

Output

$D : Approximtedandpre - processeddatastream$

1. Initial timer$t = 0$

2. $s \leftarrow acquire\,(stream)$

3. $S \leftarrow add\,(s)$

4. If$(size\,(S) > 1)$

5. For$i = 1 : size\,(S)$

6. Estimate the sensitive unique attribute from all streams

7. $w\,(i) \leftarrow S\,(i, j)$, record the j^{th}position unique value

8. End For

9. Define hash function for stream w using Eq. (1) and apply

10. For$i = 1 : size\,(w)$

11. $h\,(i) \leftarrow (a.\,w\,(i) + b)\,modec$

12. $h\,(i) \leftarrow binary\,(h\,(i))$

Algorithm 1: DAP

13. $r\left(i\right) \leftarrow trailingzeros\left(h\left(i\right)\right)$

14. End For

15. Compute maximum value:$R \leftarrow \max\left(r\right)$

16. Compute distinct tuples:$N \leftarrow 2^{R}$

17. If$\left(N == size(w)\right.$

18. Current stream s is unique and apply pre-processing

19. $p \leftarrow algorithm2\left(s\right)$

20. $t++$

21. Else

22. Discard stream s from stream S as:$S \leftarrow substract\left(s\right)$

23. $t++$

24. End If

25. Else

26. $S \leftarrow algorithm2\left(S\right)$

27. $t++$

28. End If

29. $D \leftarrow add\left(p\right)$

30. Check time constraint

31. If$\left(t \geq \lambda\right)$

Algorithm 1: DAP
32. Return (D) ,Launch ACPP phase 33. Reset timer $t = 0$, goto step 1 34. Else 35. goto step 2 36. End If

Algorithm 2

shows the pre-processing of each input stream. First, we have checked whether the attribute is a string. If it strings then, we performed the lemmatization using NLP to convert the incorrect strings into the meaningful form and remove the noise in the string. Apart from this, we have addressed the challenges of missing or incomplete data in this work for numeric attributes. We have discovered the numeric attributes and replaced them with relevant values using the function. The discover the most relevant value using statistical analysis of same attributed of other steams.

Algorithm 2: Data Pre-processing

Input

$s : inputdatastream$

Output

$p : pre - processeddatastream$

1. Acquisition of test stream s

2. For each attribute each attribute$i = 1 : size\,(s)$

3. If$(s\,(i) == string)$

4. $p\,(i) \leftarrow Lemmatization\,(s\,(i))$

5. End If

6. If$(s\,(i) == NULL)$

7. $p\,(i) \leftarrow newVal\,()$

8. End If

9. End For

10. Return(p)

B. ACPP Phase

This phase belongs to achieving the complete privacy preservation of periodically collected data stream D without losing information. We estimate the centroids using the existing k-means clustering before doing the k-anonymization. The arguments for using k-means clustering are that (1) it is straightforward to group tuples based on their similarities, (2) it is quick and creates efficient clusters, and (3) outliers in the dataset cannot be avoided using k-means and all outliers have privacy. Therefore, we form the initial centroids of input data stream D as:

$$[C, \ddot{C}] = kmeans(D, n)$$

2

Where, n defines the number of clusters (in this work, we have set n as 30, 60, 90, 120, and 150). C represents the set of n clusters where the tuples of D are distributed. $\ddot{C}$represents the centroid tuple for each cluster. The k-means algorithm failed to achieve the complete k-anonymity across all the clusters. The clusters are k-anonymized if they satisfied the constraint of having exactly k-number of tuples in each cluster. The value of k is discovered by:

$$k = \left| \frac{size\,(D)}{n} \right|$$

3

Therefore, we have proposed the adaptive clustering mechanism in this paper to achieve the complete k-anonymization privacy notion for periodically received data stream D. This is done by enhancing the output of k-means clustering as showing in algorithm 3. As showing in algorithm, it takes inputs such as $C, \ddot{C}, n, x,$ and y and return the set clusters O that ensures the k-anonymity. Before enhancing the present clusters, we have first estimate the distance between i^{th} tuple of j^{th} cluster and j^{th}centroid. This distance is measured by Manhattan distance technique in $getDist(.)$ function. It is calculated as the sum of the absolute differences among two numeric vectors of two tuples. All the distances are measured into the vector P which contains the entire tuple and its distance value. We then sorted the tuples in P in descending order of distance values. Finally, the clusters are reformed that ensures maximum k-tuples per cluster criteria to ensure the k-anonymity. The proposed clustering takes simple approach to achieve the k-anonymity for the current data stream D. The number of tuples in each cluster should be less than or equal to k. Therefore, algorithm 3 returns the clusters of similar size to achieve the k-anonymization with less IL.

Algorithm 3: K-anonymized adaptive clustering

Inputs

$C : Setofclusters$

$\ddot{C} : setofcentroidtuplesforeachcluster$

$n : numberofclusters$

$x : numberoftuplesinD$

$y : numberofattributesineachtuple$

Output

$O : setofk - anonimizedclusters$

Algorithm 3: K-anonymized adaptive clustering

1. Initialize, $P \leftarrow zeros(x, y+1)$,$q = 1$

2. For $i = 1 : n$

3. For$j = 1 : size\left(\mathrm{C}\left(\mathrm{i}\right)\right)$

4. $\mathrm{d} \leftarrow getDist\left(\mathrm{C}\left(\mathrm{i},\mathrm{j}\right), \ddot{C}(i)\right)$

5. $P\left(q, 1 : x\right) \leftarrow \mathrm{C}\left(\mathrm{i},\mathrm{j}\right)$

6. $P\left(q, y+1\right) \leftarrow \mathrm{d}$

7. $q++$

8. End For

9. End For

10. $temp \leftarrow sort\left(\mathrm{descening}, P\left(:, m+1\right)\right)$

11. for$i = 1 : size\left(temp\right)$

12. for $j = 1 : n$

13. if$\left(size\left(\mathrm{O}\left(\mathrm{j}\right)\right) \leq \mathrm{k}\right)$

14. $\mathrm{O}\left(\mathrm{i},\mathrm{j}\right) \leftarrow join\left(temp\left(i, :\right)\right)$

15. end if

16. end for

17. end for

18. Return(O)

Due to its limitations of attribute disclosure, background knowledge, and homogeneity, k-anonymity does not guarantee total privacy protection. The l-diversity resolved the k-anonymity issues. Therefore, we further extend the k-anonymized clusters with l-diversity notion in this paper. We used the entropy l-diversity idea to expand the clusters in O to meet the l-diversity requirement [38]. We calculated diversity using entropy for each k-anonymized cluster and stored the result in matrix L. The greedy approach had used to guarantee that each cluster met the l-diversity requirement. The procedure continues until all of the clusters are l-diverse. Because we are not removing any tuples from the cluster during the whole algorithm 4, the privacy concept of k-anonymity remains the same. As showing in algorithm 4, our aim is achieve the clusters with diversity below 1. We therefore rearranged the each cluster until we achieve the diversity level below 1. This is achieved by computing the current cluster with maximum diversity value and cluster with minimum diversity value. And according to that value $(Max + Min = l)$, we arranged the clusters in output vector F. Finally, we publish the privacy preserved data stream towards the intended destinations. The process of clustering repeated for each incoming data stream with similarity functionality, therefore it is named as adaptive clustering for privacy preservation.

Algorithm 4: l-diversity and publish

Input

$O : setofk - anonimizedclusters$

Output

$F : setofk - anonimizedandl - diverityensuredclusters$

1. For$j = 1 : size\,(\mathrm{O}\,(\mathrm{i}))$

2. $L\,(i) \leftarrow ComputeDiversity\,(O\,(i))$

3. End For

4. Ensures the l-diversity

5. While $(L < 1)$ do

6. $Max \leftarrow \max\,(L)$

7. $Min \leftarrow \min\,(L)$

8. $l \leftarrow Max + Min$

9. $F \leftarrow O - \{Max, Min\} + l$

10. End While

11. Publish(F)

12. Return (F)

4. Experimental Results

This section presents the outcomes of experimental work for performance analysis. To implement and evaluate the proposed model with state-of-art methods, we used the Python tool. The experiments were performed on Windows 10 Operating System with an Intel I5 processor and 8GB RAM. Each scenario has

been executed for 20 instances and then averaged their performances. The proposed method compares with three state-of-art data stream anonymization methods such as FADS [34], DAnonyIR [35], and IDEA [36]. To demonstrate the performance of the algorithms on different data distributions, we conduct experiments on two real-world datasets: Adult from the UCI repository [39]. This is standard dataset for studying k-anonymity algorithms.

To investigate the proposed method with state-of-art methods, we have measured three performance parameters such as “Degree of Anonymization (DoA), IL, and Execution Time (ET)”. The ET represents the average execution time for each scenario of 20 instances required to perform the OSN data anonymization. The DoA of any tuple is measure by measuring the number of assigned tuples in its cluster, i.e., tuple DoA is similar to the DoA of its cluster for each incoming data stream D.

$$DoA = \sum_{s=1}^{M} degree\left(\mathrm{s} \to \left(\mathrm{F}\left(\mathrm{i},\mathrm{j}\right)\right) \times i\right)$$

4

Where, $\mathrm{F}\left(\mathrm{i},\mathrm{j}\right)$ represents i^{th} tuple of j^{th} cluster, M represents number of periodically received data streams $M = 1, 2, \ldots, s$.

The IL metrics has computed according to the formulation presented in [39] as:

$$IL = \frac{SSE}{SST}$$

5

Where, SSE denotes sum of squares within cluster and SST denotes sum of squares among clusters.

A. Clustering Size Analysis

This section presents comparative results analysis for different techniques of privacy preservation of data streams. The results are analyzed with respect to the varying number of clusters (i.e., value k). According to value k, the predefined time constraint value also changes in the proposed model. Figures 3, 4, and 5 show the outcomes of DoA, IL, and ET using FADS, DAnonyIR, IDEA, and the proposed EAPPA methods.

As the cluster sizes increases, the maximum numbers of tuples in each data stream are increases. The first finding to be drawn from the DoA result (Fig. 3) is that anonymization decreases as the size of the cluster rise. Fundamentally, this is due to the fact that a small number of clusters allows for a high number of k-anonymous users to persist, but increasing the size of a cluster reduces the proportion of at least k-anonymous persons in each cluster The proposed method beat all existing methods in terms of overall performance when compared to the other two strategies. The EAPPA methodology addressed the

challenges of existing techniques, and it reflects in the achieved results. Figure 4 depicts another important result of IL for this study utilizing all four techniques. The DoA performance with contrast effects follows a similar trend in IL with changing cluster sizes. As the number of clusters grow, the number of anonymous tuples decreases, resulting in reduced loss of sensitive information. As a result, for a large number of clusters, there is less sensitive IL than for a small number of clusters. The suggested anonymization methodology EAPPA lowered the IL ratio considerably with enhanced DoA compared to state-of-art data stream privacy preservation approaches. The key reasons for performance improvement using EAPPA techniques are (1) effective mechanism to remove the redundant tuples from the streaming data with data cleaning which is missing in all existing techniques, (2) adaptive k-anonymized clusters formation with minimum IL and computational requirements (Existing methods relied on sliding window approach), and (3) k-anonymized clustered enhanced by applying l-diversity privacy. Finally, the time complexity outcomes showing in Fig. 5 claims that the proposed model takes less time to achieve strong privacy preservation (DOA) with minimum IL.

B. Data Stream Size Analysis

After analyzing the effects of varying clustering sizes on the performances, we further aimed to investigate the effective increasing stream data size on privacy preservation performances using different techniques. This section presents the outcomes for varying data stream sizes 2000, 6000, 10000, 14000, and 18000 tuples with a fixed number of clusters set to 90. We set limits to acquire the number of tuples from the stream data to investigate data stream size varying. Figures 6, 7, and 8 demonstrate the DoA, IL, and ET performances respectively using FADS, DAnonyIR, IDEA, and the proposed EAPPA methods. Figure 6 illustrates that DoA has improved as data size has grown. It has been discovered that the growth in DoA value is virtually exponential. Figure 7 shows the results of IL with different data sizes for each approach. Because the number of tuples increases, the created clusters become more significant, resulting in a lower IL. Figure 8 reveals the optimal time requirements to achieve the higher DoA and lower IL using the EAPPA method compared to all existing techniques. Finally, we averaged results in Table 1 for each method for each performance metric. We already disclosed the reasons of performance improvement using the EAPPA method over the exiting solutions.

Table 1
Comparative Analysis of Average Performances

	FADS	DAnonyIR	IDEA	EAPPA
DoA	4799	5158	**5367**	**5799**
IL	47.16	39.18	**37.16**	**35.05**
ET	338.34	**278.11**	305.34	281.36

5. Conclusion And Future Works

The novel EAPPA framework has been proposed in this paper with aim of overcoming the challenges of processing the data streams effectively for knowledge discovery. The EAPPA approach mainly focused on achieving stream data approximation and privacy preservation by considering the challenges of existing techniques such as data redundancy, sensitive information loss, and complete privacy preservation. To remove the redundant tuples from the streaming data and data noise, we have designed the DAP using FM and NLP techniques. To prevent data loss, we have designed adaptive clustering to ensure k-anonymization. Then, k-anonymized clustered enhanced by applying l-diversity privacy to achieve the complete privacy preservation of data streams. The experimental results show that EAPPA improved the DoA performance compared to recent methods by 18.45% and IL performance reduced by 17.6% with minimum computational requirements. There are some further directions to extend this work such as (1) investigating the performance of the EAPPA method using different hash functions in the FM algorithm, (2) improving the accuracy of handling the missing data, (3) analyzing the performances using other datasets.

References

1. Kolajo, T., Daramola, O. & Adebiyi, A. Big data stream analysis: a systematic literature review. J Big Data 6, 47 (2019). https://doi.org/10.1186/s40537-019-0210-7.
2. Wankhade, K.K., Dongre, S.S. & Jondhale, K.C. Data stream classification: a review. *Iran J Comput Sci* **3,** 239–260 (2020). https://doi.org/10.1007/s42044-020-00061-3.
3. Gama, J. (2012). A survey on learning from data streams: current and future trends. Progress in Artificial Intelligence, 1(1), 45–55. doi:10.1007/s13748-011-0002-6.
4. Mahajan, H.B., Badarla, A. & Junnarkar, A.A. CL-IoT: cross-layer Internet of Things protocol for intelligent manufacturing of smart farming. J Ambient Intell Human Comput 12, 7777–7791 (2021). https://doi.org/10.1007/s12652-020-02502-0
5. Mahajan, H.B., & Badarla, A. (2018). Application of Internet of Things for Smart Precision Farming: Solutions and Challenges. International Journal of Advanced Science and Technology, Vol. Dec. 2018, PP. 37-45.
6. Mahajan, H.B., Badarla, A. Cross-Layer Protocol for WSN-Assisted IoT Smart Farming Applications Using Nature Inspired Algorithm. Wireless Pers Commun 121, 3125–3149 (2021). https://doi.org/10.1007/s11277-021-08866-6.
7. Sun D, Zhang G, Zheng W, Li K. Key technologies for big data stream computing. In: Li K, Jiang H, Yang LT, Guzzocrea A, editors. Big data algorithms, analytics and applications. New York: Chapman and Hall/CRC; 2015. p. 193–214. ISBN 978-1-4822-4055-9.
8. Joseph, S., Jasmin E.A., & Chandran, S. (2015). Stream Computing: Opportunities and Challenges in Smart Grid. Procedia Technology, 21, 49–53. doi:10.1016/j.protcy.2015.10.008.
9. Ninghui Li, Tiancheng Li, & Venkatasubramanian, S. (2010). Closeness: A New Privacy Measure for Data Publishing. IEEE Transactions on Knowledge and Data Engineering, 22(7), 943–956. doi:10.1109/tkde.2009.139.

10. Fung, Benjamin & Wang, ke & Chen, Rui & Yu, Philip. (2010). Privacy-Preserving Data Publishing: A Survey of Recent Developments. ACM Comput. Surv.. 42. 10.1145/1749603.1749605.
11. Zakerzadeh H, Aggarwal CC, Barker K (2016) Managing dimensionality in data privacy anonymization. Knowl Inf Syst 49(1):341–373.
12. Zhang Y., Szabo C., Sheng Q.Z. (2014) Cleaning Environmental Sensing Data Streams Based on Individual Sensor Reliability. In: Benatallah B., Bestavros A., Manolopoulos Y., Vakali A., Zhang Y. (eds) Web Information Systems Engineering – WISE 2014. WISE 2014. Lecture Notes in Computer Science, vol 8787. Springer, Cham. https://doi.org/10.1007/978-3-319-11746-1_29.
13. Shaoxu Song, Fei Gao, Aoqian Zhang, Jianmin Wang, and Philip S. Yu. 2021. Stream Data Cleaning under Speed and Acceleration Constraints. ACM Trans. Database Syst. 46, 3, Article 10 (September 2021), 44 pages. DOI:https://doi.org/10.1145/3465740.
14. Peter M. Fischer, Kyumars Sheykh Esmaili, and Renée J. Miller. 2010. Stream schema: Providing and exploiting static metadata for data stream processing. In Proceedings of the 13th International Conference on Extending Database Technology. 207–218. DOI: https://doi.org/10.1145/1739041.1739068.
15. Ester Livshits, Benny Kimelfeld, and Sudeepa Roy. 2018. Computing optimal repairs for functional dependencies. In Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems. 225–237. DOI: https://doi.org/10.1145/3196959.3196980.
16. Reddy KSS, Bindu CS. A review of density-based clustering algorithms for big data analysis. In: International conference on I-SMAC (IoT in Social, Mobile, Analytic, and Cloud), Palladam, India 10–11 February 2017, IEEE. 2017. https://doi.org/10.1109/i-smac.2017.8058322.
17. Deepa MS, Sujatha N. Comparative study of various clustering techniques and its characteristics. Int J Adv Netw Appl. 2014;5(6):2104–16.
18. Zubaroğlu, A., Atalay, V. Data stream clustering: a review. *Artif Intell Rev* **54,** 1201–1236 (2021). https://doi.org/10.1007/s10462-020-09874-x.
19. Xiao X, Tao Y (2008) Dynamic anonymization: accurate statistical analysis with privacy preservation. In: Proceedings of the 27th ACM SIGMOD international conference on management of data, pp 107–120.
20. Qu, Y., Yu, S., Gao, L., Zhou, W., & Peng, S. (2018). A Hybrid Privacy Protection Scheme in Cyber-Physical Social Networks. IEEE Transactions on Computational Social Systems, 1–12. doi:10.1109/tcss.2018.2861775.
21. Liu, Peng & Xu, YuanXin & Jiang, Quan & Tang, Yuwei & Guo, Yameng & Wang, Li-e & Li, Xianxian. (2019). Local Differential Privacy for Social Network Publishing. Neurocomputing. 391. 10.1016/j.neucom.2018.11.104.
22. Shao, Y., Liu, J., Shi, S., Zhang, Y., & Cui, B. (2019). Fast De-anonymization of Social Networks with Structural Information. Data Science and Engineering. doi:10.1007/s41019-019-0086-8.
23. Yazdanjue, N., Fathian, M., & Amiri, B. (2019). Evolutionary Algorithms For k-Anonymity In Social Networks Based On Clustering Approach. The Computer Journal. doi:10.1093/comjnl/bxz069.

24. Zhang C., Wu S., Jiang H., Wang Y., Yu J., Cheng X. (2019) Attribute-Enhanced De-anonymization of Online Social Networks. In: Tagarelli A., Tong H. (eds) Computational Data and Social Networks. CSoNet 2019. Lecture Notes in Computer Science, vol 11917. Springer, Cham. https://doi.org/10.1007/978-3-030-34980-6_29.
25. Siddula, M., Li, Y., Cheng, X., Tian, Z., & Cai, Z. (2019). Anonymization in Online Social Networks Based on Enhanced Equi-Cardinal Clustering. IEEE Transactions on Computational Social Systems, 1–12. doi:10.1109/tcss.2019.2928324.
26. Zhao, P., Huang, H., Zhao, X., & Huang, D. (2020). P3: Privacy-Preserving Scheme Against Poisoning Attacks in Mobile-Edge Computing. IEEE Transactions on Computational Social Systems, 7(3), 818–826. doi:10.1109/tcss.2019.2960824.
27. Cai, Y., Zhang, S., Xia, H., Fan, Y., & Zhang, H. (2020). A Privacy-preserving Scheme for Interactive Messaging over Online Social Networks. IEEE Internet of Things Journal, 1–1. doi:10.1109/jiot.2020.2986341.
28. Gao, Tianchong & Li, Feng. (2020). Protecting Social Network With Differential Privacy Under Novel Graph Model. IEEE Access. 8. 185276-185289. 10.1109/ACCESS.2020.3026008.
29. Qu, Youyang & Yu, Shui & Zhou, Wanlei & Chen, Shiping & Wu, Jun. (2020). Customizable Reliable Privacy-Preserving Data Sharing in Cyber-Physical Social Network. IEEE Transactions on Network Science and Engineering. PP. 1-1. 10.1109/TNSE.2020.3036855.
30. Aldeen, Y. A. A. S., Salleh, M., & Aljeroudi, Y. (2016). An innovative privacy preserving technique for incremental datasets on cloud computing. Journal of Biomedical Informatics, 62, 107–116. doi:10.1016/j.jbi.2016.06.011.
31. Xiao, X., & Tao, Y. (2007). M-invariance. Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data - SIGMOD '07. doi:10.1145/1247480.1247556.
32. Hasan, A., Jiang, Q., Chen, H., & Wang, S. (2018). A New Approach to Privacy-Preserving Multiple Independent Data Publishing. Applied Sciences, 8(5), 783. doi:10.3390/app8050783.
33. Jianneng Cao, Carminati, B., Ferrari, E., & Kian-Lee Tan. (2011). CASTLE: Continuously Anonymizing Data Streams. IEEE Transactions on Dependable and Secure Computing, 8(3), 337–352. doi:10.1109/tdsc.2009.47.
34. Guo, K., & Zhang, Q. (2013). Fast clustering-based anonymization approaches with time constraints for data streams. Knowledge-Based Systems, 46, 95–108. doi:10.1016/j.knosys.2013.03.007.
35. Wang, J., Du, K., Luo, X. *et al.* Two privacy-preserving approaches for data publishing with identity reservation. *Knowl Inf Syst* **60,** 1039–1080 (2019). https://doi.org/10.1007/s10115-018-1237-3.
36. Wang, J., Deng, C., & Li, X. (2018). Two Privacy-Preserving Approaches for Publishing Transactional Data Streams. IEEE Access, 6, 23648–23658. doi:10.1109/access.2018.2814622.
37. L. Yang, X. Chen, Y. Luo, X. Lan and W. Wang, "IDEA: A utility-enhanced approach to incomplete data stream anonymization," in Tsinghua Science and Technology, vol. 27, no. 1, pp. 127-140, Feb. 2022, doi: 10.26599/TST.2020.9010031.

38. Siddula, M., Li, Y., Cheng, X., Tian, Z., & Cai, Z. (2019). Anonymization in Online Social Networks Based on Enhanced Equi-Cardinal Clustering. IEEE Transactions on Computational Social Systems, 1–12. doi:10.1109/tcss.2019.2928324.
39. U.M. L. Repository, Adult data set, https://archive.ics.uci.edu/ml/datasets/Adult, 2020.
40. Domingo-Ferrer, J., & Mateo-Sanz, J. M. (2002). Practical data-oriented microaggregation for statistical disclosure control. IEEE Transactions on Knowledge and Data Engineering, 14(1), 189–201. doi:10.1109/69.979982.

Figures

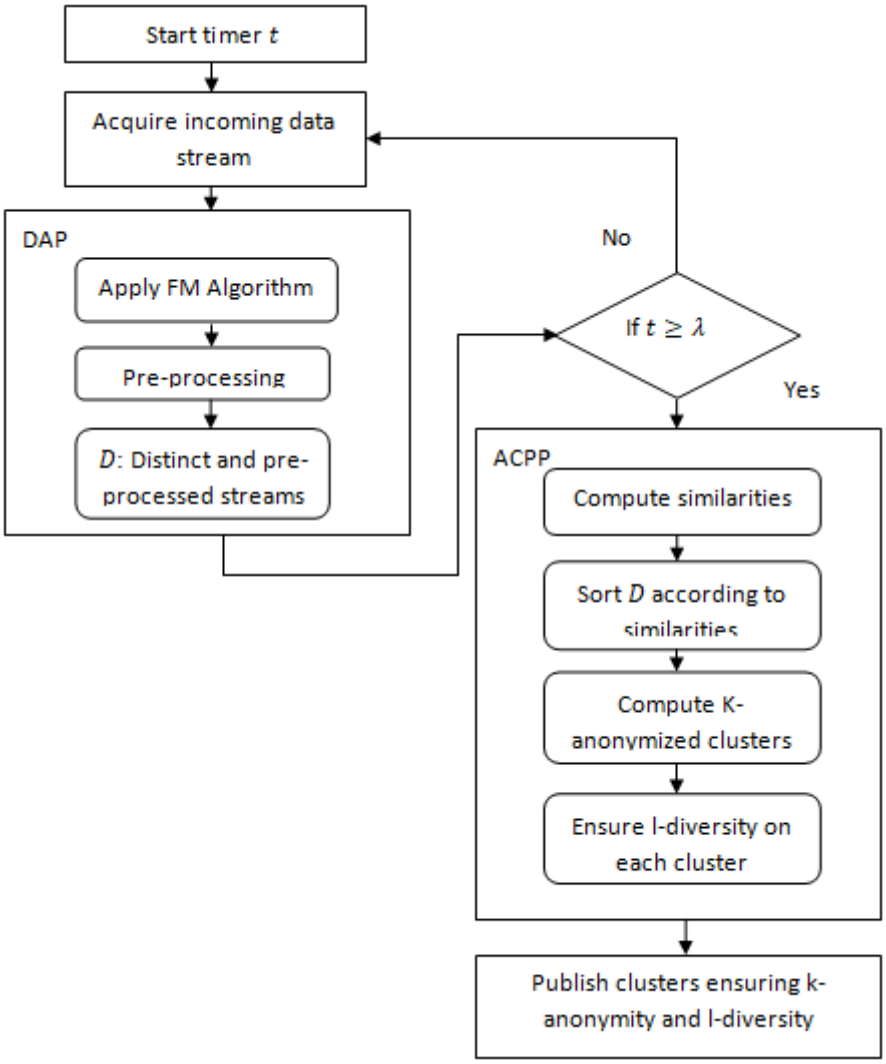

Figure 1

Architecture of proposed EAPPA system

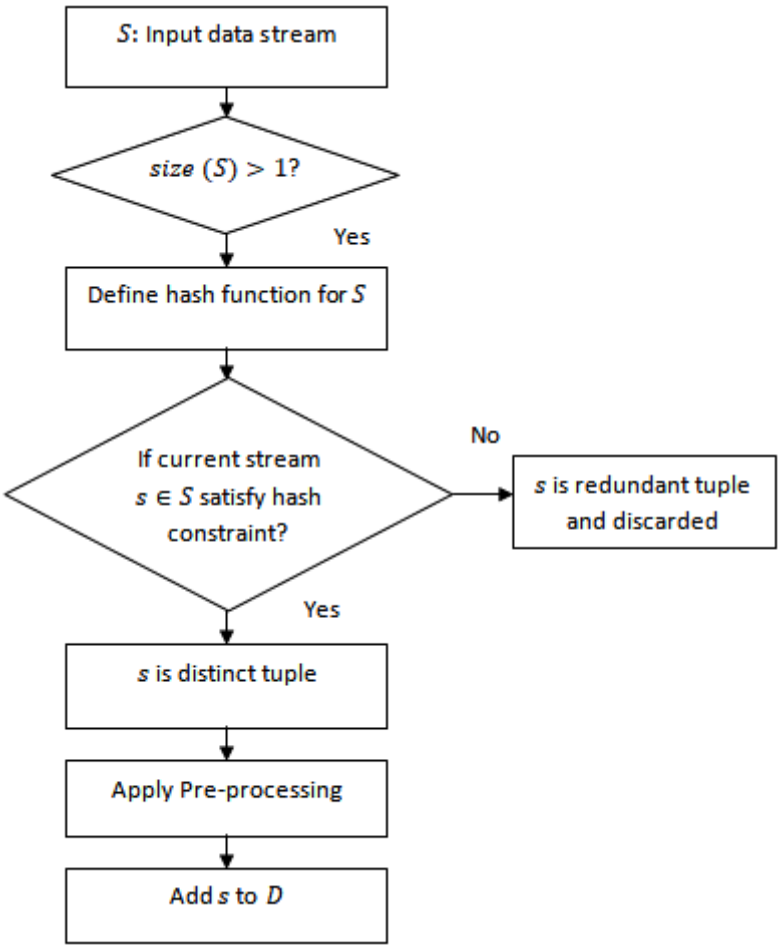

Figure 2

Architecture of DAP phase

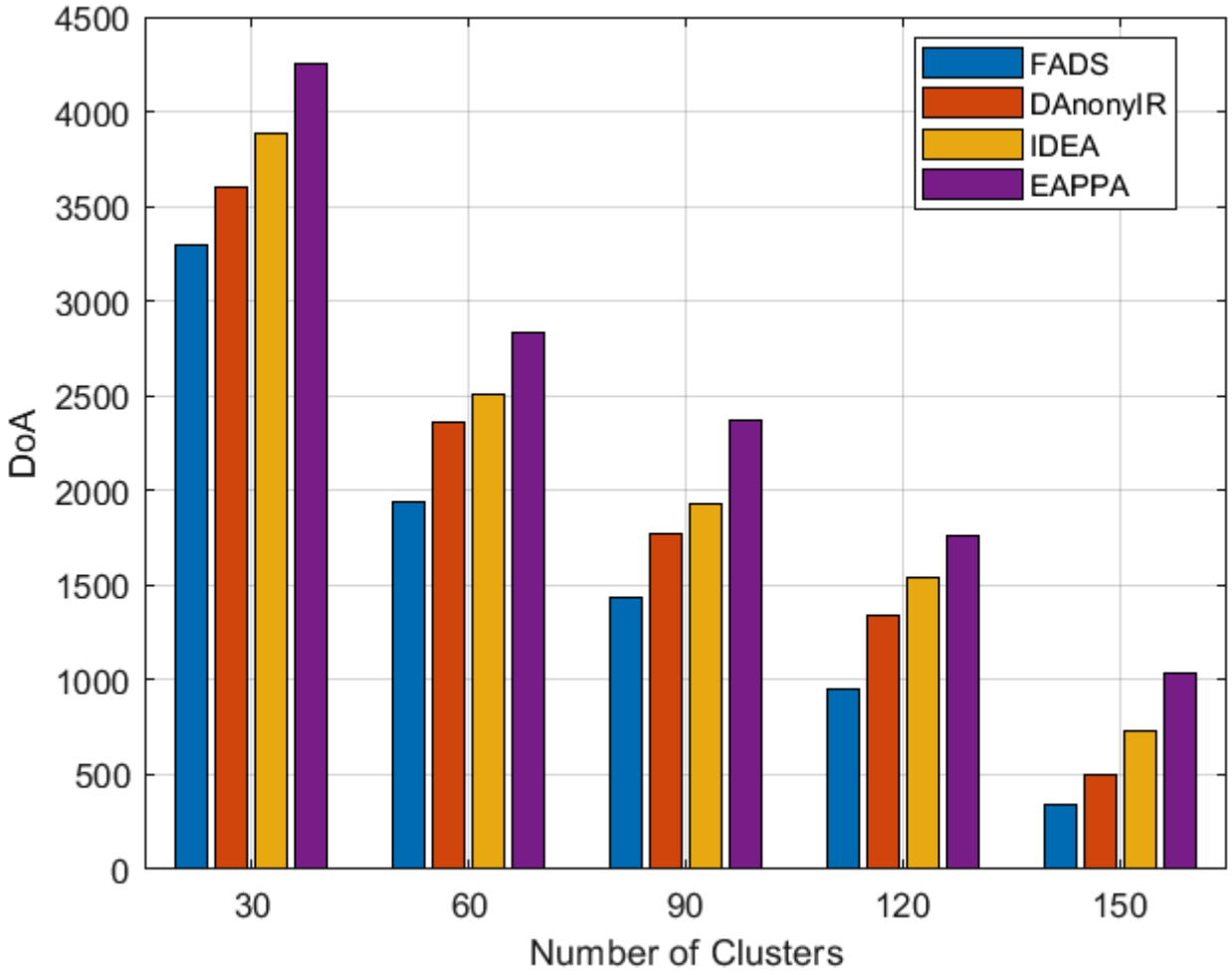

Figure 3

DoA performance investigation with varying clusters

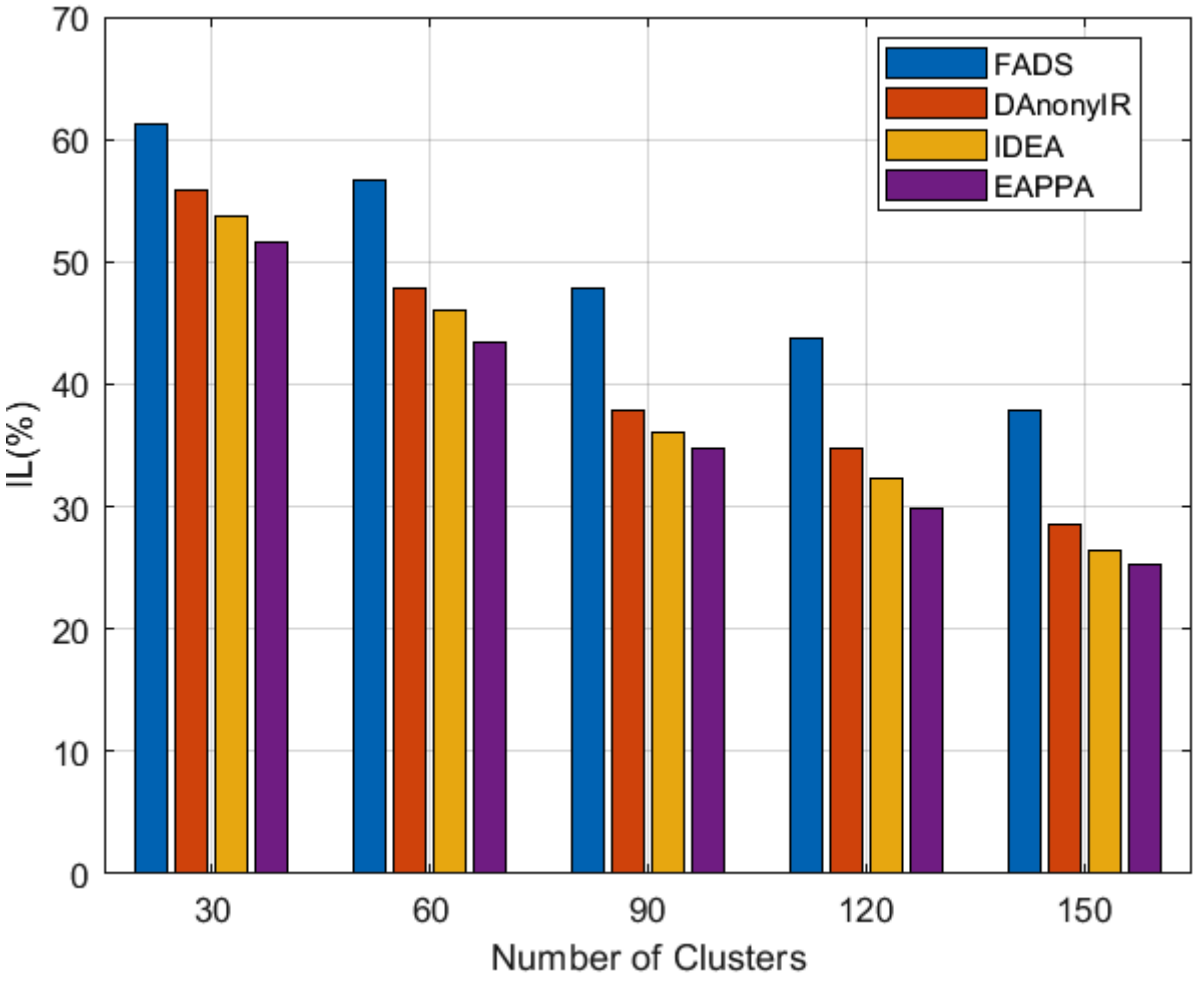

Figure 4

IL performance investigation with varying clusters

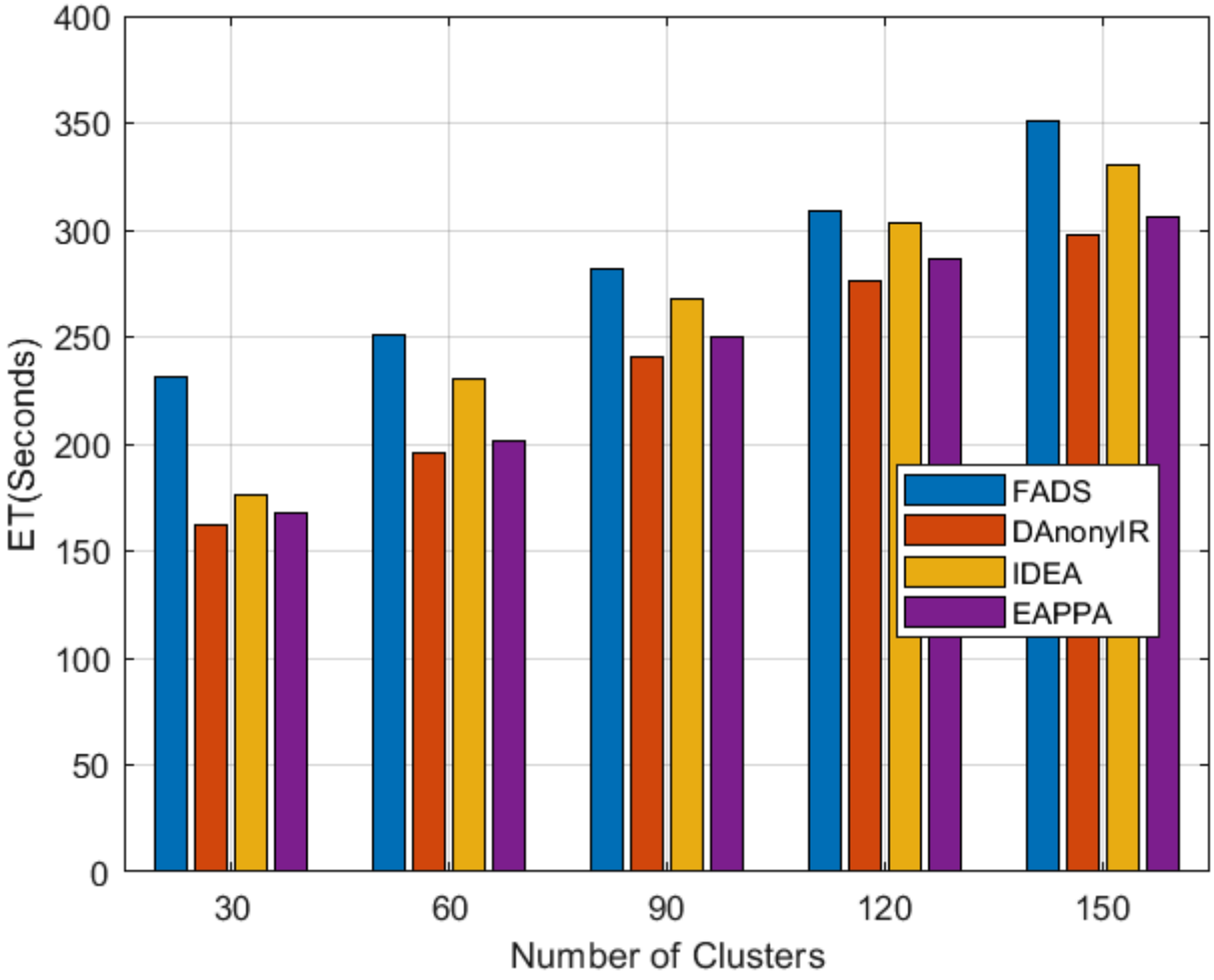

Figure 5

ET performance investigation with varying clusters

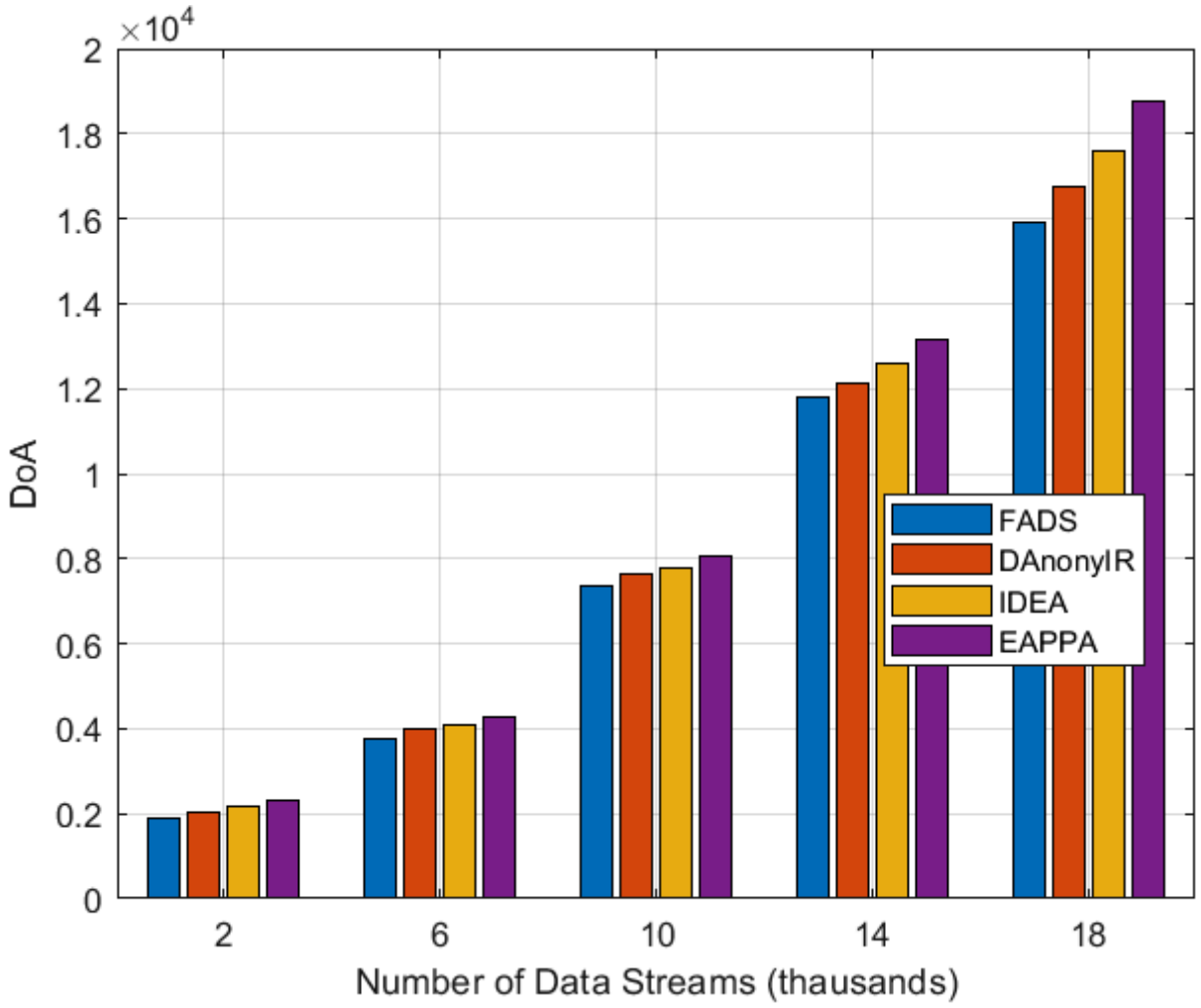

Figure 6

DoA performance investigation with varying stream data size

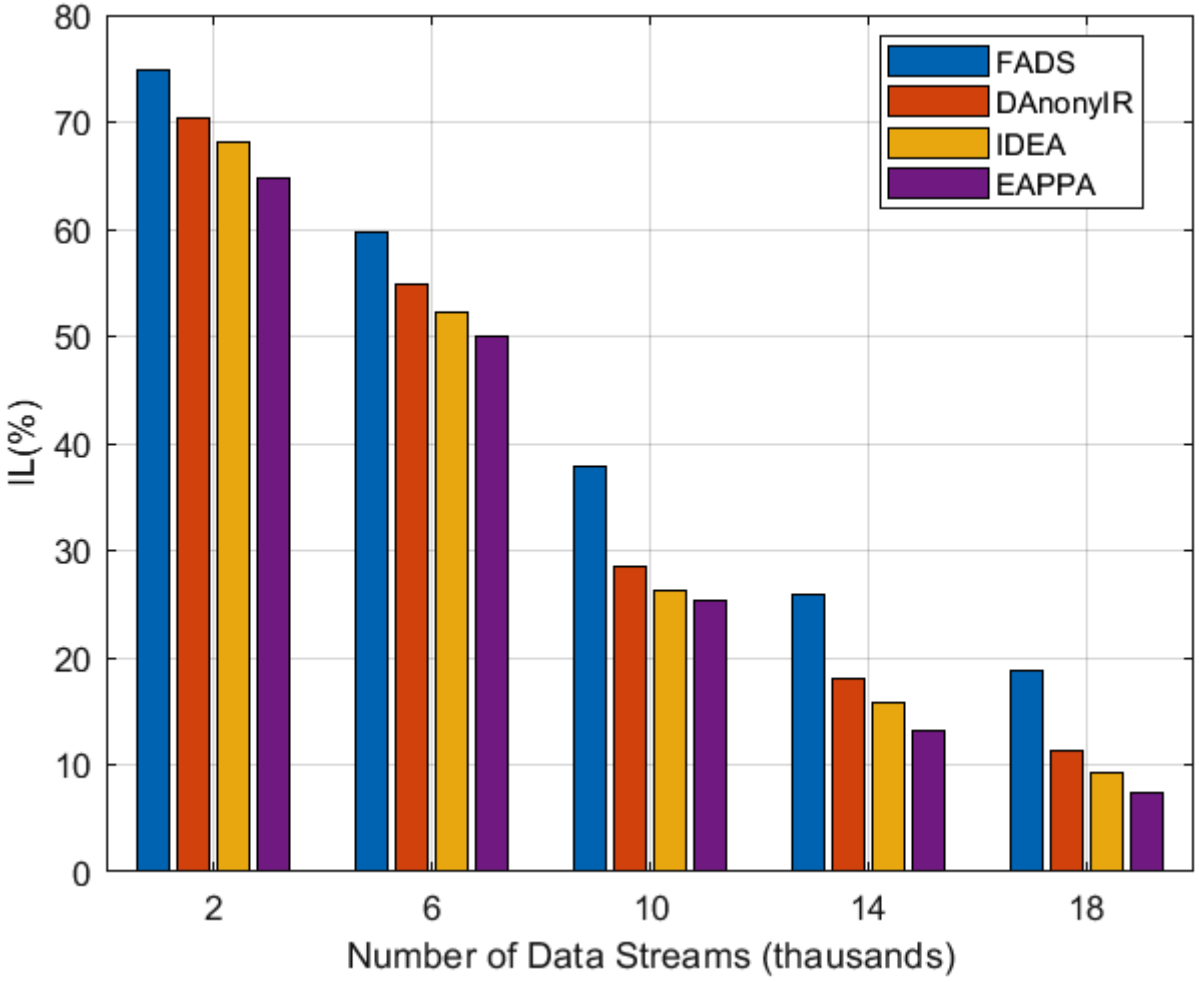

Figure 7

IL performance investigation with varying stream data size

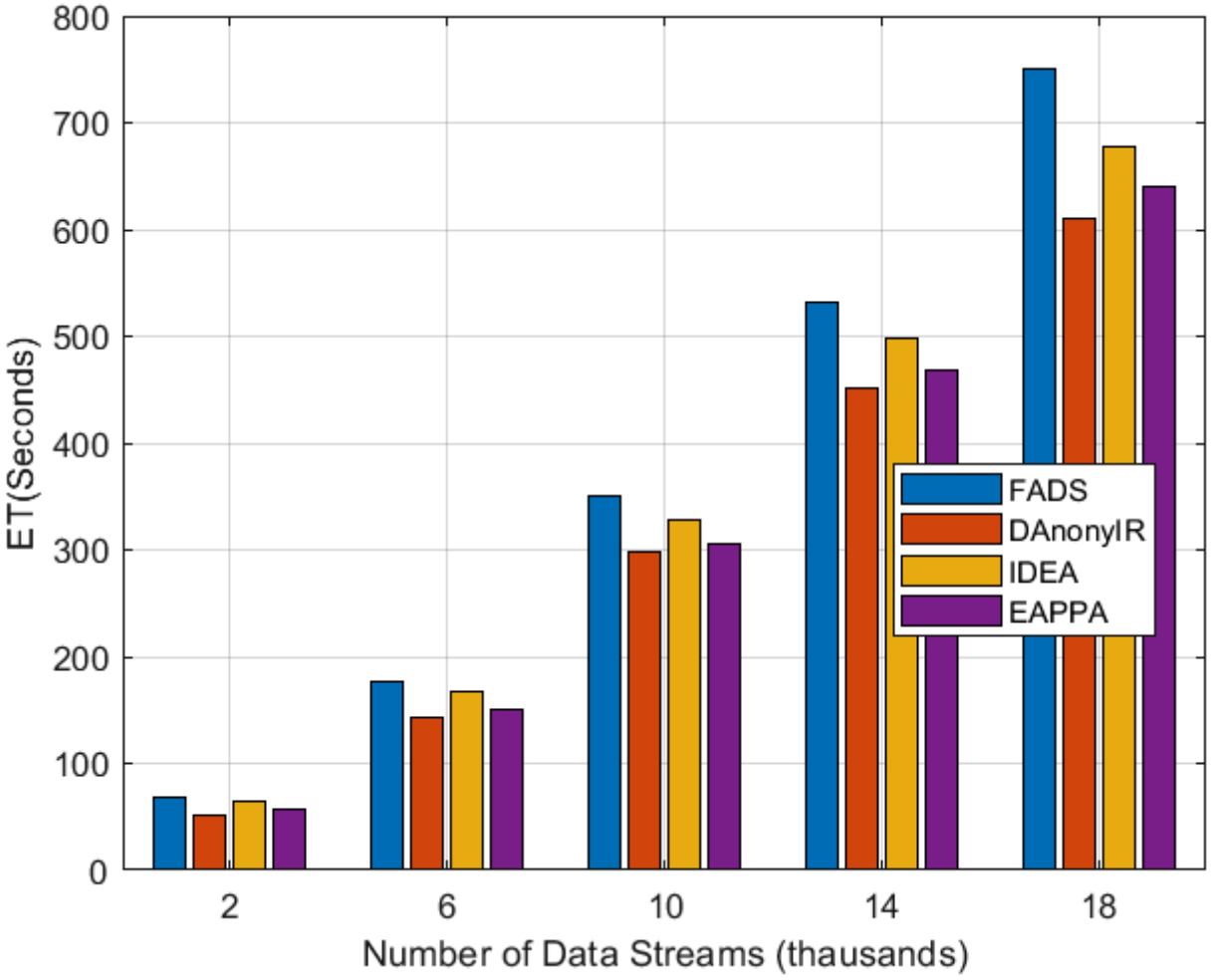

Figure 8

ET performance investigation with varying stream data size