SN Computer Science (2023) 4:344
https://doi.org/10.1007/s42979-023-01781-6

ORIGINAL RESEARCH

Semantic Attack on Disassociated Transaction Data

Asma AlShuhail[1] · **Jianhua Shao**[2]

Received: 17 October 2022 / Accepted: 9 March 2023

Abstract
Accessing and sharing information, including personal data, has become easier and faster than ever because of the Internet. Therefore, businesses have started to take advantage of the availability of data by gathering, analysing, and utilising individuals' data for various purposes, such as developing data-driven products and services that can help improve customer satisfaction and retention, and lead to better healthcare and well-being provisions. However, analysing these data freely may violate individuals' privacy. This has prompted the development of protection methods that can deter potential privacy threats by anonymising data. Disassociation is one anonymisation approach used to protect transaction data. It works by dividing data into chunks to conceal sensitive links between the items in a transaction, but it does not account for semantic relationships that may exist among the items, which adversaries can exploit to reveal protected links. We show that our proposed de-anonymisation approach could break the privacy protection offered by the disassociation method by exploiting such semantic relationships. Our findings indicate that the disassociation method may not provide adequate protection for transactions: up to 60% of the disassociated items can be reassociated, thereby breaking the privacy of nearly 70% of the protected items. In this paper [an extension to our work reported in AlShuhail and Shao (Semantic attack on disassociated transactions. In: Proceedings of the 8th International Conference on information systems security and privacy-ICISSP, INSTICC. SciTePress, pp. 60–72, 2022)], we develop additional techniques to reconstruct transactions, with additional experiments to illustrate the impact of our attacking method.

Keywords Data privacy · Semantic attack · Transaction data · Disassociation

Introduction

Transaction data consist of a set of records, with each record containing a set of terms or items. Social media platforms, online marketplaces and healthcare information systems are some examples where transaction data are generated, collected and often shared with third-party academic or commercial institutions for further study and analysis. Although this type of data publication may help organisations enhance their service offerings and build innovative solutions that would not be possible otherwise, one concern that must be addressed is the safeguarding of confidential and sensitive information included within the datasets to be released.

However, eliminating identifying information, such as National Insurance number, from a dataset may not be adequate to preserve individuals' privacy, because a combination of other information contained in de-identified data can still be used to identify individuals. For example, Table 1 contains four records or transactions, each of which describes a patient's medical diagnosis and treatments. Suppose that an adversary is aware that Mary suffers from *epilepsy* and that she is included in the dataset. He or she can then deduce that transaction four belongs to Mary, and thus discover other information associated with her.

Over the past 2 decades, the research community has devoted significant effort to understanding how to preserve the privacy of individuals when their data must be published [2]. A variety of privacy models and approaches have

This article is part of the topical collection "Advances on Information Systems Security and Privacy" guest edited by Steven Furnell and Paolo Mori.

✉ Asma AlShuhail
aalshuhail@kfu.edu.sa

Jianhua Shao
shaoj@cardiff.ac.uk

[1] College of Computer Sciences & Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia

[2] School of Computer Science & Informatics, Cardiff University, Cardiff CF24 3AA, UK

Published online: 20 April 2023

Table 1 An example of transaction data

TID	Transactions
1	Diabetes, Arthritis, Osteoporosis, Schizophrenia
2	Hypertension, Arthritis, Osteoporosis, Calcium
3	Hypertension, Diabetes, Osteoporosis, Obesity, Bulimia, Depression
4	Hypertension, Diabetes, Arthritis, Obesity, Bulimia, Epilepsy

Table 2 Disassociated data

Diabetes, Arthritis, Osteoporosis		Schizophrenia
Hypertension, Arthritis, Osteoporosis	Obesity, Bulimia	Calcium
Hypertension, Diabetes, Osteoporosis	Obesity, Bulimia	Depression
Hypertension, Diabetes, Arthritis		Epilepsy

been proposed. These strategies aim to prevent intentional or unintentional data misuse by modifying it, so that individuals' identities and their sensitive information cannot be obtained [3]. Anonymisation techniques include generalisation, suppression and perturbation [4–6].

Transaction data can be viewed as high-dimensional data that are difficult to protect. Using generalisation or suppression techniques may lead to substantial information loss. The disassociation method [7] is one of the methods that has been developed to protect transaction data by hiding sensitive links among items. This method is built on the k^m-anonymity privacy model, which states that if an attacker has knowledge of up to m items, they cannot match their knowledge to fewer than k transactions. In other words, when a dataset is disassociated, it ensures that every possible combination of m items appears in the published dataset at least k times. Using the disassociation method, items in transactions are protected by grouping them, so that each group's items satisfy the k^m-anonymity condition.

For example, Table 2 is a disassociated version of Table 1 in which *epilepsy*, for instance, is separated from its transaction, because it does not appear frequently enough with other items. Hence, knowing that Mary suffers from *epilepsy* will no longer be enough to definitely link Mary to transaction 4.

This method protects data by dissociating the links among the data items that are vulnerable to attack without modifying the data themselves, so that more of the data's utility can be retained [7]. However, the disassociation method assumes that the items do not have semantic meaning, and it does not take into consideration semantic relationships that may exist among the items in a transaction.

For example, consider the disassociated transaction 4 in Table 2. Although *Epilepsy* is separated into a different column, the fact that people with *epilepsy* seem more likely to also have *diabetes* and *hypertension* can still be used to link *Epilepsy* back to its transaction, thereby breaking the protection for the data.

The current paper is an extended version of our work originally presented at the 8th International Conference on Information Systems Security and Privacy [1]. We presented the first attempt to use semantic relationships to reconstruct original transactions from their disassociated versions in [1]. Normalised Google distance (NGD) [8] and word embedding (WE) [9] are used to score the semantic relationships among the terms and rebuild the links between sub-records. We extend our previous work by introducing an additional technique for reconstructing transactions and by presenting additional experiments that show the effectiveness of our semantic attacks.

The rest of this paper is organised as follows. In the section "Related Works", we discuss the work related to the current paper. In the section "Overview of the Disassociation Method", we give an explanation of the disassociation method. In the section "Proposed Attacking Approach", we present our approach to a semantic attack and explain the two key steps of our attacking approach. In the section "Attacking Methods", we illustrate how the chunks in a disassociated dataset can be attacked by proposing four heuristic strategies to reconstruct the original transactions. In the section "Experiments", we report the experimental results. Finally, in the section "Conclusions", we conclude the paper.

Related Works

Protecting published personal data and preventing any potential violations of privacy have been of great interest to researchers over the past few years [10]. Different ways in which adversaries can break the protection of anonymised data have been investigated. One of the most well-known kinds of attack is when attackers combine their additional knowledge, gained from external sources, with the anonymised dataset to reidentify individuals. This method is called linkage attack, and in this attack, an attacker may use quasi-identifiers, such as a postcode, gender, or date of birth, that are present in the anonymised dataset to identify individuals. The most famous incident of this type is the reidentification attack on a Massachusetts hospital discharge database in which the attacker combined it with a public voter database. Sweeney [11] was able to reidentify Massachusetts Governor William Weld by connecting his data from the voter registration list to his data in the medical dataset, even though all explicit identifiers had been removed.

In another type of attack called a minimality attack, the adversary's knowledge may be extended to include anonymisation mechanisms and privacy requirements that have been

used in anonymising the published dataset. The adversary may gain this knowledge by analysing the published dataset or its documentation to discover the mechanism behind the anonymisation method, thus allowing them to determine the privacy constraints that have been implemented. Using this information, the attacker may expose individual identities and break their anonymity [2, 12–14].

Yet, another type of attack known as an inference attack has been used to violate the privacy of published anonymised data. An adversary can infer sensitive information they do not have access to from published non-sensitive information using various techniques, such as authorised query results and data mining tools [15–17]. Several studies have demonstrated an inference attack on anonymised data [18, 19]. For instance, Kifer [20] illustrated how to use the non-sensitive attributes of a specific individual to disclose their sensitive data that have been anonymised using the anatomy technique, which allowed him to learn the correlations between attributes.

All these types of attack rely on data frequency to identify individuals and the sensitive information associated with them from a published dataset. They do not exploit the semantic relationships that may exist among data items when violating data privacy, as we do in the present paper. However, semantic inference can be used in reidentification or exposing sensitive items [21, 22]. For example, the co-occurrence of two terms in a context can be used to find the semantic relationships between them, such as the relationships between medical conditions and treatments. For instance, cancer and chemotherapy often appear together in a medical context; therefore, their semantic relationship is strong.

This type of semantic attack relies on the adequate assessment of the probability that two or more terms will occur together in a given context. The field of natural language processing (NLP) provides various tools for interpreting and comprehending semantic connections [23–26]. For instance, Chow et al. [27] used the term co-occurrences on the web as part of their inference detection model to predict and detect attacker interpretations from text. In a similar way, Sanchez et al. [28] used the World Wide Web as a corpus and employed a semantic distance measure called point-wise mutual information (PMI) [23] to identify related terms. Their work primarily targeted general text data, whereas we look at transaction data specifically. In addition, Chow et al. [27] and Sanchez et al. [28] were concerned with determining if the remaining terms after sanitisation could still be used to discover the removed terms, and our research focuses on uncovering the sensitive links between terms in a disassociated dataset.

Shao and Ong [29] proposed a semantic attack on set-generalised transactions [30]. Their de-anonymisation framework employs semantic relationships to determine which items in the generalised dataset are likely to be fake items. They used NGD to analyse the relationships among terms, which is similar to our work. Although we consider NGD a valuable tool for analysing the semantic relations among the terms, our work has a different focus than theirs. They used semantic distances to remove fake items, while we use them to reassociate terms.

Overview of the Disassociation Method

To explain how our proposed de-anonymisation approach works, we begin by discussing the disassociation method in detail. The disassociation method is an anonymisation technique aimed at protecting the identities and sensitive personal information contained in a released transaction dataset [7]. The original terms are kept intact, but the fact that a rare combination of terms exists in the same transaction is hidden by the disassociation method. In other words, this method preserves the privacy of individuals by disassociating transaction terms that help in identifying infrequent combinations to prevent an attacker from exploiting these combinations to identify individuals in a released dataset.

Preliminaries

Let $W = \{w_1, \ldots, w_m\}$ be a finite set of words called terms. A transaction T over W is a set of terms $T = \{t_1, t_2, \ldots, t_k\}$, where t_j, $1 \leq j \leq k$ is a distinct term in W. A transaction dataset $D = \{T_1, T_2, \ldots, T_v\}$ is a set of transactions over W.

Definition 1 (k^m-*anonymity*) If an adversary knows up to m terms of a record but cannot use this knowledge to identify less than k candidate records in a dataset, then the dataset is said to be k^m-anonymous. In other words, the k^m-anonymity model ensures that any combination of m terms occurs in the dataset at least k times.

For example, even if an attacker knows that a certain individual has *hypertension* and *diabetes* and that the individual's medical record is published in a 2^3-anonymous dataset, then the attacker will not be able to identify this individual's record from less than two records.

Definition 2 (*Disassociated transactions*) Let $D = \{T_1, T_2, \ldots, T_n\}$ be a set of transactions. Disassociation takes as an input D and results in an anonymised dataset $\hat{D}$, which groups transactions into clusters $\hat{D} = \{P_1, \ldots, P_z\}$. Each cluster partitions the transaction terms into a number of record chunks $\{C_1, \ldots, C_s\}$ and a term chunk C_T. The record chunks contain the terms in an itemset form called the sub-record $\{SR_1, SR_2, \ldots, SR_v\}$ that satisfies k^m-anonymity,

Table 3 Horizontal partitioning (the first iteration)

	Transactions
P_1	Diabetes, Arthritis, Osteoporosis
	Hypertension, Diabetes, Osteoporosis
	Hypertension, Diabetes, Arthritis
P_2	Hypertension, Arthritis, Osteoporosis

Table 4 Horizontal partitioning (the resulting cluster)

	Transactions
P_1	Diabetes, Arthritis, Osteoporosis
	Hypertension, Diabetes, Osteoporosis
	Hypertension, Diabetes, Arthritis
	Hypertension, Arthritis, Osteoporosis

while the term chunk contains the rest of the terms of the transactions.

Disassociation Method

There are three stages involved in disassociating transactions. The first is the horizontal partitioning (HP) of clusters into individual transactions. The second is the vertical partitioning (VP) of infrequent term combinations in a cluster into multiple groups. The third is refining, which is used to reduce information loss and increase data utility.

Horizontal Partitioning

Transactions are grouped into clusters. HP is the binary splitting of data into groups based on the frequency of a term's occurrence in the dataset using a recursive algorithm. The horizontal partitioning step's goal is to minimise information loss: each partitioned cluster should contain as few transactions and as many related terms as possible. This will result in reduced disassociation between the terms in the next stage and will improve data utility.

The algorithm first identifies the most frequent term and then uses it to split the transactions into two groups—those that include the term and those that do not. The algorithm then determines the next most frequent term for each cluster and divides the transactions based on this term. For instance, if k equals two and we need to horizontally partition the dataset in Table 1, the first iteration of the algorithm would choose *Diabetes* as the most frequent term. HP then splits the transactions into two clusters, with the first cluster containing those transactions containing *Diabetes* and the second containing all other transactions, as shown in Table 3.

Table 3 shows the results of the first iteration of HP. Although P_1 meets the size condition, P_2 is smaller than k. As a result, HP will reunite P_2 and P_1 as a single cluster (Table 4).

Vertical Partitioning

VP is a method of disassociating and hiding the combinations of infrequent terms. The procedure is executed separately on each cluster. A cluster is split vertically into two types of chunks-record chunks and term chunks. The sub-records that pass the k^m-anonymity condition are included in record chunks. Each m-sized combination of terms must occur at least k times in a record chunk. The terms that have not been placed in record chunks are moved into the term chunk. Each cluster may include several record chunks but only one term chunk.

To illustrate VP, let us consider the example in Table 4. If $m = 2$ and $k = 2$, then the terms of transactions will be disassociated into chunks, as shown in Table 5, ensuring that all resulting record chunks are 2^2-anonymous.

In transactions 3 and 4, the terms *Obesity* and *Bulimia* create a 2^2-anonymous sub-record, but both terms have not appeared enough with *Osteoporosis* or *Arthritis*. Therefore, VP pushes them to the second record chunk. In addition, VP moves all terms that have not appeared in the record chunks to the term chunk. Therefore, *Schizophrenia*, *Calcium*, *Depression* and *Epilepsy* are placed in the term chunk (Table 5).

Refining

The purpose of the refining stage is to increase the usefulness of released data while preserving anonymity. This stage focuses on term chunks and attempts to reduce the number of terms in term chunks by adding *joint clusters* that are shared across multiple clusters. The reader is referred to [7] for a more detailed description of the refining step and the disassociation algorithm.

Proposed Attacking Approach

We use a two-staged attack. The first stage is *scoring*, which involves determining the scores of semantic associations among the terms in a disassociated dataset. The second stage, which is known as *selection*, employs these semantic scores to identify which terms should be reassociated with reconstruct the original transactions. The transactions anonymised by the disassociation method serve as the input dataset for our approach.

Table 5 Disassociated transactions

	Record chunks		Term chunk
ID	C_1	C_2	C_T
1	Diabetes, Arthritis, Osteoporosis		Schizophrenia
2	Hypertension, Arthritis, Osteoporosis	Obesity, Bulimia	Calcium
3	Hypertension, Diabetes, Osteoporosis	Obesity, Bulimia	Depression
4	Hypertension, Diabetes, Arthritis		Epilepsy

Scoring Stage

In the scoring stage, we employ two measures, the NGD [8] and WE [9, 31], to determine the strength of semantic relationships that exist among terms. We use the first record's sub-records as an *anchoring chunk*. For each cluster, this step calculates the semantic scores between the terms in the anchoring chunk and the terms in other chunks. Algorithm 1 provides the pseudocode for the scoring stage.

Algorithm 1 Scoring

Input: Disassociated transactions
Output: Semantic scores
1: **for** Each cluster P **do**
2: **for** Each record chunk RC of P **do**
3: **for** Each sub-record SR of RC **do**
4: Calculate the semantic score between SR and all sub-records in C_1 by NGD or WE
5: $scores_P = scores_P \cup$ scores
6: **end for**
7: **for** Each term t_i in C_T **do**
8: Calculate the semantic score between t_i and all sub-records in C_1
9: **end for**
10: **end for**
11: **end for**
12: **return** $scores_P$

The algorithm is executed for each cluster P in the disassociated dataset $\hat{D}$. As can be seen in Table 5, there are two sorts of chunks in a disassociated dataset-record chunks $(C_1, C_2, \ldots, C_n)$ and a term chunk (C_T). Each record chunk contains a number of sub-records $(SR_1, SR_2, \ldots, SR_v)$, and the term chunk contains terms $(t_1, t_2, \ldots, t_j)$. For each sub-record SR in record chunks from C_2 to C_n and for each term in C_T, the algorithm uses NGD or WE to compute its semantic relationships with each sub-record ASR in C_1 (steps 3 and 4). In steps 7 and 8, the algorithm finds the semantic scores for each term in the term chunk. In each cluster, all resulting scores between disassociated terms and sub-records in the anchoring chunk are stored in $scores_P$ and returned by the algorithm (step 13). For example, for Table 5, to determine the semantic distance between the first sub-record (*Obesity, Bulimia*) in C_2 and the first sub-record (*Diabetes, Arthritis, Osteoporosis*) in C_1 using WE, we obtain three scores [0.54, 0.35 and 0.40]. This procedure will be performed for the next three sub-records in C_1; then, it will be performed to find the semantics scores between each term *Schizophrenia,Calcium, Depression and Epilepsy* in C_T and the four sub-records in C_1.

Selection Stage

The purpose of this stage is to reassociate the sub-records in the record chunk and the terms in the term chunk based on the semantic scores obtained from the scoring step to reconstruct the original transactions from the disassociated ones. Algorithm 2 illustrates how the selection step is executed, and we will discuss the four heuristic reconstruction methods we proposed in the following section.

Algorithm 2 Selection

Input: Disassociated transactions, Semantic scores
Output: Reconstructed transactions
1: **for** each cluster P **do**
2: **for** each record chunk RC **do**
3: **for** each sub-record SR_i in RC **do**
4: ASR_k = reconstruction (SR_i)
5: Update ASR_k in C_1 with RS_i
6: **end for**
7: **end for**
8: **for** each Term t_i in C_T **do**
9: ASR_k = reconstruction (t_i)
10: Update ASR_k in C_1 with t_i
11: **end for**
12: Rec_P = reconstructed transactions of P
13: **end for**
14: **return** Rec_P

The algorithm is run for each cluster P separately. For each record chunk from C_2 to C_n in P, a reconstruction method is executed for each sub-record RS_i in a record chunk (steps 3 and 4). This will find the most-related ASR_i in C_1 for RS_i, and the corresponding sub-record in C_1 will then be updated with RS_i (step 5). The reconstruction of the terms in the term chunk C_T is performed in the same manner (steps 8–10). After processing all sub-records in the record chunks and terms in the term chunk, the transactions are considered to be reconstructed, and step 14 returns the reconstructed transactions.

Attacking Methods

This section illustrates how our approach can attack record and term chunks. We propose four heuristic strategies that use semantic scores to rebuild transactions that have been disassociated.

To launch an attack on record chunks, a semantic relationship calculation is performed on the anchoring chunk C_1 and the chunks from C_2 to C_n. Next, the selecting step is implemented. The averaging-based attack (ABA) method and VP attack (VPA) method will be used in attacking record chunks. This is because most record chunks include sub-records that contain more than one term, which may cause varying degrees of semantic relatedness between terms in two different sub-records. However, the most-related attack (MRA) method and the related-group attack (RGA) method will not be used at this attack level, because these strategies may not accurately capture the semantic score between two sub-records.

In contrast to record chunks, the terms included in the term chunks of a cluster are single terms. The disassociation method pushes rare terms with support less than k to term chunks to ensure that no terms may be associated with fewer transactions than the size of the cluster. To apply the attack to the term chunk, the scoring step is already completed for each cluster P in the disassociated dataset between each term in the term chunk and all sub-records in the anchoring chunk. Next, the selection step is executed. All our proposed attacking methods can be used to attack term chunks.

Averaging-Based Attack (ABA)

This strategy assumes that the terms used in a single transaction share the same context. Hence, the selection step considers all the terms included in the anchoring chunk's sub-records. Therefore, the semantic scores for all terms in ASR_i are taken into account to choose which sub-record *ASR* in C_1 corresponds to a certain *SR* or term *t* in other chunks. In other words, depending on the average of the *ASR* terms, this method selects the best semantically relevant sub-record.

Algorithm 3 shows the pseudocode for the ABA method. For each disassociated input sub-record or term, the algorithm is executed. First, each *ASR* in the anchoring chunk is assigned a score equal to the weighted average of the semantic relationship (SR) scores between the terms in the SR or *t* and the *ASR* (steps 1 and 2). The sub-records in the anchoring chunk are then ordered from most to least related in *N*, here based on the averages (step 5). If the input is a sub-record *SR*, then the algorithm computes the count of the number of sub-records in a record chunk (step 8). Based on the count, the algorithm returns the number of the most semantically related sub-records *ASR* (step 10). If the input is a term *t*, the algorithm can return $k-1$ of the most-related sub-records *ASR* from the list (steps 13–15).

Table 6 ABA scoring for Table 5

Record chunks		Term chunk			
C_1	C_2	C_T			
	Obesity, Bulimia	Schizophrenia	Calcium	Depression	Epilepsy
Diabetes, Arthritis, Osteoporosis	0.43	**0.48**	0.34	0.38	0.57
Hypertension, Arthritis, Osteoporosis	0.40	0.47	**0.37**	0.40	0.54
Hypertension, Diabetes, Osteoporosis	**0.46**	0.44	0.35	**0.44**	0.57
Hypertension, Diabetes, Arthritis	**0.45**	0.43	0.29	0.39	**0.58**

The bold values represent the strongest relationships (semantically) in the table, which will be considered later in the reconstruction process of the transactions

Table 7 Reconstructed transactions (ABA)

TID	Transactions
1	Diabetes, Arthritis, Osteoporosis, **Schizophrenia**
2	Hypertension, Diabetes, Osteoporosis, **Calcium**
3	Hypertension, Diabetes, Arthritis, **Obesity, Bulimia, Depression**
4	Hypertension, Arthritis, Osteoporosis,**Obesity, Bulimia, Epilepsy**

The bold values represent the reconstructed terms for each transaction after calculating the strongest relationships by different proposed methods

term chunk. To attack this cluster, we need to recombine the disassociated sub-records SR in C_2 and each term in the term chunk with anchoring chunk C_1. Using the WE semantic metric, the method finds the scores for all the terms and sub-records in different chunks. Table 6 shows the semantic scores for Table 5.

The ABA method considers transaction terms semantically equal, such as those describing a medical condition. Using Eq. 1, ABA determines the average semantic relationship score between a term or sub-record from chunks and all the terms in the anchoring chunk. In

Algorithm 3 ABA

Input: C_1, SR or t, k
Output: ASR_i

for each sub-record ASR_i in C_1 **do**
  Calculate the average score of the total
  Semantic relationships scores for SR or t
end for
Arrange sub-records of C_1 based on the average in list N

if Input is SR **then**
  Find the SR count
  for $i = 1$ to *count* **do**
    return Top ASR_i in N
  end for
end if
if Input is t **then**
  for $i = 1$ to $k - 1$ **do**
    return Top ASR_i in N
  end for
end if

Consider the disassociated transactions in Table 5 as an illustration of this attack method. In the example in Table 5, the cluster consists of two record chunks and one

$$ABA(ASR, SR) = \frac{\sum_{i=1}^{n} \frac{\sum_{i=1}^{x}(SC)}{|x|}}{|n|}, \quad (1)$$

Table 8 RGA scoring for Table 5

Record chunks		Term chunk			
C_1	C_2	C_T			
	Obesity, Bulimia	Schizophrenia	Calcium	Depression	Epilepsy
Diabetes, Arthritis, Osteoporosis	0.34	**0.52**	0.41	0.43	0.59
Hypertension, Arthritis, Osteoporosis	0.37	**0.52**	**0.45**	**0.46**	0.57
Hypertension, Diabetes, Osteoporosis	**0.36**	0.48	**0.45**	**0.46**	0.60
Hypertension, Diabetes, Arthritis	**0.36**	0.46	0.36	0.45	**0.60**

The bold values represent the strongest relationships (semantically) in the table, which will be considered later in the reconstruction process of the transactions

SC is the semantic score between *ASR* and *SR*, *x* is the number of terms in *SR*, and *n* is the number of terms in *ASR*.

Therefore, if we use the semantic score from Table 6, the reconstructed transactions are shown in Table 7. As can be seen, ABA accurately reconstructed the original transactions.

Related-Group Attack (RGA)

In some datasets, multiple contexts may be included in a single transaction. For example, a patient's medical file can include two medical conditions that are irrelevant to each other. In this scenario, the final semantic score may be inaccurate, because the inclusion of unrelated terms in the semantic calculation of the *ASR* from C_1 could lead to the reassociation of the term *t* or sub-record *SR* with the incorrect transaction.

In the selection stage, the RGA method takes into account a scenario in which the terms might belong to different contexts. In other words, a term *t* or sub-record *SR* from chunks might be semantically connected to certain terms but not others in a sub-record *ASR* in the anchoring chunk. This makes it unreliable to consider all terms equally when selecting the optimal transaction for reconstruction.

Table 9 Reconstructed transactions (RGA)

TID	Transactions
1	Diabetes, Arthritis, Osteoporosis, **Schizophrenia**
2	Hypertension, Diabetes, Osteoporosis, **Calcium**
3	Hypertension, Diabetes, Arthritis, **Obesity, Bulimia, Depression**
4	Hypertension, Arthritis, Osteoporosis,**Obesity, Bulimia, Epilepsy**

The bold values represent the reconstructed terms for each transaction after calculating the strongest relationships by different proposed methods

In RGA, we suppose a sub-record *ASR* in the anchoring chunk may be split into two contexts. After the scoring stage, the RGA method employs the median semantic score between each *t* or *SR* to be reassociated and the sub-record *ASR* in the anchoring chunk as a division indicator. This division indicator divides each *ASR* in the anchoring chunk into two groups. The first group (*related group*) contains terms that are semantically close to *t* or *SR*, whereas the other group (*unrelated group*) contains the remainder of the terms. In the calculation step, only the semantic scores for the *related-group* terms are considered.

Table 10 MRA scoring for Table 5

Record chunks		Term chunk			
C_1	C_2	C_T			
	Obesity, Bulimia	Schizophrenia	Calcium	Depression	Epilepsy
Diabetes, Arthritis, Osteoporosis	**0.54**	**0.53**	**0.50**	0.44	0.63
Hypertension, Arthritis, Osteoporosis	0.45	**0.53**	**0.50**	**0.47**	0.63
Hypertension, Diabetes, Osteoporosis	**0.54**	**0.53**	**0.50**	**0.47**	0.63
Hypertension, Diabetes, Arthritis	**0.54**	0.50	0.40	**0.47**	0.56

The bold values represent the strongest relationships (semantically) in the table, which will be considered later in the reconstruction process of the transactions

Algorithm 4 RGA

Input: C_1, SR or t, k
Output: ASR_i

1: **for** each sub-record ASR_i in C_1 **do**
2: Calculate the division indicator for SR or t
3: Divide terms into RG and NG based on the division
4: Calculate the average semantic score for RG
5: **end for**
6: Arrange sub-records of C_1 based on the average in list N
7:
8: **if** Input is SR **then**
9: Find the SR count
10: **for** $i = 1$ to $count$ **do**
11: **return** Top ASR_i in N
12: **end for**
13: **end if**
14: **if** Input is t **then**
15: **for** $i = 1$ to $k - 1$ **do**
16: **return** Top ASR_i in N
17: **end for**
18: **end if**

Algorithm 4 shows RGA's pseudocode. The algorithm recombines disassociated terms and sub-records. For each sub-record *ASR* in the anchoring chunk, the division indicator is computed (steps 1 and 2). Based on the division indicator, the *ASR* anchoring chunk terms are split into related and unrelated groups (line 3). Only the terms in *RG* are included in *ASR*'s semantic computation, and the average of their semantic connection scores is computed in line 4. The sub-records in the anchoring chunk are then organised from most to least linked based on averages (step 6). For *SR* sub-records, the algorithm returns *ASR* (step 10). The algorithm returns $k-1$ relevant sub-records *ASR* for term *t* (steps 14–16).

To demonstrate how the RGA technique works, we use Example 5. We utilise Eq. 2 to locate the division

Table 11 Reconstructed transactions (MRA)

TID	Transactions
1	Diabetes, Arthritis, Osteoporosis, **Schizophrenia**
2	Hypertension, Diabetes, Osteoporosis, **Calcium**
3	Hypertension, Diabetes, Arthritis, **Obesity, Bulimia, Depression**
4	Hypertension, Arthritis, Osteoporosis, **Obesity, Bulimia, Epilepsy**

The bold values represent the reconstructed terms for each transaction after calculating the strongest relationships by different proposed methods

Table 12 Reconstructed transactions (VPA)

TID	Transactions
1	Diabetes, Arthritis, Osteoporosis, **Schizophrenia**, **Calcium**
2	Hypertension, Diabetes, Osteoporosis, **Depression**
3	Hypertension, Diabetes, Arthritis, **Obesity, Bulimia**
4	Hypertension, Arthritis, Osteoporosis,**Obesity, Bulimia, Epilepsy**

The bold values represent the reconstructed terms for each transaction after calculating the strongest relationships by different proposed methods

indicator. In

$$\mathrm{Div_i}(SC) = \begin{cases} SC\left[\frac{n+1}{2}\right] & \text{if } n \text{ is odd} \\ \frac{\left(SC\left[\frac{n}{2}\right]+SC\left[\frac{n}{2}+1\right]\right)}{2} & \text{if } n \text{ is even}, \end{cases} \quad (2)$$

SC is the ordered list of semantic scores for the terms of *ASR*, and *n* is the number of terms in *ASR*.

For example, to find the division indicator of the semantic scores *SC* (0.41, 0.50, and 0.53) for the first *ASR (Diabetes, Arthritis, Osteoporosis)* and *t (Schizophrenia)*, RGA performs the calculation shown in Eq. 3

$$\mathrm{Div_i}(SC) = SC\left[\frac{3+1}{2}\right] = 2. \quad (3)$$

Because the division indicator for the first *SR* is *0.50*, the term *Diabetes* is excluded from the semantic score calculation, because the semantic score between *Diabetes* and *Schizophrenia* is *0.41*, which is less than the division indicator. Consequently, *Diabetes* is moved to the unrelated group. Based on the related group, the semantic scores between chunks are provided in Table 8.

The reconstructed transactions are produced, as shown in Table 9. Here, RGA correctly reconstructed the original transactions. However, this method would be more effective with sub-records that include several terms that are likely to contain multiple contexts.

Most-Related Attack (MRA)

The MRA method emphasises the greatest semantic relationship between two groups of terms. With the RGA method, the strength of the semantic relationship between terms in a related group and a term or sub-record may vary. This is because the terms used in the transaction might be used in several contexts. MRA, however, identifies the term with the best semantic score to identify whether *ASR* is the most relevant for combining a term *t* or sub-record *SR*. In datasets with a high degree of sparsity, the semantic relationships between terms become more diverse, enhancing the possibility of having more varied semantic scores. Therefore, for each term *t* or sub-record *SR*, MRA arranges the terms of *ASR* in descending order of their degree of relationship to each other. Then, MRA will only include the most relevant term in every *ASR*. The approach will then include *t* or *SR* with the highest semantic score in *ASR*.

Algorithm 5 MRA

Input: C_1, SR or t, k
Output: ASR_i

```
for each sub-record ASR_i in C_1 do
    Find the best score in the semantic relationships for SR or t
end for
Arrange sub-records of C_1 based on the average in list N

if Input is SR then
    Find the SR count
    for i = 1 to count do
        return Top ASR_i in N
    end for
end if
if Input is t then
    for i = 1 to k − 1 do
        return Top ASR_i in N
    end for
end if
```

Algorithm 5 provides the MRA pseudocode. MRA obtains the best score from all semantic relationships between terms in *SR* or *t* and all terms in *ASR* for each sub-record in the anchoring chunk (steps 1 and 2). Based on the resulting scores, the sub-records are ranked from the highest to lowest scores (step 4). The method produces the number of associated sub-records *ASR* for *SR* here depending on how many *SR* are in a record chunk (steps 6 to 9). The algorithm can return $k-1$ most relevant sub-records *ASR* for term *t* (steps 12–14).

To demonstrate how MRA works, Table 10 displays the semantic scores between chunks after applying it to Example 5. For instance, when using MRA to reassociate the term *Calcium* from the term chunk, the term *Osteoporosis* in C_1 has the strongest semantic association. Therefore, just the sub-records containing *Osteoporosis* will be considered in reassociating *Calcium*.

Table 11 displays the results of the MRA strategy. All the original transactions were rebuilt accurately. This strategy works the best when a clear pair of terms establishes the semantic link between chunks of disassociated transactions, thereby stopping the noise from other terms.

Vertical Partitioning Attack (VPA)

This technique aims to validate the reconstruction using the VP stage from the disassociation method. The disassociation approach divides sub-records vertically into chunks based on k^m-anonymity. Unlike earlier, this method can be applied after using semantic associations to target the disassociated transactions. Instead, the VPA technique identifies potential combinations of chunks by identifying all possible sub-record combinations between each pair of chunks. As the first iteration, VPA uses the ABA method to create the first possible reconstructed transactions. Then, VPA applies VP to test the reconstruction. The reconstructed chunks are considered to be correct if the resulting chunks match the disassociated chunks. Otherwise, the VPA will move to the next possible combinations of chunks and check the VP again until finding those combinations that pass the partitioning.

This method is run for every cluster P in the disassociated dataset (Algorithm 6). First, each iteration is performed on every pair of record chunks from C_1 to C_n using the resulting reconstructed transactions from the ABA method or by identifying all potential combinations between the two record chunks (steps 2 and 3). Second, this approach will temporarily combine the two chunks based on one combination at a time before applying VP to the reconstructed transactions (steps 6–8). Suppose the vertical partitioning of the reconstructed record chunks produces the same record chunks as the disassociated transaction. In this case, the strategy adds sub-records permanently, and the reconstructed records will be kept (steps 9–11). Otherwise, the temporarily reconstructed record chunks will be discarded (step 13), and the method will check the next possible combination in step 15. This procedure will be repeated until it passes the VP for all record chunks. VPA will save the reconstructed cluster in Rec_P (step 17).

Algorithm 6 VPA

Input: Disassociated transactions
Output: Reconstructed transactions
1: **for** Each cluster P **do**
2: **for** Every two record chunks in (C_1 to C_n) **do**
3: Find ABA combinations **OR**
4: Find all possible combinations between(C_i and C_{i+1})
5: **end for**
6: **for** Each combination **do**
7: Add sub-records in C_i and C_{i+1}
8: Execute VP on current combination
9: **if** Current combination pass the VP **then**
10: Save Current combination
11: Move to next record chunk C_i
12: **else**
13: Discard Current combination
14: Check next combination
15: **end if**
16: **end for**
17: Rec_P=The reconstructed transactions of cluster P
18: **end for**

To demonstrate this method, we apply it to Example 5 in which there are two sub-records of *(Obesity, Bulimia)* in C_2. If the combination produced by ABA does not pass the VP, then the next step is to find all possible combinations. Based on the current combinations between C_1 and C_2, *(Obesity, Bulimia)* can be added based on one of the following combinations of the four sub-records in C_1: (1,2), (1,3), (1,4), (2,3), (2,4), and (3,4). For example, using the ABA combination, the VPA strategy will add *(Obesity, Bulimia)* to the third and fourth sub-records in C_1. After this, VP is applied to this combination. The combination would be correct if it produces similar record chunks as the disassociated transactions in Table 5.

The reconstructed transactions are shown in Table 12. However, suppose the ABA combination does not pass VP. In this case, there is usually more than one valid combination for combining two chunks, and the number of these valid combinations is affected by the number of transactions in a cluster. In addition, these valid combinations decrease when more chunks are combined. Therefore, even if reconstructed transactions pass the VP step, the combination may not be the same as the original combination. In addition, the VPA strategy is not applicable for terms in the term chunk; this strategy adds them randomly to transactions, which can reduce the effectiveness of this strategy.

Experiments

In this section, we describe the datasets used in our experiments and how we prepared them. We then evaluate the proposed methods empirically and test the different properties to evaluate our methods within a range of conditions.

Dataset Preparation and Experiment Setup

We used real-world datasets from EzineArticles (general articles)[1] to conduct our experiments. These articles are brief and cover a range of topics, making them a suitable source for our transactions. To construct our datasets, we chose about 1000 articles on different topics that have a varying number of keywords to form the transactions. Next, we anonymised the transactions by considering the following properties and parameters:

- Dataset density [or the type-token ratio (TTR) [32]] is the number of unique terms divided by the total number of terms appearing in all transactions of the dataset. In our experiments, we used the data density range from 0.2 to 0.7.
- The k parameter is used as a privacy constraint that needs to be satisfied in the disassociated dataset. Increasing the k value in disassociation means increasing the protection level. To evaluate this parameter's impact on the attacking performance, we tested the k values at 2, 3, 4, and 5.
- The *MaxClusterSize* parameter determines the largest size allowed as a cluster, and the value of this parameter cannot be less than the k value. In our experiments, the *MaxClusterSize* value ranges from k^2 to k^6.

To prepare our datasets for anonymisation and attack, we took the following steps to convert the articles (free text form) into transactions:

- We considered the main content of the articles only and ignored other information, such as titles, references, and external links that are contained in the articles.
- We applied tokenisation to process the articles. In natural language processing (NLP), tokenisation is the method of splitting text into smaller tokens. In our datasets, the text has been split into words.
- To appropriately analyse these tokens, we lemmatised the inflectional forms of words. This step converts a word into its dictionary form, called a lemma, using context and meaning. Therefore, for example, studies, studying and studies' will resolve into reverse direction 'study'.
- We removed any token that (1) is a stop word, (2) is a number or punctuation, or (3) is a single character. We also removed duplicate terms.
- To control topics and keywords, we employed an unsupervised technique known as non-negative matrix factorisation (NMF) for topic modelling [33] to identify the topics that appear in a collection of articles. We then clustered the articles based on the topic models.
- We classified and chose the articles based on the resulting topics from the previous step to construct transactions. From this, we derived our transactions, which contain around 4000 unique words from over 35 topics.
- To anonymise transactions using the disassociation method, we applied HP where transactions are grouped into clusters with a size between k and *MaxClusterSize*. Next, VP was applied.

Evaluation Measures

We introduced two measures. In two different ways, the first measure assesses how our approach may break privacy-transaction breakage and k^m-anonymity breakage. Transaction breakage measures how many disassociated transactions our methods can break. We consider the protection for a transaction is broken if at least one term is correctly reassigned to its transaction. The k^m-anonymity breakage computes the breakage based on attacking protected infrequent

[1] www.EzineArticles.com.

itemsets, where the infrequent itemsets are combinations of m terms that occur fewer than k times in a dataset.

The second metric indicates how much of the original data can be accurately recovered from the disassociated transactions. We employed accuracy and word mover's distance (WMD) [34] for this. The accuracy represents the proportion of correct reconstructions, here assessing how many terms in a transaction can be reconstructed using our methods. In the WMD, however, we measure information reconstruction by calculating the semantic distance between the original and reconstructed transactions.

Results and Discussion

In this section, we report the results of our attacking methods on disassociated datasets. A random attack on disassociated transactions will be used as a baseline. In the random attack, an adversary reassociates the record and term chunks at random, without any knowledge other than the published dataset. In addition, we discuss how the different k, density, and cluster size will affect our algorithms' performance. We evaluate the effectiveness of our attack in attacking record and term chunks using the four measurements discussed in the section "Evaluation Measures".

In Fig. 1, we analyse the performance of our algorithms with different k values. The k parameter controls the privacy protection level in the disassociated transaction. Increasing k means increasing protection, which can result in more terms being moved to term chunks and sub-records becoming more indistinguishable.

Higher k values improve our algorithms' accuracy. This happens for two reasons. First, increasing k implies that the number of transactions in a cluster will increase to meet the k^m-anonymity condition, which in turn will increase the number of sub-records in the anchoring chunks with the same semantic scores. This increases the probability of associating correct sub-records together. Second, anchoring chunks are more likely to have many identical sub-records with similar semantic scores; therefore, any sub-record selected to associate with a term is more likely to be valid.

However, the difference between our method's performance and random attacks decreases as k increases. Because the anchoring chunk sub-records become almost identical, the difference between their semantic scores becomes less relevant, and a random guess can do almost equally well in this case.

Regarding the accuracy of attacking record chunks, increasing k reduces the number of record chunks in a cluster, which means that fewer sub-records need to be reassociated; hence, the chance of combining the wrong sub-records decreases. Consequently, as k increases, so do the accuracy percentages, as shown in Fig. 2a.

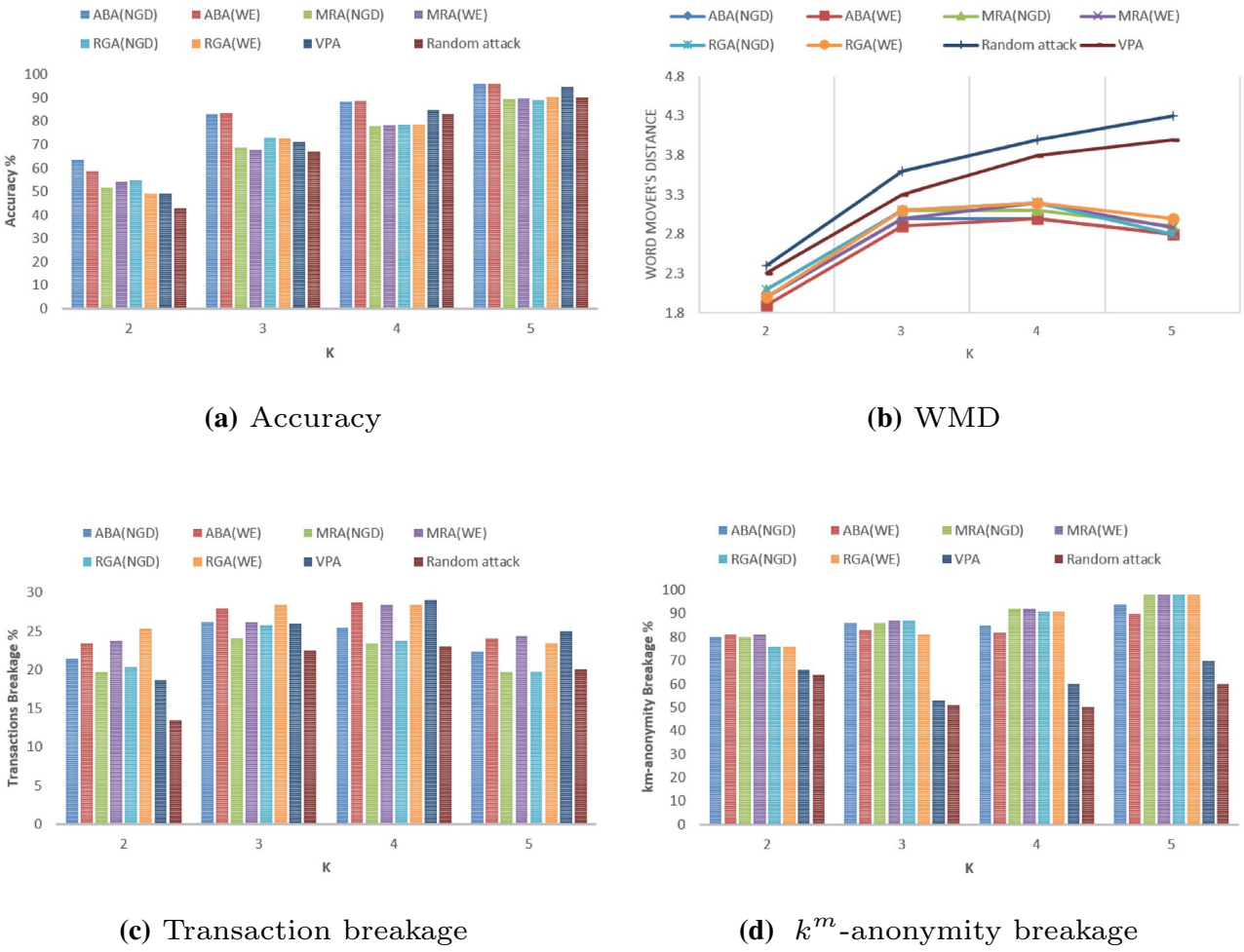
The increase of indistinguishable sub-records in the anchoring chunks increases the possibility of successfully reconstructing a transaction and finding this reconstructed transaction in the original dataset. This explains the narrowing performance gap between performances of these methods. Overall, ABA with NGD or WE performs well across k values. This relates to dataset density. The dataset employed in this experiment is dense; therefore, it depends on a few terms from the anchoring chunk to establish semantic relationships (Fig. 1a).

Figure 1b shows how the reconstructed transactions are semantically similar to the original transactions. However, as k increases, the semantic difference between the reconstructed and original transactions increases for the VPA and the random attack, while it slightly decreases for other semantic methods after k equals 4. The number of terms in the anchoring chunk affects the WMD; therefore, more terms in the anchoring chunk result in fewer terms in other chunks that need to be reassociated. Hence, as k increases, fewer terms in the anchoring chunk and different semantic distances between the terms in both the original and reconstructed transactions increase. However, the attack approaches maintained a low WMD with increasing k compared with the random attack.

Figure 1c shows how well our methods can break transaction privacy. An increasing k has different effects on the record and term chunks. Record and term chunk attacks show the opposite trends for breakage with the increasing of k. This explains the varying total transaction privacy breakage for different k values.

For attacking record chunks, as mentioned earlier, there is a negative relationship between the number of record chunks in a cluster and the k value, which means that the number of transactions that can be broken into is greater when k is smaller. However, at some points, NGD performs better than WE, whereas at other points, the opposite is true. This is due to the semantic association between the terms and how the semantic measure measures them. For the VPA approach, as the number of identical sub-records in a record chunk increases, the number of possible combinations that satisfy the k^m-anonymity requirement increases (Fig. 2b). Unlike record chunks attacking, the performance of attacking term chunks improves with a greater k value. This is because the number of distinguishable sub-records in the anchoring chunk decreases, so the number of transactions to choose from to reassociate decreases.

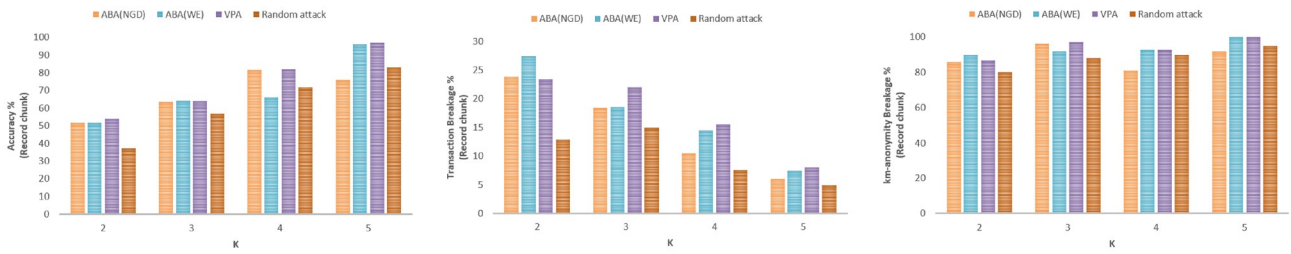
Figure 1d demonstrates how an increasing k affects the attack on protected infrequent itemsets. k^m-anonymity breakage increases with k. A greater k indicates more protected itemsets. However, because we associate terms based on semantic relationships in our attack methods, the increase



(a) Accuracy **(b)** WMD

(c) Transaction breakage **(d)** k^m-anonymity breakage

Fig. 1 The effect of k on attacking methods

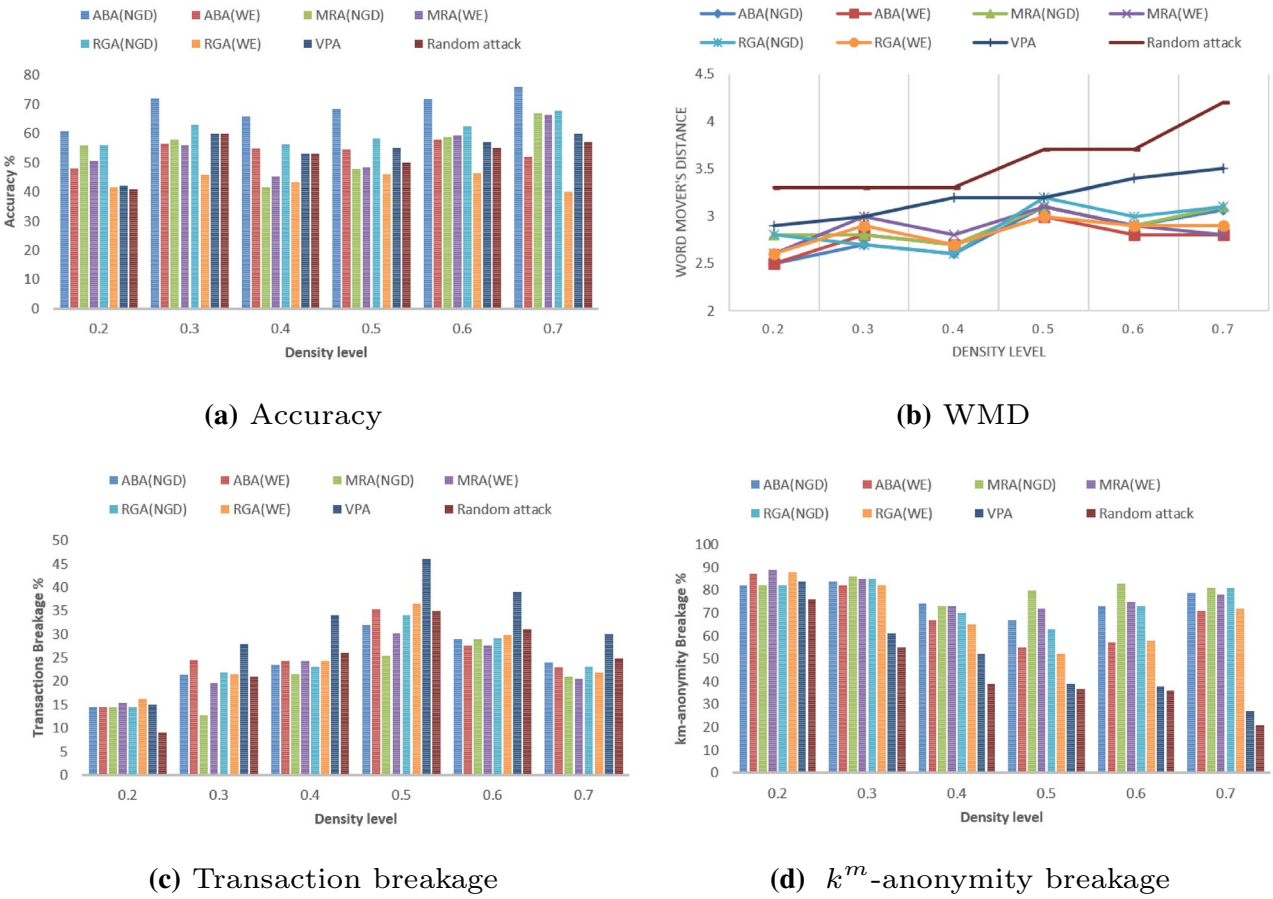


(a) Accuracy **(b)** Transaction breakage **(c)** k^m-anonymity breakage

Fig. 2 The effect of k on attacking record chunks

in the number of protected itemsets indicates a better possibility of identifying infrequent itemsets.

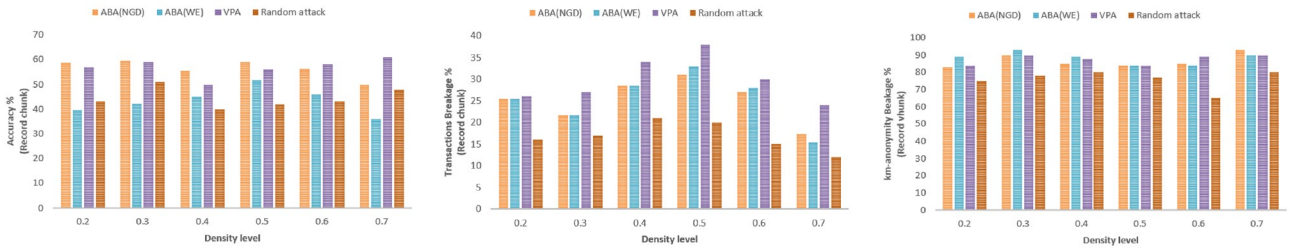
Figure 2 shows the four measures' results of the correlation between density levels and the performance of our algorithms. As the datasets become increasingly sparse, the accuracy of all attack methods improves. This is because increasing sparsity means more different terms and more varied semantic scores. However, when sparsity increases, those methods using NGD as a semantic measure outperform those using WE. This is because NGD can calculate the semantic score for any pair of terms by employing the World Wide Web as a corpus, while WE is restricted by the training corpus (Fig. 3a).



(a) Accuracy **(b)** WMD

(c) Transaction breakage **(d)** k^m-anonymity breakage

Fig. 3 The effect of density on attacking methods

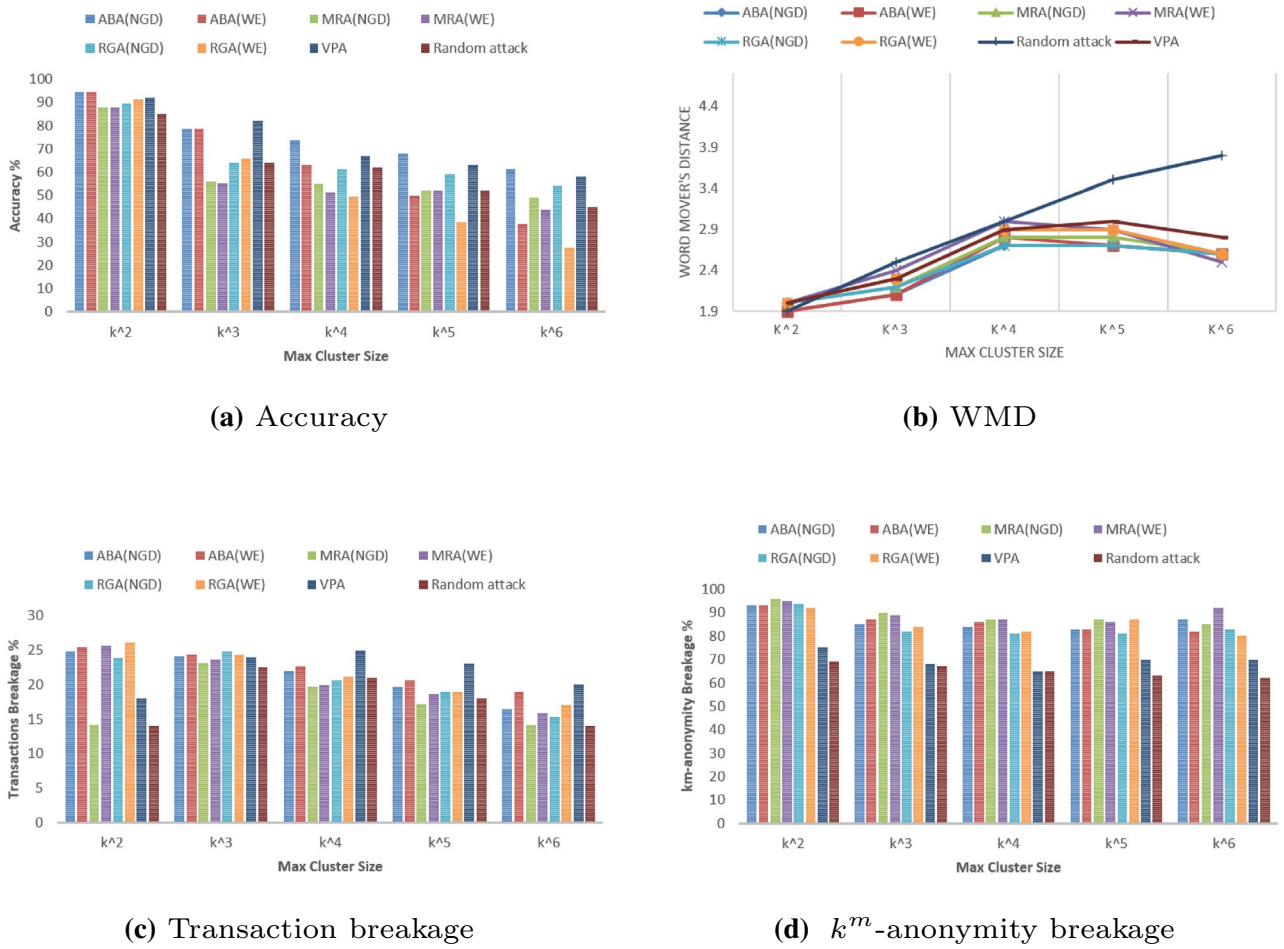


(a) Accuracy **(b)** Transaction breakage **(c)** k^m-anonymity breakage

Fig. 4 The effect of density on attacking record chunks

However, the record chunk attack is affected by an excessive increase in the sparsity level, because the number of terms with the required frequency to be included in the record chunks becomes very low (Fig. 4a). The results of the WMD measurements are shown in Fig. 3a. Because the density level did not significantly affect the overall reconstruction for most attack methods, the outcomes varied from 2.5 to 3. Figure 3c shows the results of the transaction breakage. Increasing sparsity enhances all attack techniques' breakage until 0.5 when it begins to decline. After 0.5, the number of sub-records or terms with a frequency greater than k decreases. Therefore, the record chunks will have fewer terms.

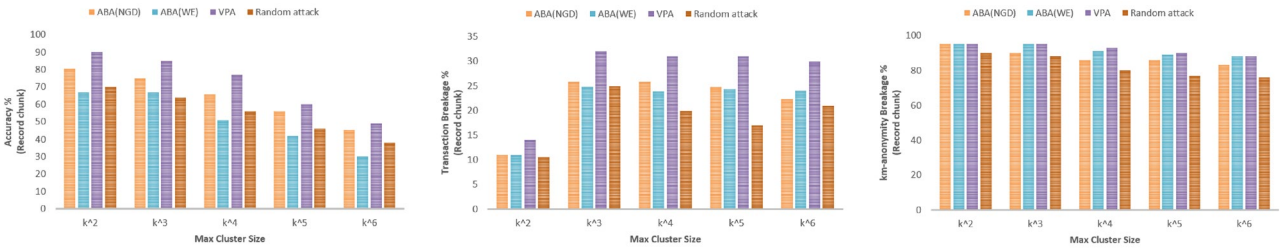
In attacking the record chunks, more transactions are exposed by the VPA than by other semantic attack methods. This occurs, because VPA requires k^m-anonymity to



(a) Accuracy **(b)** WMD

(c) Transaction breakage **(d)** k^m-anonymity breakage

Fig. 5 The effect of max cluster size on attacking methods



(a) Accuracy **(b)** Transaction breakage **(c)** k^m-anonymity breakage

Fig. 6 The effect of max cluster size on attacking record chunks

function. Therefore, because the sparsity level increases, VPA benefits from having distinct sub-records and fewer transactions per cluster (Fig. 4a).

Figure 3d illustrates k^m-anonymity breakage for changes in the density level. A higher density level shows the differences between the attacking methods. Denser datasets have more semantic relationships; therefore, this helps identify which terms need to be included from the anchoring chunks, thus affecting the overall semantic scores and reconstruction.

Our algorithms' accuracy is compared across max cluster sizes in Fig. 5a. As the number of transactions in a cluster increases with increasing sizes, the effectiveness of any attack technique is reduced. This is because of the increased probability that terms will be linked to incorrect sub-records.

In addition, increasing the size of a cluster increases the number of record chunks, making its attack more difficult. This explains the decline in accuracy across all methods (Fig. 6a).

Figure 5c illustrates the WMD percentage. The semantic similarity between the reconstructed and original transactions decreases as the cluster size increases. There are more opportunities for errors when combining terms into sub-records because of the large number of transactions in bigger clusters.

In Fig. 5b, we evaluate the effectiveness of our attack methods regarding transaction breakage. Increasing cluster sizes affects record chunks and term chunks differently. For record chunks, more transactions in a cluster reduce the number of identical sub-records in the anchoring chunk, which improves breaking rates (Fig. 6b). In contrast, increasing cluster size means more terms in the record chunks and fewer terms in the term chunks. This is because larger clusters include more terms, increasing the frequency of a term in a cluster. Consequently, these terms will be moved from term chunks to record chunks. Therefore, breaking transactions into term chunks decreases as the max cluster size increases.

Figure 5d presents the overall k^m-anonymity breakage with different max cluster sizes. As mentioned earlier, larger sizes allow for more transactions in a cluster, impacting all methods' performance and resulting in a decrease in breakage percentages. Moreover, the number of protected itemsets that can be attacked decreases.

Summary

We evaluated our proposed methods of semantic attack with different parameters and found that the ABA technique with NGD and WE had superior reconstruction accuracy and WMD results when the dataset was dense. Furthermore, in terms of transaction breakage, it also performs better than competing methods. In addition, all semantic attack techniques that employed the NGD measure outperformed their WE-based counterpart in terms of attacking record and term chunks at higher density levels. This is because the WE corpus may not include all the terms present in our transactions. Generally, even at high density levels or large k values, exploiting semantic relationships between terms can violate the privacy of a disassociated dataset. Furthermore, when the number of transactions in a cluster is fewer and the level of density is higher, the performance of the VPA method improves in most measures.

Our semantic attack performed well when the data are sparse and cluster sizes are small. Therefore, before releasing data, a data owner must examine the chosen values of disassociation parameters and evaluate if they achieve the required protection to ensure the privacy of disassociated datasets and prevent semantic attacks. In light of this, it may be useful to include our proposed method within the anonymisation procedure to enhance the privacy of the disassociation technique. A semantic attack may be performed on the disassociated transactions after disassociation and before publishing. If the semantic breakage still represents a threat, the disassociation parameter settings should be modified to find the optimal balance between privacy and data utility.

Conclusions

In this research, we investigated if the disassociation method provides sufficient protection for transaction data. We introduced a de-anonymisation method to uncover hidden links in disassociated datasets. In our attack approach, we employed semantic links between terms in an anonymised transaction dataset and utilised the VP technique to reconstruct the original transactions. Our proposed approach can reconstruct chunks with around 60% accuracy and can break over 70% of infrequent itemsets. This shows that the disassociation method may not be secure enough to protect transaction data if semantic relationships among the terms are exploited.

However, to measure semantic relationships among the terms in our approach, we employed NGD and WE. Both measurements have limitations that might affect the accuracy of semantic scoring. For example, in NGD, rare terms produce fewer pages than other terms, lowering their relatedness score. While WE is unaffected by the rarity of terms, it requires both terms to be included in the training corpus to be able to determine the semantic relationship between them.

In future work, we intend to improve the scoring stage of our approach by training a corpus on specific topics to guarantee that all terms are covered. In addition, the accuracy of attacking term chunks could be improved by creating a clustering procedure for the terms in the term chunk using the semantic relationship and associating the clustered terms with the reconstructed records.

Funding This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability The data that support the findings of this study are available from the corresponding author, Asma AlShuhail, upon reasonable request.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. AlShuhail A, Shao J. Semantic attack on disassociated transactions. In: Proceedings of the 8th International Conference on information systems security and privacy-ICISSP,, INSTICC. SciTePress, 2022; pp. 60–72.
2. Fung BC, Wang K, Chen R, Yu PS. Privacy-preserving data publishing: a survey of recent developments. ACM Comput Surv (Csur). 2010;42(4):1–53.
3. Rubinstein IS, Hartzog W. Anonymization and risk. Wash L Rev. 2016;91:703.
4. El Emam K, Dankar FK. Protecting privacy using k-anonymity. J Am Med Inform Assoc. 2008;15(5):627–37.
5. Hedegaard S, Houen S, Simonsen JG. Lair: a language for automated semantics-aware text sanitization based on frame semantics. In: 2009 IEEE International Conference on Semantic Computing. IEEE, 2009; pp. 47–52.
6. Terrovitis M, Mamoulis N, Kalnis P. Anonymity in unstructured data. In: Proc. of International Conference on Very Large Data Bases (VLDB), Citeseer. 2008.
7. Terrovitis M, Liagouris J, Mamoulis N, Skiadopoulos S. Privacy preservation by disassociation. arXiv preprint arXiv:1207.0135, 2012.
8. Cilibrasi RL, Vitanyi PM. The google similarity distance. IEEE Trans Knowl Data Eng. 2007;19(3):370–83.
9. Pennington J, Socher R, Manning CD. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on empirical methods in natural language processing (EMNLP), 2014; pp. 1532–43.
10. Smith R, Shao J. Privacy and e-commerce: a consumer-centric perspective. Electron Commer Res. 2007;7(2):89–116.
11. Sweeney L. k-anonymity: a model for protecting privacy. Internat J Uncertain Fuzziness Knowl-Based Syst. 2002;10(05):557–70.
12. Wong RC-W, Fu AW-C, Wang K, Pei J. Minimality attack in privacy preserving data publishing. In: Proceedings of the 33rd International Conference on very large data bases, 2007; pp. 543–54.
13. Cormode G, Srivastava D, Li N, Li T. Minimizing minimality and maximizing utility: analyzing method-based attacks on anonymized data. Proc VLDB Endow. 2010;13(1–2):1045–56.
14. Zhang L, Jajodia S, Brodsky A. Information disclosure under realistic assumptions: privacy versus optimality. In: Proceedings of the 14th ACM Conference on computer and communications security, 2007; pp. 573–83.
15. Farkas C, Jajodia S. The inference problem: a survey. ACM SIGKDD Explorations Newsl. 2002;4(2):6–11.
16. Turkanovic M, Druzovec TW, Hölbl M. Inference attacks and control on database structures. TEM J. 2015;4(1):3.
17. Clifton C, Marks D. Security and privacy implications of data mining. In: ACM SIGMOD Workshop on Research Issues on data mining and knowledge discovery. Citeseer, 1996; pp. 15–19.
18. Rastogi V, Suciu D, Hong S. The boundary between privacy and utility in data publishing. In: Proceedings of the 33rd International Conference on very large data bases. Citeseer, 2007; pp. 531–42.
19. Evfimievski A, Gehrke J, Srikant R. Limiting privacy breaches in privacy preserving data mining. In: Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on principles of database systems, 2003; pp. 211–22.
20. Kifer D. Attacks on privacy and Definetti's theorem. In: Proceedings of the 2009 ACM SIGMOD International Conference on management of data, 2009; pp. 127–38.
21. Basu T, Murthy C. Semantic relation between words with the web as information source. In: International Conference on pattern recognition and machine intelligence. Springer, 2009; pp. 267–72.
22. Gracia J, Mena E. Web-based measure of semantic relatedness. In: International Conference on web information systems engineering, vol 8. Springer, 2008; pp. 136–50.
23. Bouma G. Normalized (pointwise) mutual information in collocation extraction. Proc GSCL. 2009;30:31–40.
24. Sánchez D, Batet M, Viejo A. Detecting sensitive information from textual documents: an information-theoretic approach. In: International Conference on modeling decisions for artificial intelligence. Springer, 2012; pp. 173–84.
25. Staddon J, Golle P, Zimny B. Web-based inference detection. In: USENIX Security Symposium, 2007; pp. 1–16.
26. Chow R, Oberst I, Staddon J. Sanitization's slippery slope: the design and study of a text revision assistant. In: Proceedings of the 5th Symposium on usable privacy and security, 2009; pp. 1–11.
27. Chow R, Golle P, Staddon J. Detecting privacy leaks using corpus-based association rules. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge discovery and data mining, 2008; pp. 893–901.
28. Sánchez D, Batet M, Viejo A. Detecting term relationships to improve textual document sanitization. In: PACIS, 2013; p. 105.
29. Shao J, Ong H. Exploiting contextual information in attacking set-generalized transactions. ACM Trans Internet Technol (TOIT). 2017;17(4):40.
30. Loukides G, Gkoulalas-Divanis A, Malin B. Coat: constraint-based anonymization of transactions. Knowl Inf Syst. 2011;28(2):251–82.
31. JeffreyPennington R, Manning C, Pennington J, Socher R, Manning C. Glove: global vectors for word representation. In: Conference on empirical methods in natural language processing. Citeseer. Citeseer, 2014; pp. 1532–43.
32. Malvern D, Richards B. Measures of lexical richness. Encycl Appl Linguist. 2012.
33. Sra S, Dhillon I. Generalized nonnegative matrix approximations with Bregman divergences. In: Advances in neural information processing systems, vol. 18, 2005.
34. Kusner M, Sun Y, Kolkin N, Weinberger K. From word embeddings to document distances. In: International Conference on machine learning. PMLR, 2015; pp. 957–66.