Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

An In-Depth Comparative Study of Adaptive k-Anonymity Methods for Streaming Data Privacy

Rinkalben J. Prajapati[1], Jaykumar Shantilal Patel[2]

[1]Research Scholar, Gujarat Technological University, Gujarat, India.
[2]Chaudhari Technical Institute, Gandhinagar, Gujarat, India.

Email: [1]rinkal10feb@gmail.com, [2]jay_sp_mca@yahoo.co.in

Abstract

The real-time data is growing extensively due to the immense use of numerous web applications, IoT devices, social media, and network-based applications. This online streaming data, characterized by its volume and velocity, is expressed as big data. While it is accessible for business analytics and research purposes, it can often sacrifice individual privacy. There are different traditional approaches, such as k-anonymity, l-diversity, and t-closeness, that exist to safeguard individual privacy by making each data record indistinguishable from at least k other records. The dynamic nature of real-time stream data makes these methods difficult to apply. However, various research shows that modifications to these methods can effectively protect individual privacy in streaming data. This paper presents a comprehensive review of k-anonymity-based techniques that adapt sliding window models, clustering approaches, and other variations to efficiently protect data privacy while maintaining k-anonymity without compromising data utility. The review discusses the challenges faced in protecting stream data privacy and concludes with research directions to enhance these methods for adaptive and scalable privacy-preserving mechanisms for streaming data.

Keywords: k-Anonymity, Streaming Data Privacy, Real-Time Data Anonymization, Cluster-based Anonymization, Data Utility.

1. Introduction

The real-time data volume is increasing day by day with the extensive use of web applications, Internet of Things (IoT) devices, social media platforms, and networked systems. This surge in continuously produced data, often known as streaming data, possesses the fundamental properties of big data, such as volume, velocity, and variety. The availability of vast amounts of streaming data presents numerous opportunities in research and analytics for gaining insights and making informed decisions with the help of artificial intelligence, machine learning, and deep learning tools. However, it also poses major concerns about individual privacy, as sensitive information may be vulnerable due to improper handling and inadequate privacy mechanisms. To address privacy concerns, the differential privacy framework [1], alone or with perturbation methods [3], condensation techniques [2], randomization techniques, and anonymization-based privacy have been explored in the literature. Differential privacy can be achieved by adding noise or by modifying the original data values with transformed values before processing. This simple and fast method masks sensitive data by

DOI: https://doi.org/10.36548/jtcsst.2025.3.008
Received: 13.07.2025, received in revised form: 11.08.2025, accepted: 26.08.2025, published: 05.09.2025

adding mathematically calibrated noise to query results and preserves data distribution for mining. However, it sacrifices accuracy if the noise level is high. It suffers from the challenge of parameter tuning, affecting privacy and accuracy [8]. Non-perturbation-based techniques such as k-anonymity do not distort the actual data but focus on anonymizing it to prevent identity disclosure. This approach applies suppression or generalization to quasi-identifiers to preserve privacy, with high overhead for real-time data streams. Differential privacy has proven better in terms of privacy protection against background knowledge and homogeneity attacks, but k-anonymity provides better data utility and accuracy. k-anonymity [4][5][6][7] can be strengthened against background knowledge and homogeneity attacks by l-diversity [9] and t-closeness [10], respectively. k-anonymity was proven stronger than Datafly [37] by providing less distorted data and μ-argus [38] by better protection. k-anonymity-based methods have been effective for static datasets, but implementing these methods for streaming data is challenging because of its constantly changing and dynamic nature. Traditional anonymization techniques are designed for static datasets and operate offline. However, data streams are dynamic datasets for which anonymization algorithms need to operate online [11]. IoT applications have expanded in various domains such as smart homes and office systems, working with devices like smart cars, fire alarms, and security cameras; wearable devices for health monitoring; and smart city applications. Consider an IoT-based patient monitoring healthcare system that uses IoT devices, such as smartwatches, ECG implants, or home blood pressure machines, to read health-related records to the cloud system frequently and share them with doctors. These records from the cloud are anonymized and shared for analytics to enhance the prediction model and take necessary action to reduce health-related risk. It may be possible that the same patient's record is repeated many times in a day, or the number of incoming patient records varies in frequency, as the number of incoming records is higher during the day than at night. If the k-anonymity algorithm uses a fixed and lower value of k, then incoming records in the morning, which are more frequent, may be processed in less time, potentially compromising privacy. If the k value is set too high, it allows enough time to anonymize the records but delays their release. If a lower value of k is considered at night, it is sufficient to anonymize the records because fewer records are generated. In this situation, a larger value of k leads to more delay. In this context, the value of k should be adjustable to make k-anonymity adaptive so that patient information can be protected and real-time information is also available to doctors. To effectively handle and protect streaming data, it is necessary to use k-anonymity techniques that can manage continuous data streams while providing privacy, maintaining utility and handling the data with proper delay. Recent research has explored modifications and extensions of traditional k-anonymity techniques to address the challenges of streaming data. Some approaches include clustering-based techniques [12] to create equivalence classes of size k from records by considering numeric attributes [20][28] and categorical attributes. Sliding window based models try to handle and process the incoming stream data before its expiration [30]. Approaches such FADS [21], FAST [22], K-VARP [24] and UBDSA [29] are using time based sliding window, while FAANST [20], IDEA [23], CUDSA [31] uses count based sliding window models. Recent research on adaptive k-anonymity such as UBDSA [29], X-BAND [30], CUDSA [31], SUHDSA [32], and EPPAA [33] reduces information loss and average delay by using dynamic clustering and partition reuse. However, Most of the methods rely on generalization, while the importance of suppression is widely ignored. When a record cannot be grouped to ensure k-anonymity, it is suppressed which directly increases information loss and reduces data utility. It also lowers average delay and run time due to the removal of unmatched records from memory and affects the analytical results. There is a need to balance suppression and generalization or to adapt suppression thresholds. Some methods only partially adapt the anonymity parameter k and often ignore concept drift, leading to over generalization.

Also the use of static Generalization hierarchies prevents context-aware generalization that could minimize utility loss. The purpose of this review is to identify the gap by studying, analyzing, and comparing existing privacy protection techniques for stream data and to address the challenges of privacy protection of data in real-time environments by examining the relation between privacy and utility and exploring future directions for delay-aware, scalable, and adaptive k-anonymization.

2. Objective and Review Structure

The objective of this review paper is to provide a detailed study of existing k-anonymity-based privacy-preserving techniques for real-time streaming data. It focuses on and covers streaming data privacy preserving techniques using clustering approaches, sliding window models, and delay-aware k-anonymity frameworks. The aim is to evaluate these methods based on privacy parameters such as privacy protection, Information loss, data utility, scalability, delay and runtime including limitations. This review employs a structured narrative methodology to investigate the progression of k-anonymity-based privacy techniques for streaming data. The research was identified by searching the keywords “k-anoymity”, Stream data anonymization” “cluster based k-anonymity” in Google scholar, IEEE Explorer, Science Direct and SpringerLink. The review covers earlier models such as CASTLE [18] and FAST [22], as well as recent approaches such as SUHDSA [32], K-VARP [24], and EAPPA [33]. These models are categorized based on anonymization methods, clustering method used to create groups of size k for anonymization, evaluation metrics including information loss and delay, and application domains such as IoT and big data. The focus of the review is on techniques that enhance real-time privacy through adaptive or clustering-based k-anonymity models.

Section 3 presents the key requirements for achieving k-anonymity, focusing on the important roles of generalization and suppression in protecting data privacy. Section 4 presents an in-depth literature review, comparing and discussing various algorithms and techniques developed for stream data anonymization. This includes a comparative analysis of the methods in terms of their performance, effectiveness, and limitations, followed by a discussion of potential future research directions. Finally, Section 5 concludes the paper by summarizing the key findings of the review and outlining future prospects and research opportunities aimed at enhancing privacy preservation in streaming data environments.

3. Foundations of k-Anonymity and Clustering Algorithm for Data Privacy

Protecting individual privacy in shared datasets begins with removing explicit identifiers; however, this alone is insufficient. k-Anonymity addresses the risk of record-linkage attacks by ensuring that each record is indistinguishable from at least k–1 others based on quasi-identifiers, using techniques like generalization and suppression to form equivalence classes and reduce re-identification risk.

3.1 k-Anonymity for Stream Data Privacy Protection

The first step in protecting individuals' identities involves de-identifying datasets by removing directly identifying attributes, such as social security numbers. However, this measure alone is insufficient to ensure privacy. A record-linkage attack can re-identify

individuals by combining quasi-identifiers with publicly available data. k-Anonymity [4][5][6][7], is a well-established method to counteract record-linkage attacks, ensuring that each record in a published dataset shares the same quasi-identifiers with at least k−1 other records, thereby reducing the risk of re-identification. Some key terms helpful to understand the k-anonymity model are:

3.1.1 Explicit Identifiers

In publicly available dataset, Some attributes can directly reveal the identity of an individual. These types of attributes such as Full name, phone number, or social security number are considered as explicit identifiers. They are required to be removed before publishing data [4][5][6][7].

3.1.2 Quasi-Identifier

In a dataset, there are certain attributes that, individually, do not disclose a person's identity, but when combined with other attributes, can reveal it. For dataset D, Quasi-identifier set Q is subset of D Representing Q ={q_1,q_2,…,q_n} collection of n attributes that need to be anonymized before being shared; this collection is known as the set of quasi-identifiers (QIDs). Examples of such attributes include age, gender, or ZIP code [4][5][6][7].

3.1.3 Sensitive Attributes

In publically available dataset, Some attribute(s) contain private or confidential information, such as income, health status, or political views, considered sensitive attributes. Revealing of attributes of a specific person could cause harm or violate their privacy [4][5][6][7].

3.1.4 Equivalence Class

It represents a group of records from dataset D**,** having identical values for all quasi-identifiers within that group. This makes it challenging to distinguish between them, thereby safeguarding individual identities.

Latanya Sweeney's work on k-anonymity [4] introduced a privacy-preserving model, focusing on reducing the individual re-identification risk from publicly shared datasets. The fundamental principle of k-anonymity is to ensure that each record in a dataset is indistinguishable from at least k−1 other records with respect to specific identifying attributes, referred to as Quasi-identifiers [4]. This model protects against linkage attacks by ensuring that individuals cannot be uniquely identified by linking attacks if an adversary attempts to re-identify individuals by correlating data with publicly available information [9]. Authors [4] demonstrated that anonymized datasets, such as medical records, remain vulnerable to re-identification through such linkages. The k-anonymity [4][5][6][7] framework addresses this weakness and applies generalization and suppression techniques to records within equivalence classes of at least k individuals to reduce the likelihood of re-identification while preserving data utility for analysis.

3.2 Role of Generalization and Suppression in Achieving k-Anonymity

Generalization is a data transformation technique used in the k-anonymity framework for privacy-preserving data publishing. It reduces the granularity of data by replacing attribute values with more abstract values so that individual records cannot be easily re-identified.

3.2.1 Generalization

It transforms specific attribute values into more general values. It Makes records indistinguishable by reducing precision while maintaining utility.

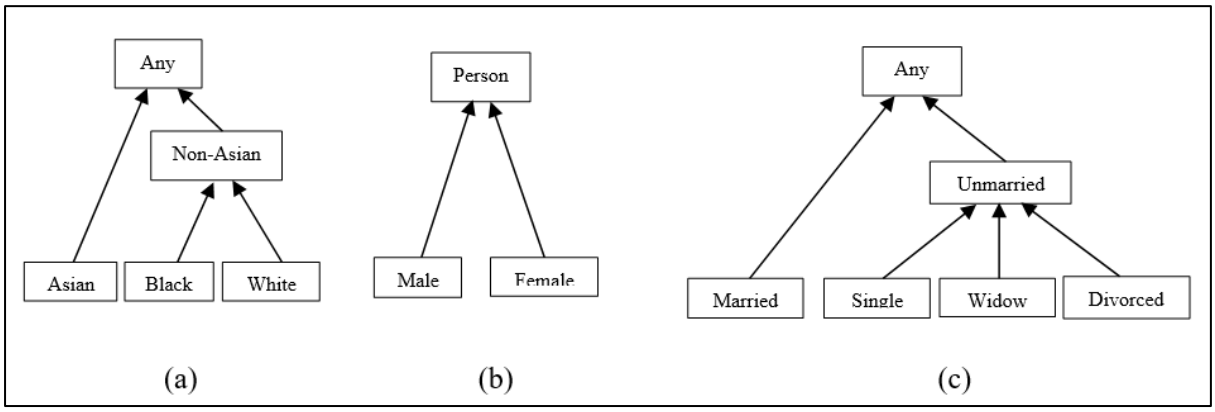

Figure 1. Generalization Hierarchy (a) Race (b) Gender and (c) Marital Status Attribute

Generalization hierarchies (GHs), sometimes referred to as Domain Generalization Hierarchies (DGH) [4], are fundamental elements in privacy-preserving data publishing methodologies, including k-anonymity [4][5][6][7], l-diversity [9], and t-closeness [10]. They offer a structured method for converting specific values of quasi-identifier (QI) attributes into more abstract categories. This abstraction reduces data granularity to safeguard individuals' privacy while maintaining the dataset's overall analytical utility. A generalization hierarchy describes multiple abstraction levels for categorical attributes. Consider Figure 1, which represents the hierarchy for three attributes: Race, Gender, and Marital Status. In Figure 1(a), the Race values Asian, Black, and White can be generalized to the more abstract value Any. Similarly, the Gender values Male and Female are generalized to Person in Figure 1(b). The Marital Status attribute has a two-level Generalization Hierarchy in which the values Single, Widow, and Divorced can be grouped under Unmarried at level 1 and further generalized to Any at level 2, as per Figure 1(c). These transformations reduce the risk of re-identifying sensitive data. A generalization hierarchy helps simplify the generalization process and also plays an essential role in distance calculation during the generation of equivalence classes for clustering-based anonymization [13]. Clustering algorithms supporting mixed-type attributes use generalization hierarchies to create clusters of similar-type values and utilize these hierarchies to measure the semantic similarity between categorical values. In generalization, values having a common ancestor in the GH are considered more similar than those diverging earlier, allowing for clustering with minimal information loss. For streaming data privacy preservation, to meet privacy constraints, GHs provide the opportunity for dynamic

generalization by identifying the lowest common ancestor across attribute values within a cluster. The generalization approach of transforming concrete values to abstract values allows anonymization techniques to maintain a balance between data utility and individual privacy. It permits attributes to be generalized incrementally to minimize information loss while ensuring that individual records cannot be uniquely identified within the dataset.

3.2.2 Suppression

Suppression removes or hides data values entirely, either partially or fully. When generalization alone isn't enough to create an anonymous group by generalizing attribute values, suppression is used to eliminate outliers or unique values. Suppression can be applied at the record level by removing the whole record or at the cell level by removing a particular value or replacing it with '*'. It is particularly useful for high-risk records that could lead to identification. Suppression can apply to individual cells or entire records. It eliminates uniquely identifying values or removes data to enforce anonymity [4][5][6][7].

3.3 Stream Data and Obstacles Encountered in Safeguarding its Privacy

A data stream is a continuous and ordered sequence of data elements that are received over time and are typically required to be processed in real-time or near real-time, under constraints such as limited memory, computational resources, and latency [22]. For stream data k-anonymization, latency can be measured as the delay, the time duration between the arrivals of records in a stream and the publication of the stream after anonymization. Latency can be referred to as average delay [22][24][29] or data aging [21]. For privacy preservation, a data stream refers to a continuous flow of structured or semi-structured records that may contain sensitive information. The dynamic and evolving nature of data such as velocity and volume makes traditional batch-oriented privacy preservation mechanisms difficult to apply. This real time data must be dynamically anonymize prior to its publication or analysis to ensure privacy. Stream data privacy preservation techniques face several challenges [14]:

- Data must be processed in real time, often in a single pass or within a predefined time limit or delay, because they are continuous and have infinite data size.
- There is limited memory and processing time, requiring efficient anonymization techniques.
- Handling concept drift as the patterns or behaviors of data keep changing over time.

To address these challenges, several privacy-preserving data stream mining (PPDSM) techniques used for static datasets, such as Perturbation Techniques [2][3], Differential Privacy [1], and k-anonymity [4][5][6][7]-based techniques, can be used with real-time adaptation for streaming data through approaches such as sliding windows, cluster-based anonymization, micro-aggregation, and online generalization. To imitate real-world continuous input conditions, the simulation of streaming datasets plays a crucial role in the evaluation of the performance of data stream processing algorithms. Real streaming data are continuous and asynchronous, as they arrive from live sources such as network sockets, APIs, IoT devices, or transaction systems. Sometimes researchers need to rely on synthetic or practically simulated streams to represent the statistical properties, dynamic nature, and variability of actual data sources, as real-time data from working environments is not easily available due to domain-specific restrictions, privacy constraints, and resource availability. This type of simulation generates incremental data for unpredictable real-world scenarios with time bounds, concept

drift, and noise patterns from batch datasets. For privacy protection algorithms, this simulation involves reading a domain-specific dataset record by record and feeding it sequentially through count-based sliding windows or time-based sliding windows to retain data for a fixed duration. A time constraint is applied to ensure timely publication after anonymization. SWAF [15] and SKY [27] generate synthetic streams from static dataset records at controlled arrival rates. CUDSA [31], UBDSA [29], K-VARP [24], FADS [21], and FAST [22] simulate the streaming data by feeding the records through the sliding window to keep recent records and limit memory usage. FAANST [20], CASTLE [18], and B-CASTLE [19] generate domain-specific streams by simulating realistically distributed records while including delay constraints to reflect real-time publishing requirements. For stream data k-anonymity, concept drift refers to the gradual or sudden change in the statistical distribution of quasi-identifiers within the incoming stream. Such changes in distribution can degrade the performance of anonymization, leading to over-generalization or delayed data release. UBDSA [29] and K-VARP [24] identify concept drift when newly arrived records in a sliding window are significantly different from earlier records, requiring dynamic adaptation of k-anonymization parameters, cluster formation, and generalization hierarchies to maintain both privacy guarantees and data utility in evolving streams. CASTLE [18] merges and splits the clusters, KIDS [16] uses density-aware clustering, UBDSA [29] re-clusters by refreshing the cluster within the sliding window when the incoming records do not match with previously formed clusters, and SUHDSA [32] updates clusters continuously to adapt to concept drift. K-VARP [24] adapts to changes in stream data by splitting and merging clusters based on changes in the frequency of QI values to minimize information loss. Sudden changes in stream data can also be adapted by changing the value of k; for infrequent arrival of records, a lower value of k generates smaller groups to decrease the value of delay, while a higher value of k helps to form larger groups to provide strong privacy when records arrive frequently. k-anonymity can adapt to changes in stream data by modifying the generalization hierarchies or by using different levels of generalization. CASTLE [18] applies incremental re-generalization, K-VARP [24] applies selective generalization on attributes, and UBDSA [29] generalizes the QI attributes level-wise depending on the current distribution of QI values within the active window, while SUHDSA [32] performs secure and low-loss real-time adjustments.

3.4 Role of Clustering Algorithm in Privacy Protection

The fundamental concept of k-anonymity [4][5][6][7] is to ensure that each entry in a dataset cannot be distinguished from at least k−1 other entries based on certain identifying attributes, resulting in the creation of groups based on the similarity of quasi-identifiers. Clustering algorithms play a vital role in achieving this goal. These algorithms organize similar data points into clusters, with each cluster needing to contain at least k entries to make any single entry indistinguishable from at least k-1 others. This approach prevents re-identification by anonymizing the data within each cluster. To attain k-anonymity [4][5][6][7], it may required to apply generalization [4][5][6][7] or direct suppression [4][5][6][7], resulting in considerable information loss (IL). Clustering group's similar entries, allowing for more precise data representation within clusters and reducing information loss [13]. Some clustering algorithms can dynamically adjust the level of generalization based on cluster characteristics, balancing privacy and data utility. Byun et al. introduced the application of clustering techniques, specifically a greedy clustering heuristic, to achieve k-anonymity via k-means clustering [13]. Let us illustrate how clustering can facilitate k-anonymity through a simplified example.

Table 1. Personal Health Information of Individuals

ID	Race	Birth Date	Gender	Zip	Marital Status	Disease
	Asian	64/04/11	F	94142	Divorced	Hypertension
	Asian	64/09/23	F	94141	Divorced	Obesity
	Asian	63/03/13	M	94139	Married	Obesity
	Asian	63/03/18	M	94139	Married	Short breath
	Black	64/09/27	F	94138	Single	Short breath
	White	64/09/27	F	94139	Single	Chest pain
	Black	64/09/27	F	94141	Widow	Short breath

Consider a Table 1 that contains personal health information of individuals. To protect this information from direct identification attributes like Personal Identification number or Person name are removed while other information such as Birth date, Gender, Zip, Marital Status and Disease are kept as it is. To protect sensitive information initially, k-means clustering with number of clusters=3 is applied to create a group of records with size =2 to achieve 2-anonymity. Based on the similarity of Quasi-identifier attributes such as Race, birth date, Gender, zip code, and Marital Status, records were initially grouped into three clusters using k-means clustering as shown in Table 2. Cluster 0 groups' male individuals, born in the same year, sharing a common zip value "94139" and marital status. Cluster 1 contains records of females with identical birthdates, located in regions with zip codes starting with "941", and shows some variation in marital status. Cluster 2 consists of females born in 1964, residing in regions with similar zip codes, and mostly having similar marital statuses.

Table 2. Record Assigned to Clusters Using k-Means Clustering (no. of Clutters=3)

Race	Birth Date	Sex	Zip	Marital Status	Disease	Cluster
Asian	63/03/13	M	94139	Married	Obesity	0
Asian	63/03/18	M	94139	Married	Short breath	0
White	64/09/27	F	94139	Single	Chest pain	1
Black	64/09/27	F	94141	Widow	Short breath	1
Asian	64/04/12	F	94142	Divorced	Hypertension	2
Asian	64/09/13	F	94141	Divorced	Obesity	2
Black	64/09/27	F	94138	Single	Short breath	2

Table 3. Personal Health Information of Individuals Satisfying 2-Anonymity

Race	Birth Date	Sex	Zip	Marital Status	Disease	Cluster
Asian	1963	M	94139	Married	Obesity	0
Asian	1963	M	94139	Married	Short Breath	0
Non-Asian	64/09/27	F	941**	Un-married	Chest Pain	1
Non-Asian	64/09/27	F	941**	Un-married	Short Breath	1
Any	1964	F	941**	Un-married	Hyper-Tension	2
Any	1964	F	941**	Un-married	Obesity	2
Any	1964	F	941**	Un-married	Short Breath	2

Once the records are clustered, the data can be generalized to make individuals within each cluster indistinguishable from others, as shown in Table 3, according to the generalization hierarchy of the respective QI attribute by replacing the original value with a more abstract value. Here, for the Race attribute, the values "White" and "Black" are generalized to “Non-Asian” in cluster 1; birth dates were reduced to the birth year "1963” in cluster 0 and “1964” in cluster 3. Values of the Zip attribute were masked by replacing the last digit with a wildcard "9414*". Sensitive attribute values, such as disease, are left unchanged to preserve medical relevance; others were generalized to protect privacy. This process ensures 2-anonymity, meaning that each person's information is shared by at least two others, making re-identification difficult while still allowing useful analysis of the data. Figure 2 represents the basic stream data k-anonymity framework to protect the privacy of stream data. It works in four phases: preprocessing, clustering, k-anonymization, and publishing. The preprocessing phase initializes streaming parameters such as sliding window size, delay, cluster size, quasi-identifiers, sensitive attributes, and other input measures. The clustering phase groups records or creates partitions of size k. The anonymization phase anonymizes the records by generalizing quasi-attributes using the generalization hierarchy and applies suppression if required, as per the developed approach. k-anonymized stream data are publicly published to applications for analytics. Different research enhances it according to the objective of the research. In past research, several studies have been conducted to protect individual privacy using the k-anonymity approach and its variations. To achieve this, different clustering algorithms have been used to create equivalence classes of size k.

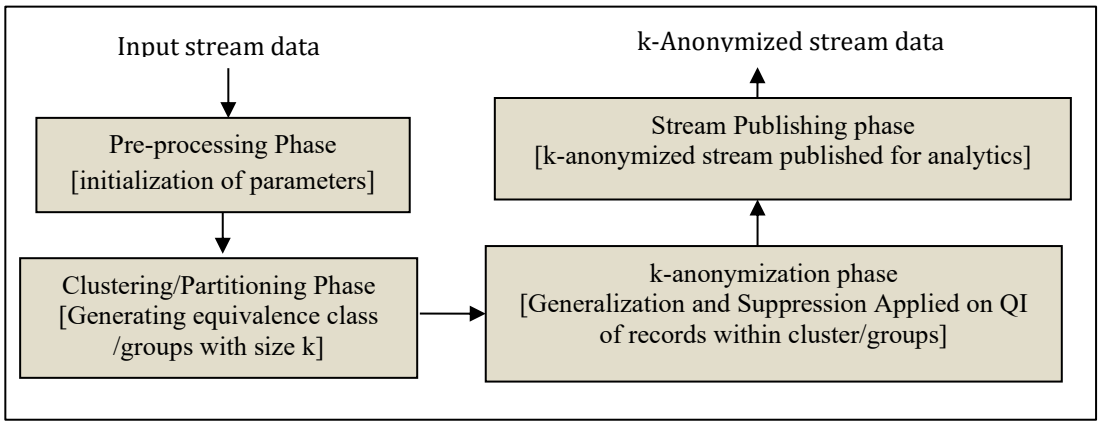

Figure 2. Basic Framework for Stream Data k-Anonymization

The next section presents a detailed literature review of different privacy-preservation frameworks/approaches for streaming data using clustering algorithms.

4. Literature Review

This section contributes a comprehensive study of existing privacy-preserving techniques for streaming data. It includes various approaches and their operational characteristics by highlighting their strengths, limitations, and contributions to streaming data privacy preservation. It includes Clustering-Based Models used for similarity-based grouping to achieve k-anonymity while minimizing information loss; Sliding Window-Based Approaches to anonymize data within bounded time or count windows to support real-time constraints. and Delay-Aware and Utility-Driven Techniques that incorporate latency constraints and utility metrics to balance privacy with data usability. These approaches cannot be specifically categoried, as they can employ combinations of clustering algorithms, sliding window models, and concepts of delay awareness and utility to ensure the privacy of streaming data.

Author Byun, et.al.,in[11] addressed the limitations of traditional k-anonymity and ℓ-diversity methods and proposed an efficient methodology to anonymize continuously expanding datasets while simultaneously preventing inference attacks. This methodology aims to reduce the risk of vulnerabilities that emerge due to the release of multiple anonymized versions of a dataset over time, enabling adversaries to deduce sensitive information by comparing changes across these releases. Author proposed a framework that can securely integrate newly added records with previously anonymized data reducing computational overhead and maintaining consistent equivalence classes to prevent pattern recognition. This approach enhances privacy protection along with a balance of data utility and efficiency by avoiding the unnecessary re-anonymization of the entire dataset. It can be considered a foundational strategy for privacy-preserving incremental data publishing [11], significantly enhancing both security and usability in dynamic database environments.

Wang et al. to propose the Sliding Window Anonymization Framework (SWAF) [15]to overcome the limitations of traditional k-anonymity. It combines Specialized K-anonymization (SK) for initial processing with Incremental K-anonymization (IK) for continuous updates and supports both count-based and time-based sliding windows. Experimental results on the Adult and Jeff Corporation Sales Transactions datasets show low information loss and fast processing and scalability. However, for higher values of k and larger window sizes it increases computational cost. Fung et al. addressed privacy risks that arise from linking multiple anonymized data releases [2][7] and Their method in [17] maintained k-anonymity across sequential publications by reducing information loss compared to traditional approaches while protecting against correspondence attacks. However, it does not address the risks of attribute disclosure that may emerge over time. Li et al. introduced Stream k-anonymity (SKY) [27], designed for high-speed streams with strict delay constraints and immediate processing of incoming records. SKY employs a specialization tree to dynamically generalize quasi-identifiers and introduces the Information loss metric LM to measure the trade-off between privacy and utility. It achieved efficient anonymization with reduced information loss proving its applicability to domains such as market transactions and telecommunications.

The CASTLE: Continuously Anonymizing Data Streams [18] by Cao et al. introduces an innovative framework focusing the challenge of real-time anonymization of streaming data. It operates within a dynamic environment, performing continuous k-anonymization over

sliding windows by applying tuple-level generalization to support both time-based and count-based windows. It optimizes generalization strategies to minimize IL and performs utility driven anoymization. Performance of CASTLE was measured on the UCI Adult dataset to evaluate parameters such as information loss, processing time, and tuple delay. Results proved CASTLE's scalability and efficiency in minimizing IL across various workload and privacy settings. B-CASTLE [19], was advancement of CASTLE [18] by Wang, Lu, et al., overcomes limitations such as significant information loss and inefficiencies of simultaneous re-clustering of all tuples. It optimizes CASTLE [18] by dynamically reallocating tuples into clusters based on their distribution, ensuring balanced cluster formation by selectively merging only the most pertinent clusters during publication, It improves data utility and handles necessary delay constraints while maintaining k-anonymity. Experimental results shows that B-CASTLE is scalable for high-speed streaming applications and outperforms CASTLE [18] in terms of achieving superior cluster distribution, reducing information loss, and enhancing efficiency. Zakerzadeh et al. also addressed the limitations of CASTLE [18] and developed a cluster based Fast Anonymizing Algorithm for Numerical Streaming Data (FAANST) [20] to anonymizes numerical streaming data. It was executed on both synthetic and real-world numerical datasets and results confirms improvement over CASTLE [18] by reducing data loss, improving efficiency, and limiting tuple suppression. It suffers from reduced data utility at the cost of higher runtime and more suppressed tuples. Its application is restricted by not supporting categorical attributes and by allowing some tuples to remain in the system longer than desired.

Zhang, J. et. al., proposed KIDS [16], a dynamic sliding windowbased anonymization framework that clusters incoming numerical data and ensures compliance with k-anonymity by utilizing a Top-Down Specialization (TDS) Tree to process streaming data dynamically. Results are compared with CASTLE [18] and FAANST [20] for performance metrics such as data loss, execution time and cluster quality and it shows lower data loss, faster execution time and More balanced cluster formation, improving anonymization without unnecessary suppression. However, Computational complexity increases with larger window sizes and it does not support more advanced privacy models like ℓ-diversity. Guo and Zhang proposed FADS, a fast clustering-based anonymization [21] for data stream to efficiently process and anonymize tuples while ensuring low IL and reduced running time compared to CASTLE [18] and FAANST [20]. It reads incoming tuples, adds them to the clusters, and publishes them when no more tuples arrive. It reuses previously anonymized tuples to meet the constraints if k-1 tuples are unavailable forpublication. Although FADS improves efficiency and data utility, it suffers from early tuple publication for newly arrived tuple is published prematurely because it serves as one of the k–1 nearest neighbours of a waiting tuple, potentially increasing information loss. Mohammadian et al. introduced FAST [22], a parallel anonymization algorithm designed for big data streams. To address the early tuple publication issue in FADS [21], It incorporates a proactive heuristic that estimates round time, ensuring that data are published before reaching a predefined expiration time. Experimental comparisons with FADS [21] demonstrate that FAST is both efficient and effective in anonymizing big data streams, achieving lower information loss and reduced cost metrics across various parameters.

Authors Yang et al., present a novel approach, the Incomplete Data Stream Enhancement Algorithm (IDEA) [23] for handling missing or incomplete data in streaming environments. It integrates advanced data imputation techniques with a utility-aware mechanism, ensuring that the imputed values are both statistically accurate and practically useful for downstream decision-making tasks. It incorporates a utility function that optimizes the selection of imputed values, thereby enhancing the overall effectiveness of real-time analytics. It is evaluated against benchmarked traditional imputation methods, including mean

imputation, KNN-based imputation, and regression-based approaches demonstrating its superior adaptability in dynamic streaming environments. Experimental results indicate that it outperforms others in maintaining data accuracy while preserving analytical utility. However, the computational complexity of an effective utility function presents challenges in resource-constrained environments. Future research can focus on enhancing computational efficiency and designing automated utility functions to improve scalability and applicability across diverse domains.

Otgonbayar et al. introduce a Framework K-VARP [24] to ensure k-anonymity in varied and dynamic data streams generated by Internet of Things (IoT) devices. It addresses the challenges of highly dynamic data streams with diverse and evolving structures. It identified and grouped incoming data streams based on their schema and performed k-anonymization on each group using generalization and suppression techniques, via suitable buffering. It maintains data utility by adjusting the level of generalization dynamically. The study utilized a combination of synthetic and real-world IoT datasets and evaluated for privacy preservation, information loss, latency/throughput, and data utility. The K-VARP algorithm exhibited less information loss compared to FADS [21] and maintained high data utility due to intelligent partitioning and generalization. It is scalable for high-speed data and adapts well to real-time stream changes. Authors Zhou, B., et.al. Proposed a novel privacy-preserving data publishing method for continuous data stream [26] that considers both the distribution of data entries and the statistical distribution of data streams by integrating anonymization techniques with real-time data publishing. It balances privacy and data utility by adapting data stream distribution and prevents re-identification attacks. Experiments on real and synthetic datasets validate the effectiveness and efficiency of the proposed method.

Zakerzadeh, H., and Osborn, S. L. proposed a Delay-sensitive approaches for anonymizing numerical streaming data [28]. It performs dynamic clustering of incoming numerical data and satisfies anonymity constraints within a strict time frame. Experiments were performed on datasets from the UCI Machine Learning Repository to evaluate performance metrics such as data loss, execution time, and cluster quality under varying conditions. It demonstrates superior performance over CASTLE [18] and FAANST [20] by achieving lower processing delays and improved data retention. However, strict privacy requirements may result in a reduction of data utility.

The X-BAND [30] is designed by Otgonbayar et al., introducing a new mechanism called the expiration-band for stream data anonymization by allowing multiple scans of expired data tuples to find the best cluster with the least information loss. It works based on a weighted distance function using K-Nearest Neighbor (KNN) to minimize missing data by considering the similarity of quasi-identifiers (QIDs) and attribute distances. It also stores un-anonymized and expired data tuples temporarily and provides another chance for inclusion without disrupting the data stream's order. Performance of X-BAND was compared with FADS [21], achieving lower information loss by 5–11% for Adult data and 1–3% for PM2.5. X-BAND proves its effectiveness and efficiency by anonymizing varied data streams with better data utility and by handling the challenge of missingness prediction and adaptive distance metrics.

Sopaoglu and Abul introduce the Utility-Based Data Stream Anonymization (UBDSA) algorithm [29] to enhance overall data utility by balancing data quality and data aging in data stream anonymization and to minimize both average delay and information loss (IL). They introduce a novel cluster assignment distance metric, Cardinality Aware Information Loss (CAIL), to evaluate clusters based on data similarity to ensure sufficient record similarity to preserve utility after anonymization. It also measures cluster size by accounting for the number

of records in a cluster, as larger clusters may lead to increased IL due to the need for more generalization, while smaller clusters might reduce processing delay by compromising privacy requirements. By integrating these factors, it effectively balances IL and processing delay. Experimental results show UBDSA has better balanced performance for average delay and information loss compared to CASTLE [18] and FADS [21]. Authors Sopaoglu and Abul developed Classification Utility Aware Data Stream Anonymization (CUDSA) [31] to protect sensitive attributes by reducing the risk of attribute-linkage attacks, aiming to maximize downstream classification accuracy of the anonymized data while satisfying k-anonymity. They created an optimization method to minimize information loss, maximize classification accuracy, and enhance diversity in sensitive attributes by providing a facility for users to adjust various parameters for IL, accuracy, and window size. It processes a data stream with a predefined delay limit, forming clusters with at least k records by storing incoming tuples. Experimental results evaluate CASTLE [18], FADS [21], and CUDSA for k-anonymity, accuracy, and information loss. All achieve k-anonymity; however, only CASTLE [18] with ℓ-diversity formally protects against attribute-linkage, CUDSA encourages diversity of sensitive attributes through entropy, and FADS [21] does not specifically address it. FADS [21] yields the lowest information loss, CASTLE [18] the highest with lower accuracy, and CUDSA strikes a balance by accepting moderate information loss to achieve higher classification accuracy.

The authors Joo, Y. & Kim, S. implemented SUHDSA [32] as an improvement of UBDSA [29], which anonymizes real-time data streams by enforcing k-anonymity within a delay-bound window to cluster incoming records based on quasi-identifiers (QI) to minimize IL using pre-computed generalization trees and the CAIL metric. It dynamically adjusts the delay threshold to balance data utility and latency, publishing clusters once the oldest buffered record reaches the delay limit. It improves performance by separating QI and non-QI attributes to reduce computation, pre-computing generalization hierarchies for faster IL calculations, clustering records based on shared ancestors to lower IL, and skipping unnecessary cluster splits to enhance performance without compromising privacy. Experimental results show that SUHDSA outperforms UBDSA [29], achieving faster runtimes of 24–30 seconds and lower information loss of 14%–77% under identical conditions. A. Sadeghi-Nasab et.al., proposed framework integrating a novel clustering approach with Apache Flink for real-time stream data anonymization [25], Data is clustered with size k within time windows, window size, and expiration limit. Categorical data is grouped via Domain Generalization Hierarchy and numerical data is binned using a frequency-based method to reduce overhead and improve performance. Clusters that meet the K-threshold are anonymized and published; others are suppressed or merged at the end to minimize IL. It is evaluated against CASTLE [18], FADS [21] and UBDSA [29] and outperforms by achieving 5.68–18.26% IL and 12.33–66.62% less data delay.

Rahul.A.P and Pramod.D.P have proposed a comprehensive framework EAPPA-Efficient Approximation and Privacy Preservation Algorithms [33] for real-time data stream privacy preservation, addressing key challenges such as redundant data and timely anonymization. It integrates two core phases: the Data Approximation and Preprocessing phase, which employs the Flajolet-Martin (FM) algorithm for efficient duplicate elimination and Natural Language Processing techniques for data cleaning; and the Adaptive Clustering and Privacy Preservation phase, which applies k-means clustering followed by adaptive refinement to ensure k-anonymity and l-diversity using entropy-based evaluation. It evaluated using the UCI Adult dataset and compared FADS [21], DAnonyIR[34], and IDEA [23]. It gives significant improvements in reducing IL, enhancing Degree of Anonymization, lowering

Execution Time and memory usage. It offers strong performance and adaptability to real-world dynamic data, its effectiveness depends on the quality of NLP preprocessing and parameter tuning for clustering, making it a notable contribution with some operational dependencies. The paper [39] proposes an improved k-anonymity algorithm by considering the trade-off between privacy protection and data usability for large-scale datasets. Author of paper measured the IL as a weighted IL function which assigns different weights to QIs depending on their influence on sensitive attributes. To enhance cluster stability and reduce variance, they designed a hybrid approach by combining a greedy algorithm with improved 2-means clustering to initialize cluster centers using mean-center selection. Experimental evaluations show that, it achieves stronger privacy with lower information loss and better data availability compared to conventional clustering-based anonymization techniques, balancing utility for big data applications such as IoT and healthcare analytics domains. The paper [40] presents a novel framework for privacy-preserving anonymization of continuous big data streams by using in-memory computing on Apache Spark. It uses a one-time clustering strategy to create optimal clusters in single pass and reuse them for incoming stream data to reduce high computational cost of repeated clustering while still ensuring k-anonymity. It parallelizes the anonymization through Spark so it became suitable for large-scale and real-time applications by achieving high throughput and efficiency. Experimental results are compared with CRUE, Mean-Shift [41], and DBSCAN [42] showing that it is consistently outperforms them by providing lower IL, better data quality, and scalability for different data sizes and value of k. Detailed comparative table summarizing the key aspects of each paper is in Table 4 and Performance comparison of Different approaches are described based on approximate results in Figure 3.

4.1 Discussion and Summary

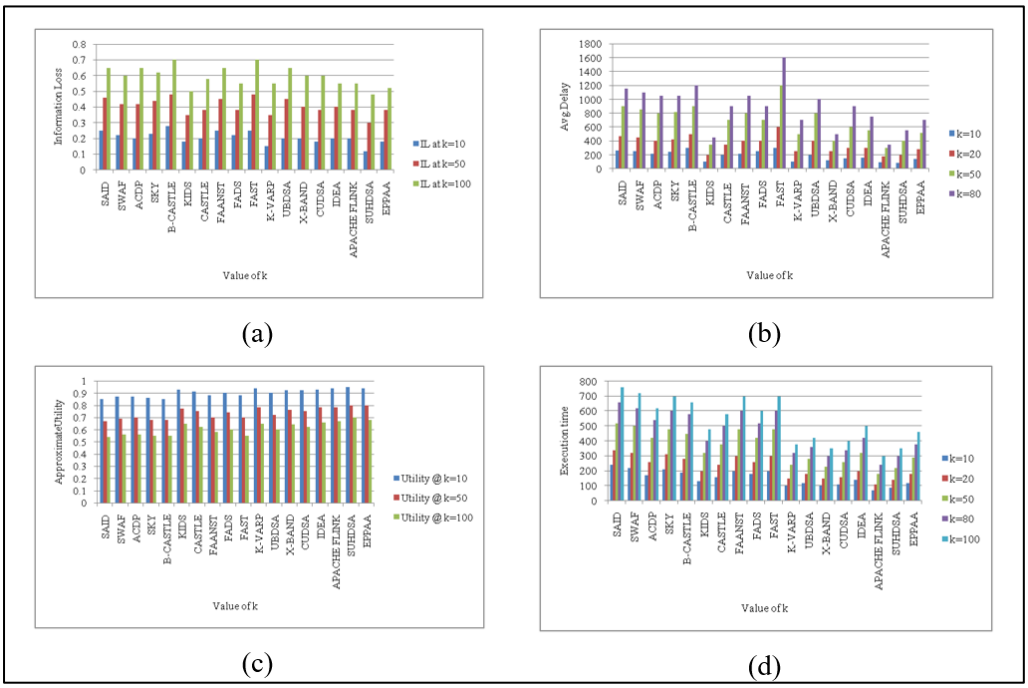

Figure 3. Comparison Chart of Value of k Parameter Against (a) Information Loss (b) Average Delay (c) Approximate Utility (d) Execution Time for Different Approaches

For stream data privacy preservation through k-anonymity and making them available to real time application such as healthcare monitoring or IoT systems for crucial analysis

Table 4. Comparative Analysis of Research Referred from Literature

Approach	Method	Performance Measure	Strength	Limitation
SWAF [15]	Frequent pattern clusters	Heavy runtime. balances privacy/utility	Preserves important patterns,	Complexity increases with pattern space,
SKY [27]	Specialization Tree with δ-constraint k-anonymity	Higher IL and weak Utility than KIDS. Delay and Run time similar to SWAF	Reduces unnecessary generalizations	May not scale well to very high-dimension or ultra-fast streams
ACDP [17]	Top-down specialization	Higher IL & Larger Execution Time	Handles incremental, continuous data	Needs static hierarchy
B-CASTLE [19]	Dynamic cluster adjustment and merging	Higher IL & Delay than CASTLE. Low Utility and High Runtime	Dynamic adjustment, lower info loss, better efficiency	experimental results influenced by data randomness
KIDS [16]	Sliding window + Top-Down Specialization	Lower IL & Delay. Utility Preserved Moderated Runtime	Effective accuracy with density handling	Initial delay, high early information loss
CASTLE [18]	Delay-constrained clustering	IL moderated & Maintains Utility, High Runtime	Strong baseline, reusable clusters	slower under large data streams
FAANST [20]	k-means (numeric only)	IL increases with Delay. High value of k Decreases Utility& High Run time	good efficiency, low data loss	Doesn't handle categorical; k-means centroid limits
FADS [21]	Nearest neighbour-based Cluster	Moderated Delay High Execution time	Low complexity, strong reuse cluster strategy reduces loss	focusing on numeric attributes, Slower due to diversity constraints;
FAST [22]	Multithreaded cluster processing (Proactive cluster + parallel threads)	High value of k increases IL and high utility loss. High Computation time	Parallelism reduces delay, efficient	Cost metric unclear uneven tuple suppression, thread balancing
K-VARP [24]	Partition based clustering with KNN	good utility	Imputation free, flexible reuse	Complex logic

UBDSA [29]	CAIL metric clustering	IL rises moderately, delay increases, Utility Decrease and linear growth in runtime with k	Tuneable delay-quality trade-off	Limited scalability insights
X-BAND [30]	KNN + Expiration Band	Similar to CUDSA.	Better reuse, less missingness	High runtime at large Γ
CUDSA [31]	Clustering-based on weighted multi-objective optimization	Slightly better IL, average delay, better utility &faster Run time than UBDSA.	Fast and scalable Handles mixed-type attributes	Requires taxonomy/ domain hierarchy
IDEA [23]	Cross-Partition + Reuse	Keeps IL low, improves utility and higher Run time for incomplete streams.	Handles missing data, compact clusters	High memory usage
APACHE FLINK [25]	Custom Similarity-based clustering + DGH trees	Excellent balance of IL, runtime and Utility. Linear Runtime	Outstanding delay performance, scalable	Parameter tuning using similarity thresholds
SUHDSA [32]	Heuristic utility-aware clustering	low IL, best Utility and Low runtime	adaptive delay control	Static generalization tree & heuristic suppression
EPPAA [33]	Generalization driven Partition based grouping	Lowest IL, High Utility and less runtime due to Partition Reuse.	Partition Reuse, Adaptive clustering and Scalable	Not adaptable to concept drift

While preserving data continuity requires records within the clusters to be generalized to make them indistinguishable from one another and reduce the use of data suppression, in k-anonymity, the value of k plays a crucial role, as it directly affects the balance between privacy protection and data utility. As per Figure 4(a), a higher value of k includes more records in the equivalence class, strengthens privacy, but results in higher information loss due to more generalization, thereby reducing utility, as shown in Figure 4(c). Execution time for different values of k is shown in Figure 4(d). For stream data, a higher value of k increases the number of records within the equivalence class, increasing the wait time of records to match the other records, which leads to an increase in delay, as shown in Figure 4(b). The value of k also affects information loss, delay, and utility for various distribution scenarios in stream data, such as imbalanced data, which leads to higher information loss, lower utility, and higher delay for higher values of k [24][29]. In contrast, uniformly distributed streams yield a steady increase in information loss and moderated delay with an increase in k [21]. A rapid bursty stream with a proper value of k results in low information loss and delay, whereas a slow rate results in higher information loss and delay [29]. In clustering, a scenario sometimes occurs when records do not naturally align with any existing cluster, raising the necessity for further generalization

without compromising data utility to match the cluster. To facilitate this, utility-aware clustering techniques can identify the cluster that incurs minimum information loss. Frameworks such as CASTLE [18] and SUHDSA [32] work on this principle to identify appropriate clusters based on information loss metrics and hierarchical generalization to balance data privacy and utility for analysis. Suppression is applied to the record when it requires extreme generalization to anonymize it, resulting in increased information loss. Suppression provides robust privacy by ensuring that no inadequately anonymized records are released, at the cost of data loss. Suppression is advantageous when a record is an outlier and distorts the cluster by needing more generalization or having a higher privacy risk. Frequent use of suppression makes the dataset biased, and sometimes rare and smaller groups are affected. In a streaming data environment, suppression should be the last choice for utility-driven generalization due to the need for real-time and complete data required to comprise a valid cluster. Numerous k-anonymity-based techniques have been adapted for streaming environments, but challenges such as handling mixed-type attributes, minimizing delay, handling concept drift, and maximizing data utility always need more attention. The literature review and comparative analysis show that modern frameworks such as K-VARP [24] and SUHDSA [32] improve efficiency and scalability. K-VARP needs more advancement to remove limitation such as handling concept drift for varied streams, high suppression of smaller or dissimilar clusters to achieve anonymization and computation overhead to calculate Jaccard similarity and R-likeness for merging merge the clusters. Also, it does not assure Minimum information loss, as it does not explore all possibilities to merge groups by heuristic-based merging. Although The SUHDSA [32] is effective, it has limitations in generalization, suppression, and clustering. It depends on fixed generalization hierarchy, which is not adaptable to evolving data and may cause high information loss with diverse or high-cardinality attributes. In terms of suppression, SUHDSA [32] also discards records by suppressing them when cluster reaches its threshold or to achieve k-anonymity, without investigating alternatives to preserve data utility. SUHDSA [32] uses a static CAIL metric to cluster the records, making it inadaptable to heterogeneous or changing data Streams.

4.2 Research Opportunities and Future Directions

Based on the literature review, numerous studies have been conducted to protect privacy via k-anonymity-based anonymization for streaming data. However, several research challenges remain unaddressed and they need further research and exploration.

Current methods primarily use fixed generalization hierarchies, which are inadequate for dynamic streaming data. Future research should explore context-aware or learning-based generalization, where hierarchies evolve with the data to preserve utility and minimize unnecessary information loss. Suppression supports robust privacy mechanisms at cost of data utility. Future research should focus on to develop k-anonymity strategies that minimize suppression or utilize it as a last resort when all utility-preserving options are not available. Streaming data often encounter changes in distribution over time and algorithms such as K-VARP encounter difficulties with these variations. Future approaches should include adaptive window, threshold tuning, and drift detection mechanisms to dynamically adjust cluster to satisfy k-anonymity in real time.

The research review identifies that some clustering techniques are limited to numerical attributes. Efficient handling of both numerical and categorical attributes remains a challenge. Future research should focus on developing time-efficient and scalable clustering algorithms that support mixed data types without compromising real-time performance. While

safeguarding privacy, there is a need for frameworks that evaluate the trade-offs between privacy and utility before applying generalization or suppression. This includes integrating multi-objective optimization or utility score thresholds into clustering and anonymization, ensuring that privacy protection does not result in analytical discrimination. Many techniques are not optimized for low-delay streaming environments. Future approaches should focus to handle low-delay data stream providing guarantee to protect privacy without sacrificing throughput or delay constraints, for real time data applications.

5. Conclusion

In the evolving environment of real-time data analytics, maintaining individuals' privacy at the expense of data utility is still one of the fundamental necessities. Traditional anonymization techniques such as k-anonymity, L-diversity, and T-closeness were originally designed to be applied to static data and are not particularly suitable for the dynamic and high-speed environment of streaming data. This systematic review investigates the application of k-anonymity in stream data settings based on the acceptance of clustering algorithms, sliding window models, and delay-sensitive models. This study was conducted through a review of underlying models for stream data privacy like CASTLE and FAST, as well as recent work like SUHDSA, K-VARP, and EAPPA. The study discusses trade-offs and improvements in preserving privacy in real-time data streams. Clustering-based techniques have great potential to find a balance between utility and privacy if applied with adaptive techniques and semantic-aware generalization. Integration of sliding window models with k-anonymity provides delay-aware anonymization for streaming data, resulting in delay-sensitive techniques such as SKY and SUHDSA, which balance latency and data quality. There are some limitations involved with these developments, such as K-VARP accommodating high suppression rates. SUHDSA relies on fixed generalization models and heuristic clustering, which limits its applicability in rapidly evolving or heterogeneous data settings. Suppression has been utilized to be beneficial in some cases to offer strong privacy while leading to data loss and potential bias. It should be used only when buffering or generalization methods are not sufficient to ensure k-anonymity. In the future, there is scope for research directions in developing a framework for maintaining privacy from streaming data that offers flexible and scalable generalization and suppression treatment and enhances responsive clustering methods that work well with mixed-type streaming data under delay. The system should also balance utility and privacy without depending on strict rules or structures and needs to be examined with benchmark datasets for privacy metrics such as information loss, data utility, scalability, suppression rate, and execution time. Generally, positive progress has been made and continues to be made in stream data privacy-preserving anonymization for streaming environments. However, by solving current problems and refining existing models, future plans will be more efficient and beneficial in protecting individual privacy while accommodating growing demands for real-time data analysis.

References

[1] Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith. "Calibrating noise to sensitivity in private data analysis." In Theory of cryptography conference, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, 265-284.

[2] Aggarwal, Charu C., and Philip S. Yu. "A condensation approach to privacy preserving data mining." In International Conference on Extending Database Technology, pp. 183-199. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[3] Agrawal, Rakesh, and Ramakrishnan Srikant. "Privacy-preserving data mining." In Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, 439-450.

[4] Sweeney, Latanya. "k-anonymity: A model for protecting privacy." International journal of uncertainty, fuzziness and knowledge-based systems 10, no. 05 (2002): 557-570.

[5] Samarati, Pierangela. "Protecting respondents identities in microdata release." IEEE transactions on Knowledge and Data Engineering 13, no. 6 (2002): 1010-1027.

[6] Samarati, Pierangela, and Latanya Sweeney. "Generalizing data to provide anonymity when disclosing information." In PODS, vol. 98, no. 188, 1998, 10-1145.

[7] Samarati, Pierangela, and Latanya Sweeney. "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression." (1998).

[8] Hewage, U. H. W. A., Roopak Sinha, and M. Asif Naeem. "Privacy-preserving data (stream) mining techniques and their impact on data mining accuracy: a systematic literature review." Artificial Intelligence Review 56, no. 9 (2023): 10427-10464.

[9] Machanavajjhala, Ashwin, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. "l-diversity: Privacy beyond k-anonymity." Acm transactions on knowledge discovery from data (tkdd) 1, no. 1 (2007): 3-es.

[10] Li, Ninghui, Tiancheng Li, and Suresh Venkatasubramanian. "t-closeness: Privacy beyond k-anonymity and l-diversity." In 2007 IEEE 23rd international conference on data engineering, IEEE, 2006, 106-115.

[11] Byun, Ji-Won, Yonglak Sohn, Elisa Bertino, and Ninghui Li. "Secure anonymization for incremental datasets." In Workshop on secure data management, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, 48-63.

[12] Zubaroğlu, Alaettin, and Volkan Atalay. "Data stream clustering: a review." Artificial Intelligence Review 54, no. 2 (2021): 1201-1236.

[13] Byun, Ji-Won, Ashish Kamra, Elisa Bertino, and Ninghui Li. "Efficient k-anonymization using clustering techniques." In International conference on database systems for advanced applications, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, 188-200.

[14] Aggarwal, Charu C., ed. Data streams: models and algorithms. Vol. 31. Springer Science & Business Media, 2007.

[15] Wang, Weiping, Jianzhong Li, Chunyu Ai, and Yingshu Li. "Privacy protection on sliding window of data streams." In 2007 International Conference on Collaborative

Computing: Networking, Applications and Worksharing (CollaborateCom 2007), IEEE, 2007, 213-221.

[16] Huang, Wangfei, Lifei Chen, and Qingshan Jiang. "A novel subspace clustering algorithm with dimensional density." In 2010 2nd International Conference on Future Computer and Communication, vol. 3, IEEE, 2010, V3-71.

[17] Fung, Benjamin CM, Ke Wang, Ada Wai-Chee Fu, and Jian Pei. "Anonymity for continuous data publishing." In Proceedings of the 11th international conference on Extending database technology: Advances in database technology, 2008, 264-275.

[18] Cao, Jianneng, Barbara Carminati, Elena Ferrari, and Kian-Lee Tan. "Castle: Continuously anonymizing data streams." IEEE Transactions on Dependable and Secure Computing 8, no. 3 (2010): 337-352.

[19] Wang, Pu, Jianjiang Lu, Lei Zhao, and Jiwen Yang. "B-castle: An efficient publishing algorithm for k-anonymizing data streams." In 2010 Second WRI Global Congress on Intelligent Systems, vol. 2, IEEE, 2010, 132-136.

[20] Zakerzadeh, Hessam, and Sylvia L. Osborn. "Faanst: fast anonymizing algorithm for numerical streaming data." In International Workshop on Data Privacy Management, pp. 36-50. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[21] Guo, Kun, and Qishan Zhang. "Fast clustering-based anonymization approaches with time constraints for data streams." Knowledge-Based Systems 46 (2013): 95-108.

[22] Mohammadian, Esmaeil, Morteza Noferesti, and Rasool Jalili. "FAST: fast anonymization of big data streams." In Proceedings of the 2014 international conference on big data science and computing, 2014, 1-8.

[23] Yang, Lu, Xingshu Chen, Yonggang Luo, Xiao Lan, and Wei Wang. "IDEA: A utility-enhanced approach to incomplete data stream anonymization." Tsinghua Science and Technology 27, no. 1 (2021): 127-140.

[24] Otgonbayar, Ankhbayar, Zeeshan Pervez, Keshav Dahal, and Steve Eager. "K-VARP: k-anonymity for varied data streams via partitioning." Information Sciences 467 (2018): 238-255.

[25] Sadeghi-Nasab, Alireza, Hossein Ghaffarian, and Mohsen Rahmani. "Apache flink and clustering-based framework for fast anonymization of iot stream data." Intelligent Systems with Applications 20 (2023): 200267.

[26] Zhou, Bin, Yi Han, Jian Pei, Bin Jiang, Yufei Tao, and Yan Jia. "Continuous privacy preserving publishing of data streams." In Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, 2009, 648-659.

[27] Chaudhuri, Surajit, Vivek Narasayya, and Ravishankar Ramamurthy. "Diagnosing estimation errors in page counts using execution feedback." In 2008 IEEE 24th International Conference on Data Engineering, IEEE, 2008, 1013-1022.

[28] Zakerzadeh, Hessam, and Sylvia L. Osborn. "Delay-sensitive approaches for anonymizing numerical streaming data." International journal of information security 12, no. 5 (2013): 423-437.

[29] Sopaoglu, Ugur, and Osman Abul. "A utility based approach for data stream anonymization." Journal of Intelligent Information Systems 54, no. 3 (2020): 605-631.

[30] Otgonbayar, Ankhbayar, Zeeshan Pervez, and Keshav Dahal. "$ X-BAND $: Expiration Band for Anonymizing Varied Data Streams." IEEE Internet of Things Journal 7, no. 2 (2019): 1438-1450.

[31] Sopaoglu, Ugur, and Osman Abul. "Classification utility aware data stream anonymization." Applied Soft Computing 110 (2021): 107743.

[32] Joo, Yongwan, and Soonseok Kim. "SUHDSA: Secure, Useful, and High-Performance Data Stream Anonymization." IEEE Transactions on Knowledge and Data Engineering (2024).

[33] Patil, Rahul A., and Pramod D. Patil. "Efficient approximation and privacy preservation algorithms for real time online evolving data streams." World Wide Web 27, no. 1 (2024): 5.

[34] Wang, Jinyan, Kai Du, Xudong Luo, and Xianxian Li. "Two privacy-preserving approaches for data publishing with identity reservation." Knowledge and Information Systems 60, no. 2 (2019): 1039-1080.

[35] Edmunds, E., S. Muthukrishnan, Subarna Sadhukhan, and Shinjiro Sueda. "MoDB: database system for synthesizing human motion." In 21st International Conference on Data Engineering (ICDE'05), IEEE, 2005, 1131-1132.

[36] Otgonbayar, Ankhbayar, Zeeshan Pervez, and Keshav Dahal. "Toward anonymizing iot data streams via partitioning." In 2016 IEEE 13th International conference on mobile ad hoc and sensor systems (MASS), IEEE, 2016, 331-336.

[37] Sweeney, Latanya. "Guaranteeing anonymity when sharing medical data, the datafly system." In Proceedings of the AMIA Annual Fall Symposium, p. 51. 1997.

[38] Hundepool, Anco, and L. C. R. J. Willenborg. "μ-and τ-argus: Software for statistical disclosure control." In Third international seminar on statistical confidentiality. 1996.

[39] Yuan, Linlin, Tiantian Zhang, Yuling Chen, Yuxiang Yang, and Huang Li. "An Innovative k-Anonymity Privacy-Preserving Algorithm to Improve Data Availability in the Context of Big Data." Computers, Materials & Continua 79, no. 1 (2024).

[40] Shamsinejad, Elham, Touraj Banirostam, Mir Mohsen Pedram, and Amir Masoud Rahmani. "Anonymizing big data streams using in-memory processing: A novel model based on one-time clustering." Journal of Signal Processing Systems 96, no. 6 (2024): 333-356.

[41] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." IEEE Transactions on pattern analysis and machine intelligence 24, no. 5 (2002): 603-619.

[42] Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise." In kdd, vol. 96, no. 34, 1996, 226-231.