iCAT+: An *Interactive Customizable Anonymization Tool* using Automated Translation through Deep Learning

Momen Oqaily, Mohammad Ekramul Kabir, Suryadipta Majumdar,Yosr Jarraya, Mengyuan Zhang, Makan Pourzandi, Lingyu Wang, and Mourad Debbabi

Abstract—Data anonymization is a viable solution for data owners to mitigate their privacy concerns. However, existing data anonymization tools are inflexible to support various privacy and utility requirements of both data owners and data users. In most cases, this limitation is due to a lack of understanding of those requirements as well as the non-customizability of the existing tools. To address this limitation, we propose *iCAT+*, which is an interactive and customizable anonymization approach. More specifically, we first automate the interpretation of data owners' and data users' textual requirements by deploying a Convolutional Neural Network (CNN) model for Natural Language Processing (NLP). Second, we introduce the concept of the *anonymization space* to model possible combinations of per-attribute anonymization primitives based on the level of privacy and utility that each primitive provides. Third, we design an ontology model that maps the translated requirements into their appropriate anonymization primitives in the defined anonymization space corresponding to the plain data. Fourth, we evaluate the efficiency and effectiveness of *iCAT+* based on both real and synthetic network data. Finally, we assess its usability through a real user study involving participants from industry and research laboratories. Our experiments show the effectiveness and efficiency of our solution (e.g., requirement translation accuracy of 99% at the data owner side and 98% at the data user side, with a computational time of around one minute for the Google cluster dataset).

Index Terms—Network data anonymization, property-preserving anonymization, anonymization space, deep learning, requirement translation.

1 Introduction

Network data has recently become an increasingly valuable asset that enables various applications for different stakeholders in many sectors [1]. At the same time, the reluctance in sharing that data, especially from the fear of sensitive information leakage, is also well-known (e.g., [2], [3]). Moreover, this reluctance is exacerbated by potential financial implications of privacy regulations (e.g., the General Data Protection Regulation (GDPR) [4]), by an increasing trend of emerging attacks (e.g., frequency analysis and data injection attacks [5]) and high profile data breaches and misuse incidents[1] [2], and by the growing availability of large-scale data analytics that might further empower the attackers.

To this end, data anonymization is a widely adopted solution for mitigating data owners' concerns [6]. On the other hand, since data users are often not interested in the plain data itself, but in its semantics [7], anonymized data also could be useful for data users to attain their goals. Nonetheless, a key challenge in applying anonymization solutions is that the effectiveness of such solutions critically and solely depends on how well the data owner makes the right choices of anonymization primitives. Such choices must achieve the significant trade-off between utility and privacy so that the data owners' sensitive information is properly hidden, whereas the desired information by the data receivers is well-preserved. On the other hand, such a choice is highly dependent on a proper understanding of all the requirements from both data owners and data users. To fulfill those requirements at a satisfactory level, a data owner needs to: (i) understand his/her own privacy requirements as well as the utility requirements of potential data users; (ii) understand the capabilities (in terms of anonymization primitives) of the available anonymization tools; (iii) map all the privacy/utility requirements correctly and consistently onto the capabilities of the anonymization tools; and (iv) select the right combination of anonymization tools for different data attributes to achieve the desired trade-off between utility and privacy.

However, since most data owners may not be familiar with all the privacy concepts and solutions, they would find those tasks challenging, if not infeasible. Additionally, they may not have much incentive to understand the utility requirements of data users. Moreover, due to the lack of an efficient and automated translator, data users may also fail to translate their requirements in an accurate way to present their demands to the data owners. As a ramification, the data owners may simply decide to withhold the datasets, as indicated in several studies (e.g., [8]). Such tendency prac-

- *M. Oqaily, M.E. Kabir, S. Majumdar, L. Wang, and M. Debbabi are with The Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada. E-mail: m_oqaily, m˙kabi, majumdar, wang and debbabi@encs.concordia.ca*
- *Y. Jarraya and M. Pourzandi are with The Ericsson Security Research, Ericsson Canada, Montreal, QC, Canada. E-mail: yosr.jarraya and makan.pourzandi@ericsson.com*
- *Mengyuan Zhang is with The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China. E-mail: Mengyuan.zhang@polyu.edu.hk*

1. https://www.identityforce.com/blog/2021-data-breaches.
2. https://www.techworld.com/security/uks-most-infamous-data-breaches-3604586/

ticed by data owners of withholding the datasets is understandable since fulfilling those tasks would demand a systematic knowledge of the search space, i.e., all possible combinations of anonymization primitives and their mapping to the privacy/utility requirements. Moreover, most existing anonymization tools (e.g., Loganon [9], Camouflage [10]) only support a limited number of choices, providing one-size-fits-all solutions, and manually mapping the privacy and utility requirements onto the tools' anonymization capabilities. In summary, there are two major gaps: (i) the first gap is between the capability of a typical data owner and the expectations of current solutions from data owners to identify the right combinations of anonymization primitives for given privacy/utility requirements; and (ii) the second gap is between the wide-range of privacy/utility requirements and limited support of anonymization capabilities by existing tools.

To fill in both gaps, our key idea is to design an automated, interactive, and customizable data anonymization framework that can translate privacy/utility requirements to identify the right combination of anonymization primitives that satisfy those privacy/utility requirements. More specifically, we first design a Convolutional Neural Network (CNN) model to automatically translate both data owners' and data users' requirements expressed in natural language (i.e., English) into their corresponding anonymization primitives. Second, we identify the different data attributes and generate possible combinations of anonymization primitives, namely, *the anonymization space*. Third, we build an ontology that develops rules for mapping the translated requirements to the anonymization space that can provide privacy-utility guarantees. Finally, we apply those mapping rules to translate requirements to the anonymization space, and provide a set of anonymization combinations that satisfy both parties' requirements. In summary, our main contributions are:

1) We propose an automated approach to translate the requirements (expressed in English) of both data owners and data users by implementing a CNN model, and mapping them onto the anonymization space through the NLP and the ontology modeling. Such automated translation and mapping of user requirements improve the usability for data users and data owners as well as reduce the potential human errors (i.e., the effectiveness of the translation of the requirements is around 98% for data users and 99% for data owners).
2) To the best of our knowledge, our notion of *anonymization space* is the first model that systematically characterizes and organizes existing anonymization primitives based on their relative capabilities in terms of privacy and utility. This model provides data owners a comprehensive and yet intuitive understanding of the available anonymization choices, and it also, for the first time, allows the data users to be actively involved in making their own decisions.
3) We design and implement an automated tool, *iCAT+*, that integrates popular anonymization primitives into a single framework, and selects and configures the proper primitives that satisfy the requirements of data owners and data users by utilizing the proposed anonymization space. Compared to most existing anonymization tools, *iCAT+* interactively provides more flexibility (access to the entire anonymization space) and better usability (automated requirement translation), and can support the largest combination of attributes and anonymization primitives.
4) We experimentally evaluate the effectiveness of *iCAT+* using both synthetic and real (e.g., Google cluster dataset [11]) data, while the usability of *iCAT+* is appraised through a user study involving participants from both industry and research labs.

A preliminary version of this work, proposing the basic idea of customizable anonymization for network data, has appeared in [12]. In this paper, we significantly improve our previous work [12] by introducing an automated translation model by leveraging deep learning to attain a more accurate and effective requirement translation as well as better user experience. More specifically, our major extensions are: redesigning the system architecture to accommodate the new CNN module and the new feedback module (Section 3); (i) deploying a CNN model that provides an automated and highly accurate language processing predictions ranging from semantic to syntactic constituent for the purpose of machine translation (Section 4); (ii) proposing a new feedback module, which triggers an option for manual interpretation if a request cannot be translated properly, and consequently, updates the ontology model with the respective interpretation of that failed request (Section 5); (iii) conducting new experiments on both real and synthetic data to measure the efficiency of the requirement translation and the deep learning technique and comparing the performance of *iCAT+* with our previous work [12] to illustrate the significant improvement (Section 6); and (iv) conducting three new case studies focusing on the size of the anonymization space, the multilevel anonymization, and the user satisfaction of the anonymized data, respectively, for one of the most popular public datasets, e.g., Google cluster dataset [11] (Section 7).

The remainder of this paper is organized as follows. Section 2 provides the motivation behind *iCAT+* and discusses the preliminaries. Section 3 presents an overview of *iCAT+* and anonymization space. Sections 4 and 5 elaborate on the requirement translation and requirement mapping steps, respectively. Section 6 presents our experimental results. Section 7 discusses case studies based on *iCAT+*. Section 8 provides detailed discussions and Section 9 reviews related works. Finally, Section 10 concludes this paper.

2 Motivation and Preliminaries

In this section, we further illustrate our motivation using an example. Additionally, we define our threat model and discuss the considered anonymization primitives.

2.1 Motivating Example

Figure 1 depicts a scenario where data owners (on the right) would like to anonymize their data using anonymization tools (in the middle) before handing over the data to the data users (on the left). Specifically, first, we consider three data users (Alice: an external auditor, Bob: a university

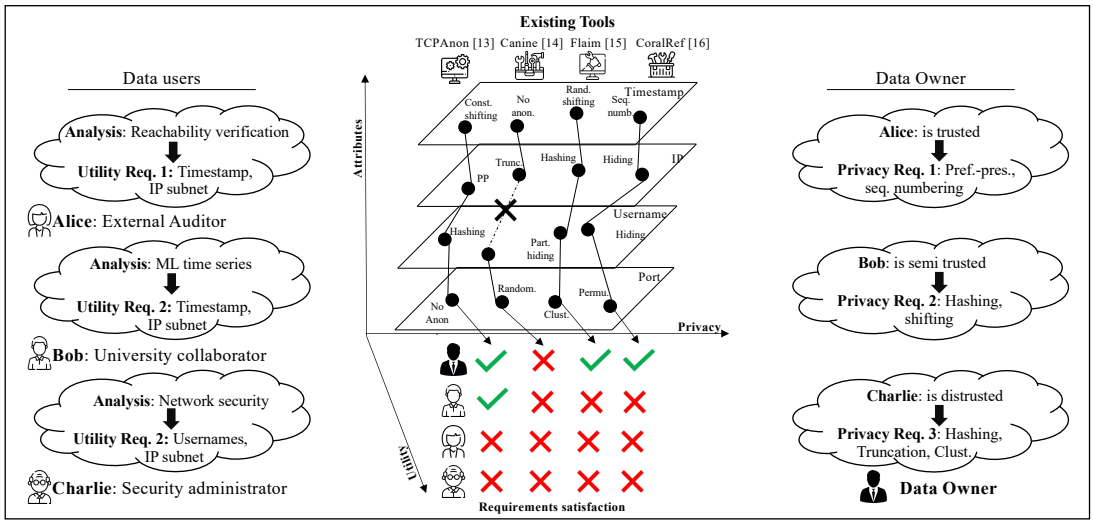


Figure 1: The motivating example.

collaborator, and Charlie: a security administrator) who intend to conduct different analysis tasks, i.e., reachability verification, ML time series analysis, and network security, respectively. Second, we consider a data owner who has different trust levels to those users: distrusted, semi-trusted, and trusted. Third, we consider four different existing anonymization tools (i.e., TCPanon [13], Canine [14], Flaim [15] and CoralRef [16]) which might be used for this scenario only if they ensure appropriate anonymization for the desired trust levels (i.e., *privacy requirements*) of the data owner as well as preserve data quality for the planned analysis tasks (i.e., *utility requirements*) of data users. However, matching both requirements and identifying the most appropriate anonymization tools might become a non-trivial task for data owners and users for the following reasons.

- Even though each data user (e.g., Alice) might have an understanding of his/her analysis tasks (e.g., reachability verification), identifying their correct and concrete utility requirements (e.g., preserving both sequences of timestamps and subnets in IPs will be needed for reachability verification) might not always be feasible (as confirmed later by our user-based experiments in Section 6). This is mainly because multiple iterations of interactions between a data owner and data user have to be performed to identify the correct utility requirements.
- Even though a data owner might be capable of understanding his/her trust level on each data user (e.g., Alice is trusted), relying on data owners for identifying concrete privacy requirements (e.g., Alice can only be given prefix-preserving and sequentially numbered data) might not be practical mainly due to the fact that real-world data owners are usually not so considerate and might simply go with whatever is suggested by a handy anonymization tool [17].
- As shown in the middle of Figure 1, most existing tools (e.g., [13], [14], [15], and [16]) only implement a small set of anonymization primitives (e.g., constant shifting, hashing) suitable for a subset of the data attributes (e.g., timestamp, IP). Furthermore, those tools are not customizable enough to accommodate specific combinations of privacy and utility requirements. As a result, most tools fall short to satisfy the requirements from data owners and data users.

To address the aforementioned challenges, we propose *iCAT+*. Intuitively, *iCAT+*: (i) automatically translates data users' utility requirements in terms of data attributes; (ii) automatically translates data owners' privacy requirements in terms of anonymization primitives; and (iii) defines the entire "anonymization space" (that maps data attributes and their related anonymization primitives) instead of covering a subset of it. We elaborate on *iCAT+* in Section 3.

2.2 Threat Model

We define the parties who are involved in the data anonymization process and their trust relationships in a more realistic approach as follows:

- **Data owner:** who has useful datasets that can be used for different purposes and is interested in protecting the privacy of his/her data to avoid any data misuse. The data owner has different trust levels to the data users, which determines the exposure of data that s/he allows.
- **Data users:** who have different intentions of using the data (e.g., auditing, research purposes), are interested in having the maximum data utility to achieve valid results. The data users trust the data owners and are willing to share their use cases with them.

In the following, we elaborate on both in-scope and out-of-scope threats.

In-scope threats. We assume that both data owners and users are willing to follow the procedure to express their requirements, while the data user is interested in obtaining output with a higher utility if the tool provides him/her with such an opportunity. Moreover, we consider the case where the data user might tamper with the learning and requirement translation process to obtain a higher utility output.

Out of scope threats. The intention of *iCAT+* is not to mitigate any weakness or vulnerability of the underlying anonymization primitives (e.g., frequency analysis, data injection attacks, or data linkage attacks). Consequently, those primitives are used as a blackbox in our data anonymization module and can be replaced by other, better primitives when available. Also, we do not consider the case where a data user uses the tool with the data owner's privileges, where s/he has more capabilities. Finally, any integrity breach of the translation in NLP techniques is beyond the scope of this work.

2.3 Anonymization Primitives

In spite of existing many data anonymization primitives in the literature, most current tools only support a limited number of primitives (a detailed review of related works is provided in Section 9). Table 1 provides a list of such common anonymization primitives, examples of plain data, and the corresponding anonymized data obtained using those primitives. However, this list is not meant to be exhaustive, and our model and methodology can be extended to include other anonymization primitives.

Primitive	Plain Data Input	Anony. Output
Prefix-preserving	IP1:**12.8.3**.4; IP2:**12.8.3**.5	IP1:**51.22.7**.33; IP1:**51.22.7**.19
Truncation	IP1:12.8.**3.4**	IP1:12.8.**X.X**
Const. Substitution	Version:2.0.1	Version:VERSION
Const. Shifting	Time1:**2019**-03-31; Time2:**2019**-03-30	Time1:**2022**-03-31; Time2:**2022**-03-30
Random Shifting	Time1:**2019**-03-31; Time2:**2019**-03-30	Time1:**2003**-03-31; Time2: **2015**-03-30
Sequ. Numbering	Time1:2019-03-31; Time2:2019-03-30	Time1:T1; Time2:T2
Partial Hiding	Time1:2019-**03-31**	Time1:2019-**X-X**
Hashing	ID:40018833	ID:H3%s2*D9
Clustering	Port1:**25**; Port2:**77**	Port1:**20**; Port2:**70**
Permutation	Port1:**25**; Port2:**77**	Port1:**77**; Port2:**25**
Randomization	Port1: **25**; Port2:**77**	Port1:**42**; Port2:**29**

Table 1: Examples of plain data inputs and their corresponding anonymized outputs.

3 An Overview of *iCAT+*

In this section, we provide overviews of the *iCAT+* approach, as well as our anonymization space, and preference up and utility down (*PU/UD*) rules.

3.1 Approach Overview

Figure 2 depicts three major processes of *iCAT+* including their corresponding steps: (i) requirements translation (Steps 1-3), (ii) anonymization space identification (Steps 4-6), and (iii) requirements mapping (Steps 7-8). In the following, we elaborate on each of them.

Requirement translation: To ameliorate the burden of both the data owners and users, *iCAT+* accepts the privacy constraints and the utility requirements as plain text in English. In Steps 1-2, it translates those requirements into a combination of anonymization primitives and data attributes using NLP. In Steps 3-6, *iCAT+* provides a feedback option to allow users to perform manual interpretation, in case the NLP fails to translate any requirement. **Anonymization space identification**: In Step 7, *iCAT+* filters and pre-processes the plain data entered by the data owner to remove undesired columns and rows. In Step 8, it extracts the total number of attributes (e.g., six columns) and their corresponding types from the processed input data (e.g., IP address, string, and timestamp). Then, *iCAT+* identifies the anonymization space based on the number of attributes and data types generated from the previous steps. **Requirement mapping**: In Step 9, based on the attribute type of each requirement, *iCAT+* maps those requirements into anonymization primitives in the anonymization space corresponding to the input data. Finally, based on the intersection between the data owner and data user requirements, *iCAT+* provides a set of anonymization combinations to the data user which also meets the data owner requirements.

Note that the CNN model is newly added to the architecture and provides automated and highly accurate language processing predictions, ranging from semantic to syntactic constituents for the purpose of machine translation. Moreover, the new feedback module, which triggers an option for manual interpretation if a request cannot be translated properly, and updates the ontology model with the respective interpretation of that failed request to improve the requirements translation process. These steps will be further detailed in Sections 4, 5.1, and 5.2, respectively.

3.2 An Overview of Anonymization Space

This section first describes the need for an anonymization space and then define our proposed anonymization space.

The Need for an Anonymization Space. There is a need to determine a systematic approach to represent and organize all the possible choices of anonymization primitives for applying on a given dataset to offer the freedom of choice to data owners for heightening the privacy level even after ensuring the acceptable utility level for users. More specifically, first, assigning a trust level for each data user and translating this trust level into a privacy requirement is not a straightforward process due to the limited capabilities available by existing tools. Second, the existing anonymization primitives (i.e., shown but not limited to in Table 1) provide a wide range of anonymization possibilities and lead to cover a large number of trust levels as listed below.

- Each data attribute may be anonymized using a different collection of the anonymization primitives (e.g., IPs may work with prefix preserving, truncation, hashing, etc., while IDs with clustering, hashing, etc., and both can be either completely hidden or given as simple text without any anonymization).
- Or, different anonymization primitives applied to an attribute may yield different levels of, and sometimes incomparable, privacy and utility (e.g., for IPs, hashing provides more privacy/less utility than prefix preserving, whereas they are both incomparable to truncation or randomization).
- Or, the data owner and data users' requirements typically involve multiple attributes, as demonstrated in Figure 1, and sometimes in a complex fashion, e.g., the data owner might say "I can only give you the data with the IPs hashed, or with the IDs clustered, but not

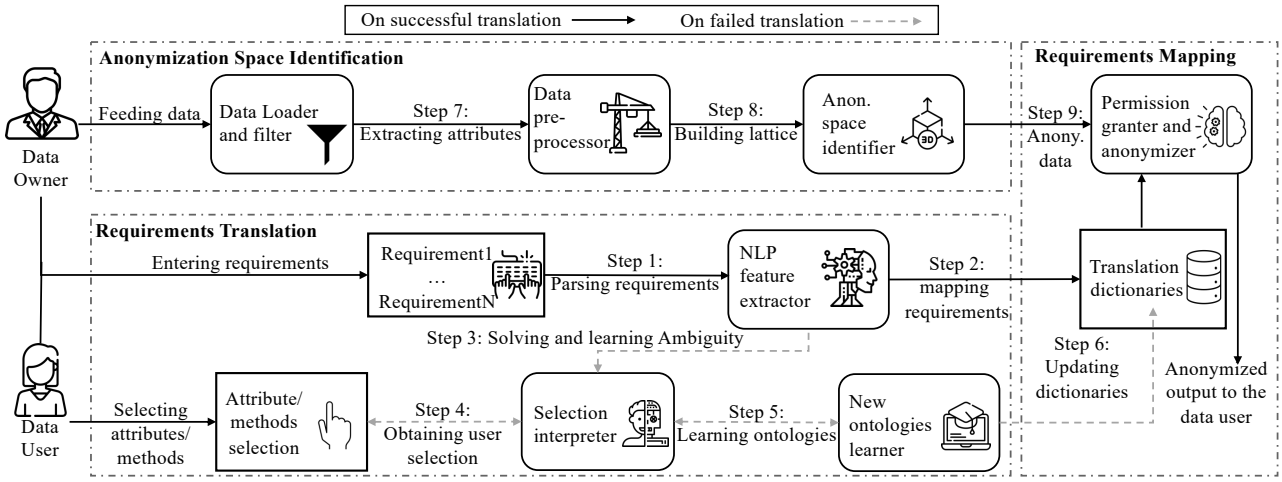


Figure 2: An overview of *iCAT+*.

both", while a data user asks "I know I may not get the data with the IPs truncated and the IDs hashed, but what would be my next best option?"

Definition of the Anonymization Space. To meet the above-mentioned needs, we propose a novel concept, namely, *anonymization space*, by considering each data attribute as a *dimension*, and each combination of anonymization primitives that can cover all the attributes as a *point* inside the *anonymization space*. Since anonymization primitives are not always comparable in terms of privacy/utility, inspired by Denning's Axioms [18], [19], we consider the collection of anonymization primitives applicable to each attribute to form a lattice [20] on their relationships in terms of privacy and utility. The product of all those lattices is the *anonymization space*. The formal definition and an example are as follows.

Definition 3.1 ***(Anonymization Space)***
Given $\mathbb{A} = \langle a_1, a_2, \ldots, a_n \rangle$ *as a set of attributes to be anonymized, and given* $F_i = \{f_1, f_2, \ldots, f_m\}(1 \leq i \leq n)$ *as the anonymization primitives set applicable to* a_i*, we define:*

- *The attribute anonymization lattice* $\mathcal{L}_i(1 \leq i \leq n)$ *as a lattice* $\langle F_i, \prec \rangle$ *where for any* $f_1, f_2 \in F_i$*, we have* $f_1 \prec f_2$ *iff* f_1 *provides better utility and more stringent privacy than* f_2 *when applied to* a_i*, and*
- *The anonymization space corresponding to* $\mathbb{A}$ *is denoted by* $\prod_{i=1}^{n} \mathcal{L}_i$*.*

Example 3.1 Figure 3.A (top) shows some examples of anonymization primitives, Figure 3.B (middle) shows their applicability (using their indices) to six attributes, and Figure 3.C (bottom) shows the six attribute anonymization lattices. Due to space limitations, we omit the anonymization space representation (which would have a size of $20,736$ different anonymization combinations).

3.3 An Overview of *PU/UD* Rules

The privacy up and utility down (PU/UD) rules are to address one of the major challenges in mapping privacy or utility requirements into anonymization primitives to ensure acceptable utility levels for all data users without infringing the data owner's privacy. Inspired by the Bell–LaPadula (BLP) model [21], in PU/UD rules, we adopt a concept of jointly enforcing the privacy and utility requirements through a simple access control mechanism. In this mechanism, data users actively participate in the anonymization process to maximize their utility levels, while this mechanism itself ensures the data owner's privacy concerns.

By considering each point (i.e., the collection of anonymization primitives) in the anonymization space as a privacy/utility *level*, the data owner's privacy requirement can be mapped to a level in such a way that anything above this level can satisfy the privacy requirement. This mapping is an automated process, and its implementation details will be discussed in Section 5. Since this approach yields higher privacy, we define this as the *privacy-up* rule. Simultaneously, a data user's requirement can be mapped to a level in this anonymization space below which any level would satisfy the utility requirement, namely, the *utility-down* rule. Hence, these *privacy-up* and *utility-down* rules can be combinedly called *PU/UD* rules. The formal definition and an example are as follows.

Definition 3.2 ***(PU/UD rules)*** *Given the set of attributes* $\mathbb{A}$*, the corresponding anonymization space* $\mathbb{AS} = \prod_{i=1}^{n} \mathcal{L}_i$*, then:*

- *The Privacy Up (PU) lattice denoted by* $L_p \in \mathbb{AS}$*, represents the nodes that have the least utility level compared to what is specified by the data owner in the privacy requirement.*
- *The Utility Down (UD) denoted by* $L_u \in \mathbb{AS}$*, represents the nodes that have the least privacy level compared to what is specified by the data user in the utility requirement.*

Respectively, any $L \in \mathbb{AS}$ *will satisfy both requirements if* $L_p \prec L$ *(PU) and* $L \prec L_u$ *(UD) are both true.*

Example 3.2 Figure 4 shows an example of anonymization space corresponding to the IP and ID attributes and the PU/UD rules for two data users, Alice and Charlie. The data owner requires hashing **(Ha)** for IPs and no anonymization **(NA)** for IDs. By the privacy-up rule, all levels inside the upper shaded area will also satisfy privacy requirements. Alice's and Charlie's utility requirements are as follows.

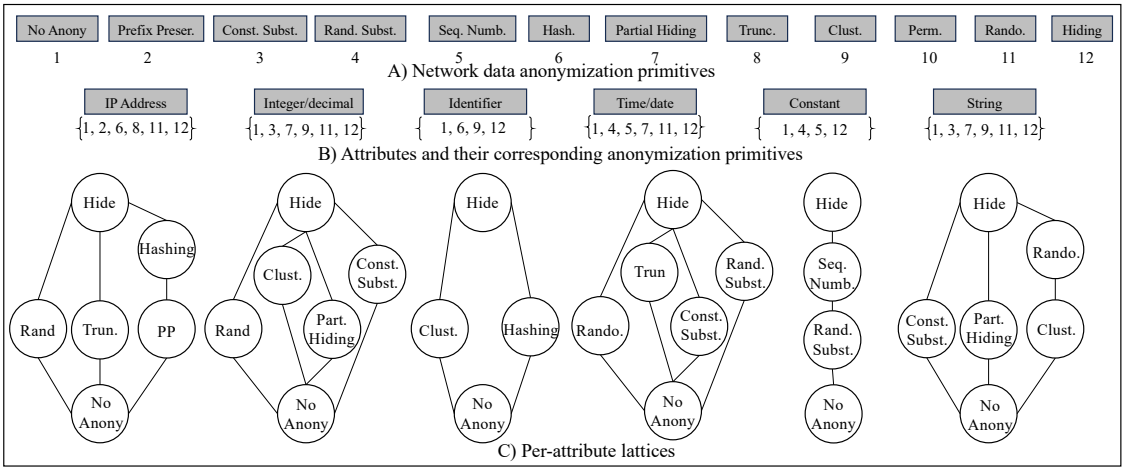


Figure 3: *An example of anonymization space:* **A**) examples of anonymization primitives with their indices, **B**) examples of data attributes and their applicable anonymization primitives, and **C**) the per-attribute anonymization lattices.

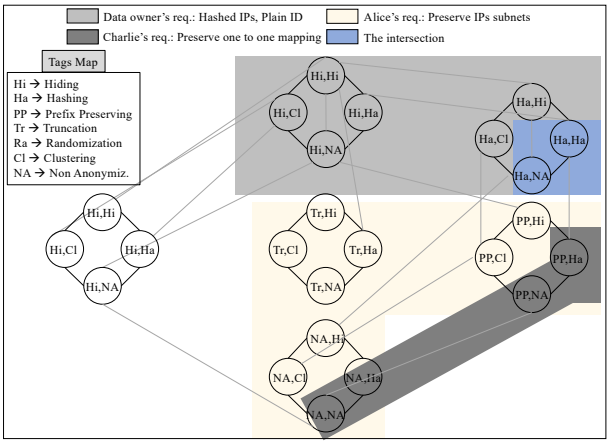


Figure 4: An example of *anonymization space* for attributes IP and ID, and the *PU/UD* rules for Alice and Charlie.

1) Charlie requires to preserve the one-to-one mapping for both IPs and IDs. Following the utility-down rule, the dark gray area highlights all the levels that satisfy Charlie's utility requirements. Also, the area with crossing lines includes all levels that satisfy both the privacy and utility requirements, i.e., ⟨**Ha**, **Ha**⟩ and ⟨**Ha**, **Na**⟩.
2) Alice requires to preserve the IP subnets. The light gray area highlights all the levels that satisfy Alice's utility requirement. Since there is no intersection between the upper shaded area and the light gray area, no level can satisfy both the privacy-up and utility-down rules, which means no anonymization primitive can satisfy both the privacy and utility requirements for Alice. However, the anonymization space makes it easy to choose an alternative level that will satisfy the privacy requirement while providing the best possible utility to Alice, e.g., ⟨**Ha**,**Na**⟩.

Note that, the privacy up and utility down (PU/UD) rules given in this paper are only meant as examples instead of universal rules since the relationship between different anonymization primitives in terms of their privacy and utility levels might may also depend on the specific context or application. For the privacy up and utility down (PU/UD) rules given in the above example, we leverage the results of an existing study [22]. Specifically, the authors in [22] provide detailed comparisons of the state-of-the-art anonymization methods (as shown in Table 1) in terms of utility, privacy, and possible attacks. We leverage such results to classify the privacy and utility levels that each primitive provides in relation to other primitives, and hence design our privacy up and utility down (PU/UD) rules.

4 Requirement Translation

This section details the requirement translation step.

4.1 Machine Translation

To ensure a user-friendly interface, *iCAT+* permits both the data owner and the data users to express their requirements in English. As a ramification, the primary challenge of translating any requirement is to understand that requirement linguistically.

To overcome this challenge, *iCAT+* leverages three pre-trained models to perform our natural language processing (NLP) tasks as follows: i) convolutional neural network (CNN) architecture [23]; ii) BERT model [24]; and iii) RoBERTa model [25]. The input of these models is an English sentence representing the requirement, and a set of language processing predictions are the expected outputs. The three models show very close results in terms of translation accuracy, however, the CNN model is the lightest-weight model in terms of size and speed, and hence is the best overall choice as discussed in the following. More details about the models' evaluation are presented in Section 6.2. For the rest of this work, we will center our machine translation based on the CNN architecture [23] as it results in the best performance/accuracy trade-off; however, in Section 8, we show general guidelines to adopt the other two models as if the dataset/user changes these results might change.

The deployed NLP architecture is effective in extracting the salient n-gram features from sentences to yield informative semantics of sentence representation, and has been reported as a popular deep-learning model that is

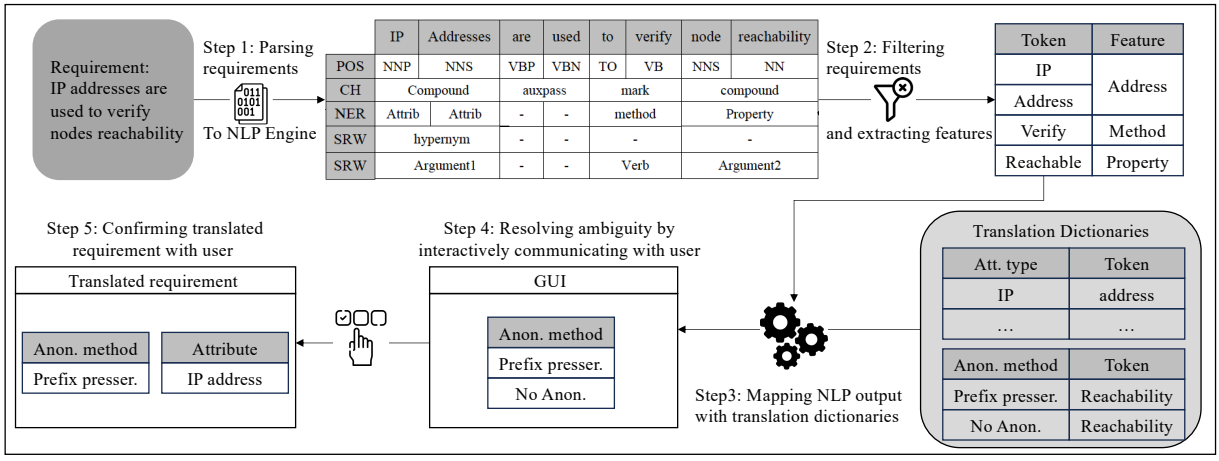


Figure 5: An example to explain the requirement translation process.

increasingly applied in sentiment analysis [26]. A detailed survey of popular deep learning models demonstrates that the representations learned from the text in [23] contain a great deal of information about meaning in and out of context [27]. The deep neural network architecture is trained in an end-to-end fashion to process the input sentence by several layers of feature extraction. In order to ensure that the features in deep layers are relevant to the task, the NN model is automatically trained by backpropagation. Our deployed architecture consists of three special layers. The first layer extracts the features for each word. The second layer extracts features from the sentence while treating it as a sequence with local and global structures. The third layer captures the most relevant features over the sentence. Unlike other NLP tools, the extracted features range from semantic to syntactic constituents. Table 2 discusses the NLP standards which are used here. In the following, we briefly discuss each layer and the model parameters:

Layer 1 - Transforming Indices into Vectors: The deployed architecture deals with raw words instead of engineered features. The first layer in the NN architecture is to map words into valued vectors, by applying the lookup-table to each of its words, for processing by following NN layers, whereas words are considered as indices in a finite dictionary of words. Note that the class label of each word in a sentence depends on a given predicate. Therefore, the NN architecture must be configured to encode which predicate is considered in the sentence. Layer 2 - Variable Sentence Length: The lookup table layer maps the original sentence into a sequence of n identically sized vectors. However, the size n of the sequence varies depending on the sentence and how many words it have. Note that, normal NNs are not able to handle sequences of variable lengths, hence a classical window approach only considers words in a window of size k around the word to be labeled. Therefore, a window layer is added so that one can extract local features in lower layers, and more global features in subsequent ones. Layer 3 - Max Overtime: A third layer that captures the most relevant features over the sentence by feeding the window layers into a "Max" Layer, which takes the maximum over the sentence for each of the output features. Note that, as the layer's output is independent of the sentence size and hence is of fixed dimension, subsequent layers can be classical NN layers. As the proposed architecture provides a way to mark the word to be labeled, it will also be able to use features extracted from all windows in the sentence to compute the label of one word of interest.

Model Parameters. The Part of Speech (POS), the Name Entity Recognition (NER) and the chunking tasks were trained with the window version with $k = 5$. Where a linear model has been chosen for the POS and NER tasks and a hidden layer of 200 units for chunking. The language model task has a window size $k = 11$, and a hidden layer of 100 units. For SRL, the network had 3 layers with three lookup tables as follows. A convolution layer with $k = 3$ and 100 hidden units with a lookup table for the words in lower case. Two hidden layers of 100 hidden units with two that encode relative distances to the word of interest and the verb. After that, those selected features are fed to a number of fully connected layers, while these layers serve as a classifier based on the learned feature. Finally, a Softmax layer is used as the output layer to predict the probabilities of each word in the target language.

As an example, Figure 5 shows how a data owner's requirement (e.g., "IP addresses are used to verify nodes reachability") is processed to obtain the attribute data type *IP* and the associated anonymization primitive *Prefix-Preserving*. Since the aforementioned requirement may have multiple interpretations for the anonymization method, the user interacts with the tool through a GUI interface to solve the issue. This will be further discussed in Section 4.2.

4.2 Ambiguity Resolution

If a particular requirement has multiple numbers of translation candidates, this may lead to an ambiguous condition, i.e., which one from those translation candidates should be chosen. The reasons behind such ambiguous situations and the corresponding solutions are as follows.

- At the requirement parsing step, due to the typos or NLP failures, the sentences entered by the user might be mistakenly parsed. As a ramification, the require-

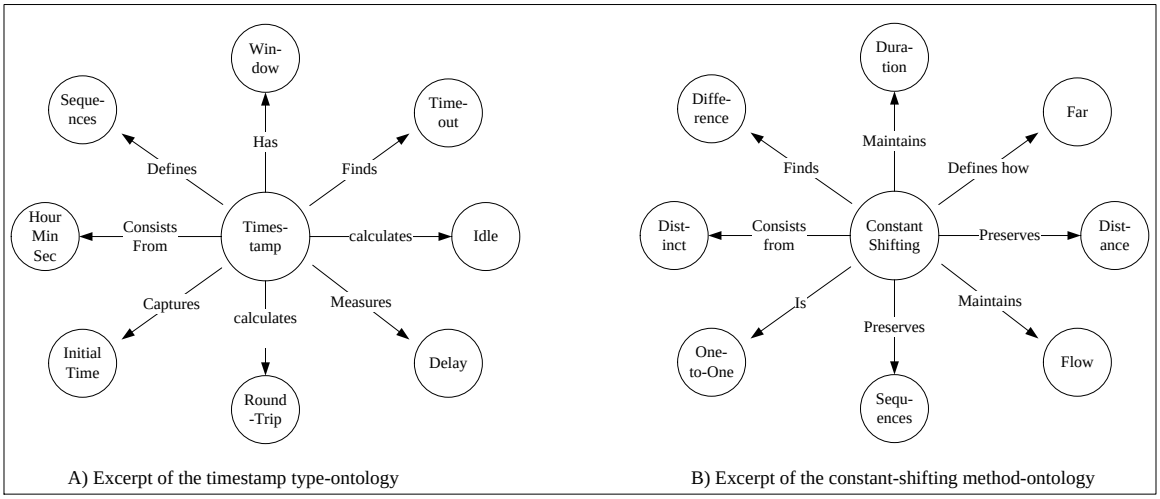


Figure 6: Ontologies of A) timestamp type-ontology and B) constant shifting method-ontology.

NLP Feature	Meaning
Part-Of-Speech (POS)	Labeling each word with a unique tag that indicates its syntactic role (i.e., plural, noun, adverb)
Chunking (CH)	Labeling a sentence with syntactic constituents such as noun or verb phrase (i.e., noun phrase (NP)).
Named Entity Recognition (NER)	Labeling atomic elements in the sentence into categories (i.e., attribute, method, property)
Semantic Role Labeling (SRL)	Giving a semantic role to a syntactic constituent of a sentence.
Semantically Related Words (SRW)	Predicting whether two words are semantically related (i.e., synonyms, holonyms, hypernym)

Table 2: A list of NLP features used in *iCAT+* and their definitions.

ment translation fails and the user has to re-enter the requirement.

- On the other hand, a particular requirement can be translated into different anonymization methods. As an example, we may consider a requirement, *Req-1: each IP address must be mapped to one IP address* and that can be satisfied with both the IP hashing and the prefix-preserving. In this case, the ambiguity solver of *iCAT+* will display a small multi-choice menu to the user, such that this ambiguity can be resolved interactively, with the click of a button. This is worthy to mention that we evaluate the user selection and propose a solution to help him/her choose the right translation from the multi-choice menu in Section 7.
- On the other hand, a requirement even can be expressed in many different ways. For example, one requirement states that the sequence of events in the logs should be maintained, while the other requires that the correlation between the logged records should be preserved. And in both requirements, the order of the data is mandatory for the analysis task and should be preserved to serve the use purpose.
- There is also a possibility that a data user's minimum requirement leads to the utility level being higher than what is allowed by the data owner according to his/her privacy requirements. In this case, *iCAT+* suggests alternative anonymization primitives that offer the closest utility level to what is specified by the data owner. A further discussion is presented in Section 5.2.2, while we describe the requirement mapping in detail.

Datasets	Category	Format	Records	# of Attr.	# of req.
DS1: Google cluster	Real	CSV	2,000	9	56
DS2: Neutron	Synth.	log	2,000	18	62
DS3: Nova	Synth.	DB	2,000	22	44
DS4: BHPOBS	Real	text	1,027	22	43

Table 3: Different datasets used in evaluating *iCAT+*.

Domain	IT				Other	
Category	Research		Industry		Research/Industry	
Expertise level	M.Sc.	Ph.D.	Junior	Senior	Junior	Senior
Percentage	23.5%	6.5%	32.5%	14.1%	8.8%	14.6%
Overall percentage	30%		46.6%		23.4%	

Table 4: participants distribution over user experience levels.

5 Anonymization Space Creation and Requirement Mapping

In this section, we describe the procedures of building the anonymization space and mapping the requirements on that space in an appropriate way to ensure the privacy of the data owner and the utilities for data users.

5.1 Anonymization Space Identification

iCAT+ identifies an anonymization space based on the data received from a data owner to provide more choices to users. Different steps of identifying such a space are as follows. To identify the Anonymization Space modeling, and hence to build the anonymization space lattice, first, *iCAT+* loads the available data from the data owner, deletes empty rows or columns and converts the dataset into data frames, and detects all the data attributes and their types based on predefined patterns. For example, the time and date format, the IP format, the IDs based on sequence numbering or predefined patterns from the data owner. Second, *iCAT+* allows the user to perform several data filtering operations

manually or automatically to remove records from data. For example, column deletion, row deletion, frequency deletion based on the number of occurrences of a text, and searched deletion based on the existence of a keyword. Thus, the anonymization space lattice is identified based on the attribute-type lattices corresponding to each data attribute in the input data. Finally, the privacy/utility access control model is generated as explained in section 3.2 based on the translation of the requirements as discussed in the following section.

5.2 Requirement Mapping

All the requirements are mapped with the anonymization primitives as follows.

5.2.1 Ontology Modeling

An ontology specifies the conceptualization of a domain in terms of concepts, attributes, relations and other distinctions that are relevant for modeling the domain. Specifically, it ensures a common understanding of information and makes explicit domain assumptions thus allowing for a better sense and use of data. In this paper, we utilize ontology modeling to define the relationship between requirements and data attributes/anonymization primitives [28].

To explain the *ontology learning* process, first, we define the following concepts for data owners and users: (i) *anony-methods*; (ii) *method-functionality*; (iii) *attribute-types*; and (iv) *attribute-synon*. The instances of the *anony-methods* are the existing anonymization primitives and the *method-functionality* instances are manually created based on the functionality and unique properties that each anonymization primitive can achieve. Moreover, the instances of the *attribute-type* concept are the given attribute types, and the *attribute-synon* instances are manually created based on the use/synonymous of each attribute type. Note that those are initially manually generated and then automatically updated by the feedback module based on the user interaction with *iCAT+*, whenever a requirement fails to translate by the NLP module as we will discuss in Section 6.2. After that, we find the relationships between those concepts' instances by defining relations between the *anony-methods* and the *method-functionality* concepts. Also, by defining relations between the *attribute-types* and the *attribute-synon* instances. As an example, Figure 6 shows that the type-ontologies related to the *timestamp attribute* type and the *method-ontology* related to the constant shifting anonymization primitive. After that, we store the resulted ontologies into two separate tables, namely, the *type-ontology* and the *method-ontology*.

5.2.2 Requirement Matching

For requirement matching, the learned ontologies are applied to the processed and the filtered requirements provided by the NLP to find the data attributes and the anonymization primitives. Every tokenized word in the processed requirement is matched with the attribute type and the anonymization method ontology tables as shown in Figure 5 and discussed as follows.

- For each tokenized word of each annotated requirement, first, the tokenized word is matched with the type ontology and then with the method ontology.
- If the tokenized words are mapped to only one record from the attribute type ontology table and one record from the attribute method ontology table, then the requirement is translated properly, and the mapper will pass to the second requirement.
- If none of the tokenized words matches any record in both the type and the method ontology tables, the word is removed from the sentence annotation table.
- If the user tokenized words fail to map to any record from the type and/or the method ontologies or if the tokenized words have multiple matching, then the mapper will return an error message to the user reporting this issue and forward this conflict to the ambiguity solving process.

5.3 Permission Granter and Anonymization

After translating all the requirements and generating the corresponding anonymization space lattice, the data users are allowed to access only that portion of the anonymization space that is approved by the data owner. Hence, *iCAT+* associates the data user identity with the privacy level specified by the data owner, and that is required to determine the anonymization sub-space assigned to them based on the PU/UD access control rules as discussed in Section 3.3. Finally, based on the granted anonymization primitives, data users can choose different anonymization combinations to anonymize the data and get the final anonymized output.

6 Evaluation of *iCAT+*

In this section, we measure *iCAT+*'s effectiveness and performance through experiments using both synthetic and real datasets. Additionally, we evaluate *iCAT+*'s usability through a user-based study with participants from both industry and academia working on data analysis.

6.1 Experimental Setup

Our experimental setup for the evaluation is as follows.

6.1.1 Dataset Specification

We consider four different datasets, i.e., DS1, DS2, DS3, and DS4 for the experimental evaluation of the *iCAT+*. DS1 is the Google cluster dataset [11] (i.e., traces from requests processed by the Google cluster management system), while DS2 is cloud logs collected from OpenStack Neutron services (i.e., the networking service of Openstack). DS3 is a database dump of the OpenStack Nova service, and DS4 is the BHP-OBS machine learning dataset [29]. We select these datasets for the following reasons: (i) the privacy constraints and requirements are already known for datasets from the industrial collaborator; (ii) these public datasets are widely used in research labs [11]. In Table 3, we provide more details about the selected datasets (i.e., category, format, number of records, etc.).

6.1.2 User-based Study Specification

To evaluate the usability of *iCAT+*, we prepared questionnaires and conducted a survey with an expanded group of participants including those who have a background that is not directly related to the field (e.g., civil engineering,

accounting, and electrical engineering). The new participants are also selected from both academia and industry, with 8.8% of them at junior level and 14.6% senior, to better evaluate the usability of iCAT+ for users with diverse backgrounds.

Participants. In this process, we considered two types of participants, i.e., data owner participants and data user participants. To solicit participants, we placed a flyer on the university campus, reached out to participants from other domains, and also sent it to our industrial collaborators. The on-campus flyer requires that participants: (i) should be able to pose clear requirements (e.g., how to use the data and what properties need to be preserved); (ii) should be able to evaluate the usefulness and usability of the data after the experiments. On the other hand, the request sent to the industry indicates that participants: (i) should be able to write their institutional privacy constraints and requirements that govern data sharing; (ii) should be able to verify whether the final anonymized output of the data meets those requirements/constraints. At the end of this data acquisition procedure, we received feedback from nine researchers from different university labs, fourteen participants from four industrial organizations and 7 participants from other departments. Table 4 summarizes the participants' experience levels for each category in percentage, where we categorize them based on their educational level and industrial experience (i.e., for Research: M.Sc. and Ph.D., and for Industry: junior and senior).

Procedures. We divided our study into four data anonymization operations based on the considered datasets and asked the participants to select one of those four; corresponding to their domain. After that, the participants had to input their requirements and interact with *iCAT+* until the anonymization operation was completed. Finally, we asked the participants to fill a post-experiment questionnaire to report the correctness or satisfaction level of the usefulness of data and the privacy constraints. We also recorded the requirements entered by the participants to evaluate the effectiveness of *iCAT+*.

6.2 Effectiveness of Requirements' Translation

In the first set of experiments, we compare the translation accuracy, time consumption, and size of the three different NLP models, namely, the CNN model [23], BERT model [24], and RoBERTa model [25], using DS1 presented in Table 3. Figure 7.A shows that the three are very close in terms of accuracy with respect to the four NLP tasks we tested (i.e., Part-Of-Speech (POS), Chunking (CH), Name-Entity-Recognition (NER), and Stop-Word-Removing (SWR)). However, the CNN model consumes the lowest time to perform the tasks as shown in Figure 7.B with respect to the other NLP models. This is mainly because the CNN is smaller in size and hence it is faster (specifically, the CNN model is around 0.5GB in size, while the other two models are almost 1.5GB each, as shown in Figure 7.C). Therefore, for the rest of this work, we will center our machine translation based on the CNN architecture [23] as it results in the best performance/accuracy trade-off. Additionally, in Section 8, we provide general guidelines for adopting the other two models.

In the second set of experiments, we calculate the percentage of correctly translated requirements to measure the effectiveness of the system (in terms of translation capability) for four different datasets depicted in Table 3. Hence, we manually investigated the recorded user's requirements and categorized the failures as follows: (i) the privacy leakage/utility loss caused by both data owners/users through wrongly chosen anonymized methods; (ii) the failures caused due to misinterpretation of *iCAT+* on either the data owners or the data user's requirements.

Figure 8.A depicts that the overall effectiveness of translating data owners' requirements is significantly high, while DS2 shows the lowest accuracy but even that is 97.1%. The primary reason behind such higher accuracy is our highly efficient CNN model for the NLP in language processing predictions and consequently assists in developing a rich ontology table. However, any failure of the NLP (e.g., typos of user's input can easily guide the NLP to make a wrong interpretation) may have a significant impact on overall performance. After finding such translation failure, *iCAT+* immediately triggers the ambiguity solver for asking a manual interpretation. This solver reduces the error rate through interactive communication with the users, where they can directly intervene in the case of any uncertain requirement. Hence, there is no failure reported from the ontology modeling mapping as depicted in Figure 8.A. Not only for data owners, but the ambiguity solver also assists in attaining high accuracy in the translation of data users' requirements as depicted in Figure 8.B.

The translation accuracy is also influenced by the number of attributes. Table 5 depicts that a higher number of attributes may lead to a higher probability of translation failure. The reason behind such a negative influence is that while the number of attributes is higher, there is a higher probability that the same attribute type may appear multiple times in the dataset and hence, causes a translation failure. As an example, the dataset DS2, in Table 3, is a cloud log dataset, and the attribute ID appears five times (i.e., project ID, tenant ID, event ID, VM ID, and host ID). The user has to be precise in writing his/her requirement to differentiate between these attributes when s/he writes that requirement which is involved with the ID attribute type. As a ramification, if a user requirement is not precise to differentiate between the attributes of the same type, the translation operation will fail because the tool will not be able to select the relevant attribute to the requirement.

Our new results show that the Ease of use, interactivity, and user-friendly category has an overall score of 5.7/7 and the No-need for support/background to use category has an overall score of 5/7. Those satisfactory results are attributed to three new features of *iCAT+*: i) *iCAT+* adds a pop-up message to explain the data anonymization primitives with examples to ensure that the user has a good understanding and makes the right selections; ii) *iCAT+* includes information about the use-cases and the anonymization output, which is accessible at any time with the press of a button; iii) The feedback module and the enhanced ambiguity solver module of *iCAT+* are continuously learning from user inputs to improve the requirement translation. Finally, Table 5 shows a comparative analysis of the number of failed requirements presented in Figure 8. We can only observe

Datasets	Total requirements	Data owner			Data user	
		Utility loss	Manual validation	No-translation	Utility loss	No-translation
DS1	56	2	3	0	4	0
DS2	62	0	2	0	7	0
DS3	44	1	2	0	2	0
DS4	43	4	4	0	5	0

Table 5: An evaluation of the failed translation of requirements.

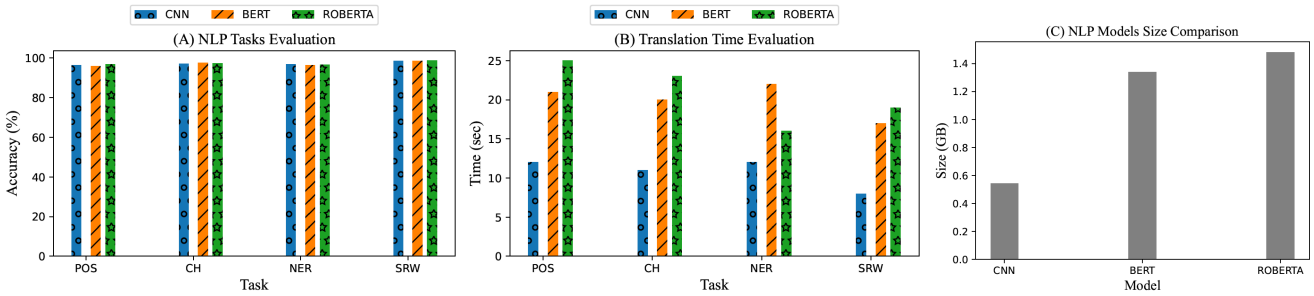


Figure 7: NLP models comparison: A) NLP Tasks evaluation; B) Time consumption evaluation; C) Size comparison.

privacy loss from the data owners' side due to failures in the NLP modeling. Utility loss could be caused by data users' sides due to an incorrect translation of data owners' requirements and a misinterpretation of the anonymization methods by data users. We discuss these issues and their potential solutions in Section 8.

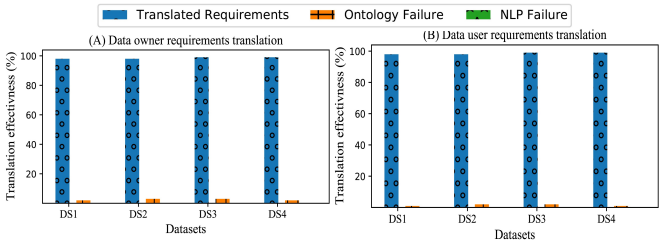


Figure 8: The effectiveness of requirement translation at A) data owner and B) data user sides.

6.3 Usability

Since there are many existing anonymization tools in practice, along with the performance comparison of *iCAT+* with these existing tools, we also intend to evaluate its acceptance probability a large number of users. For this purpose, we conduct a survey based on two questionnaires, as we mentioned earlier. The first questionnaire follows the standardized usability questionnaire [30] and consists of 19 questions. This questionnaire determines the users' satisfaction towards the services provided by the tool (e.g., whether this tool converges the views and bridges the gaps between data owners and users). On the other hand, the second one surveys the sensitivity of the attributes and the trust level in different actors used to propose privacy/utility access control mechanisms for different attributes anonymization.

The surveys are summarized in Table 6, where we categorize the evaluation criteria and rate their respective average score out of seven, as instructed in the used questionnaire [30]. The results indicate that the data users are satisfied with being a part of the anonymization process by expressing their requirements. On the other hand, the data owner participants from the industry clearly show interest in this tool for being able in owning different anonymization levels of the same input data instead of the encrypt/hide policy which they usually use. Data users also report that the tool requires some privacy expertise, especially during the implementation of the ambiguity solver. As mentioned earlier, to address such issues, we have revised our design by adding concrete examples to the tool for the anonymization primitives making them more understandable for the tool users.

6.4 Evaluation of Resource Consumption

To evaluate the overhead from different modules of *iCAT+*, we intend to estimate the required time, memory, and CPU consumption. All the experiments are performed on a machine running the MACOS 11.2 operating system equipped with Intel Quad-Core i5 CPU 3.8GHz and 16GB 2400 MHz DDR4 RAM.

Figure 9 depicts the required time, memory, and CPU consumption of the data anonymization process for four different datasets. We measure these resource consumptions for five distinct events: (i) Data loading and pre-processing (i.e., data owner side only); (ii) Anonymization space and access control matrix generation (i.e., data owner side only); (iii) Ontology mapping and learning (i.e., on both data owner and data users sides); (iv) NLP processing using the CNN results (i.e., on both data owner and data users sides); (v) The resource consumption of data anonymization (i.e., data owner side only). Figure 9 illustrates that from the data owner side, after a one-time effort to load the data, other operations have negligible consumption. On the other hand, the overhead resulting from NLP processing and data anonymization is mainly related to the original implementation of these models and does not require too long to be operated [23].

6.5 Comparison

In this section, we compare the performance of *iCAT+* with the preliminary version of this work (*iCAT* [12]). For a fair comparison, we use the same machine to perform these experiments. Figure 10 illustrates that *iCAT+* ensures a

Category	Question	Score out of 7	Average out of 7
Ease of use, interactivity, and user friendly	It was simple to use *iCAT+*	6.4	5.8
	I can effectively complete my work using *iCAT+*	5.5	
	I am able to complete my work quickly using *iCAT+*	4.6	
	I am able to efficiently complete my work using *iCAT+*	5.7	
	I feel comfortable using *iCAT+*	5.6	
	It was easy to learn to use *iCAT+*	4.7	
	I believe I became productive quickly using this system	6.2	
	The interface of this system is pleasant	6.6	
	like using the interface of this system	6.7	
Errors detecting, reporting and recovery	*iCAT+* gives error messages to fix problems	5.9	5.85
	I recover easily/quickly when I make a mistake	5.8	
iCAT+ does not need support/ background to use	It is easy to find the information I needed	5.3	5
	The information provided for *iCAT+* is easy to understand	4.4	
	The information is effective in completing the tasks	4.6	
	The information organization on *iCAT+* screens is clear	5.9	
The information provided with this system is clear (e.g., online help and other documentation)		NA	NA
This system has all the functions and capabilities I expect it to have Comment		6.2	6.2
The overall satisfaction	I am satisfied with how easy it is to use *iCAT+*	5.6	5.9
	I am satisfied with this system	.2	

Table 6: The results of usability based on a questionnaire designed following [30].

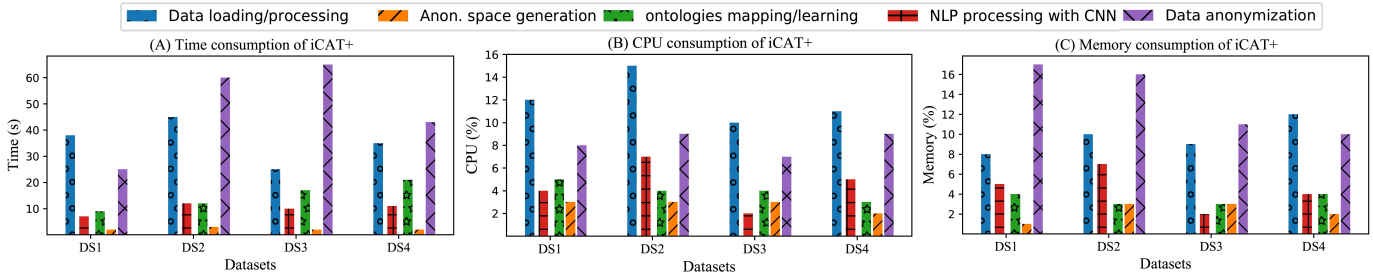


Figure 9: The resources consumption by *iCAT+*: A) Time consumption; B) CPU consumption; C) Memory consumption.

significant improvement in the effectiveness over *iCAT*. This improvement achieved in the translation process is mainly due to three unique features of *iCAT+*: (i) the CNN-based NLP model that improves the requirements translation process; (ii) the feedback module that enriches the translation dictionary with new ontologies in case of translation failure; (iii) the pop-up messages that educate the users about the anonymization methods functionalities and use. Hence, the interactive behavior of *iCAT+* assists to correct a translation failure through the feedback module and reduce the probability of dropping requests compared to the preliminary version of this tool.

On the other hand, Figure 11 depicts the resources consumption comparison between these two works for the changed processes, namely, the ontologies mapping and NLP processing. The resources consumption (i.e., time, memory, and CPU) for the ontologies mapping process has slightly increased in *iCAT+*. This increment is mainly due to the overhead resulting from the new ontologies learning process and updating the translation dictionary. On the other hand, since the newly deployed NLP processor has optimized implementation compared to the preliminary version of this tool, the NLP process resources consumption is decreased. Hence this can be depicted that though iCAT+ consumed slightly higher resources due to the ontologies learning process, the efficiency of requirements translation was significantly improved as shown in Figure 10.

7 Case Studies

In this section, using three case studies, we demonstrate the importance of our proposed anonymization space concept and we also highlight the strengths of *iCAT+* methodology over other anonymization tools.

7.1 A Study on the Size of Anonymization Space

We study the impact of three publicly available data sets, which are widely used by the researchers [31], on the anonymization space or its size. The selected data sets vary from network traces to cloud logs and IoT data, etc. The main objectives of this study are: (i) to measure the anonymization space size for different datasets; (ii) to emphasize that the anonymization decision by data owners (i.e., represented by the selection from the multi-choice menu when ambiguity occurs) can vary the privacy/utility level of the final anonymized output.

Table 7 depicts the size of the anonymization space (i.e., the total number of anonymization combinations that can apply to the corresponding dataset). Based on the data owner's privacy requirements, a sub-space is selected for the data user and finalized the information to the final anonymized output. As per our best knowledge, *iCAT+* is the first tool to allow data owners to determine the location of the final anonymized output from a utility and privacy point of view. Unlike other tools, *iCAT+* is able to quantify the final anonymized output in terms of utility and privacy

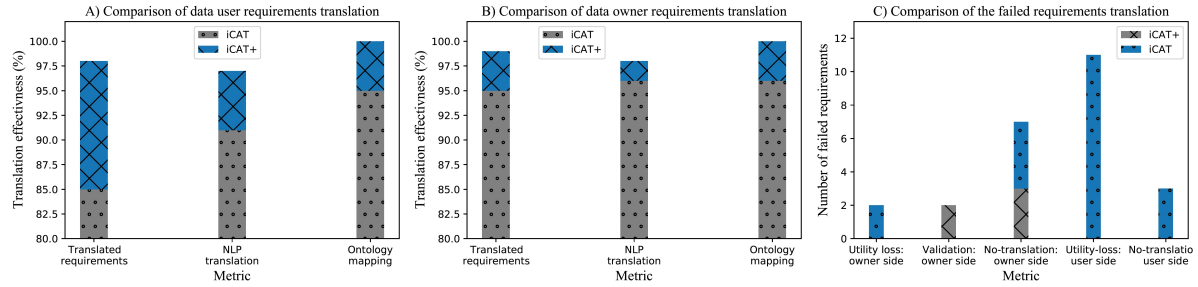


Figure 10: A comparison of the translation effectiveness between *iCAT* and *iCAT+*: A) for the data user requirements translation; B) for the data owner requirements translation; C) for the failed requirements translation.

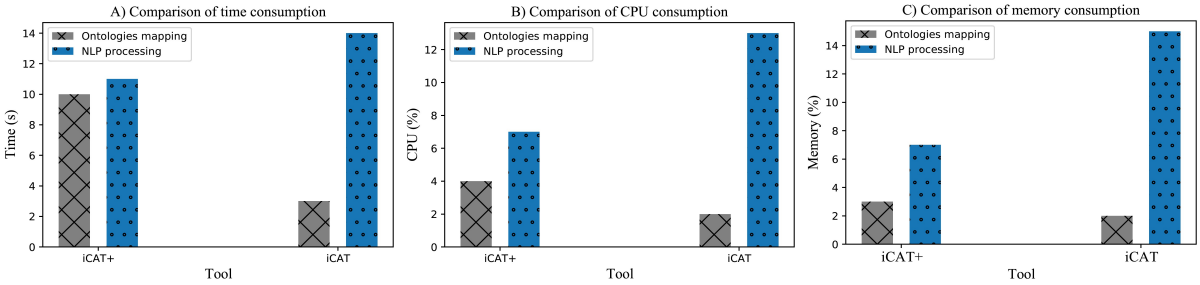


Figure 11: A comparison of the resources consumption between *iCAT* and *iCAT+*: A) for the time consumption; B) for the CPU consumption; C) for the memory consumption.

Datasets	Category	Source	Format	Number of Attributes	Size of the Anonymization Space
DS1: Google cluster	Real	UCI data repository	CSV	9	10.07 M
DS2: OpenStack Neutron	Synthetic	Generated in our lab	log	10	60.5M
DS3: OpenStack Nova	Synthetic	Generated in our lab	DB	8	1.7M
DS4: BHPOBS ML	Real	UCI data repository	text	8	1.7M
DS5: IoT data	Real	UCI data repository	CSV	11	362.8M
DS6: Network traces	Real	UCI data repository	pcap	7	280k

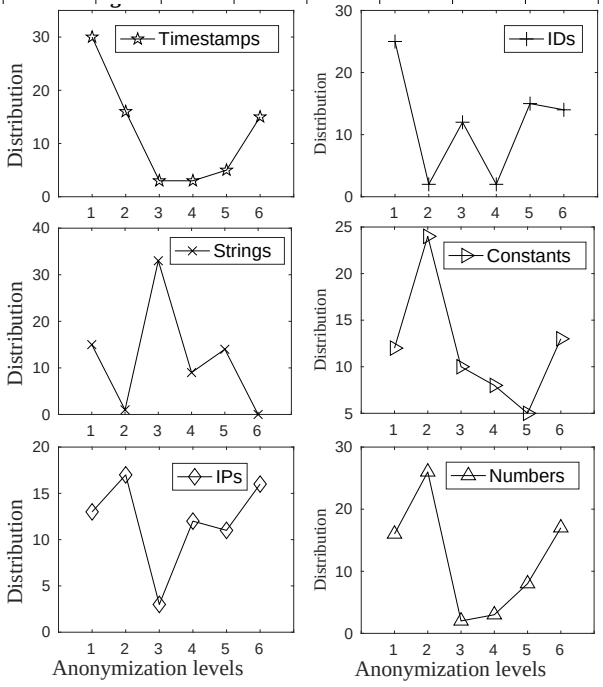
Table 7: The size of the anonymization space for the selected datasets.

concerning all other anonymization possibilities. Note that the anonymization space is never fully generated or stored by iCAT+. iCAT+ works along each dimension separately, based on the translated data user and data owner requirements, to identify the right primitive along each dimension, and then the combination of those primitives will conceptually form a chosen 'point' within the anonymization space (without actually generating or searching this anonymization space). Therefore, the cost to perform this for specific input data is proportional to the summation of the domain sizes of attributes, not their multiplication (which would have resulted in an exponential increase in the size).

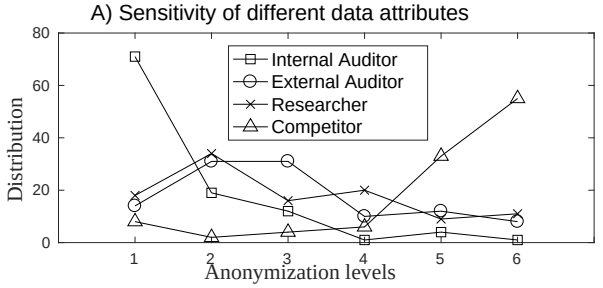
7.2 A Study on the Multi-level Anonymization

The objective of this study is to determine the need for multi-level anonymization by studying the sensitivity of the attributes and the trust level for different actors. To attain this purpose, we prepare an online questionnaire form that has been filled by participants from both academia and industry as we discussed in Section 6. This questionnaire asks participants to anonymize data given a set of anonymization primitives and different data receivers. The results of this questionnaire are listed in the table of Figure 12. To demonstrate the trend, we also apply the marginal distribution and draw the trend of each attribute and actor of this survey as depicted in Figures 12.A and 12.B. These two figures depict that the attributes and actors are associated with different sensitivity levels. The attributes e.g., *Time*, *ID*, *Constant*, and *Numbers* have similar data-sharing strategy; internal actors could have low privacy and high utility results, while competitors would be only provided with high privacy and low utility data. The main reason is that those attributes are not as sensitive as personally identifiable information, but still can leak information that can be used to stage security attacks. On the other hand, attributes *IP* and *Numbers* (e.g., salary in our survey) are considered to be sensitive attributes for all levels of actors who prefer to apply at least Level 2 anonymization on them. This can be due to sharing policies or cultural background which makes them less willing to share the information carried by those attributes. Figure 12.B confirms the trust levels of the actors through the levels of anonymization methods they are mostly assigned. Internal auditors are mostly granted Level 1 anonymization only, while competitors could only get Level 6 anonymization results. On the other hand, external auditors and researchers (generally under a non-disclosure agreement) share similar trusted levels. This shows the participants share similar visions related to the internal auditor and competitors and consider the external auditors and researchers harmless.

Attribute	Actor	Level1	Level2	Level3	Level4	Level5	Level6
Time	I	95%				5%	
	E	45%	38%	6%	6%		5%
	R	25%	50%	10%	5%		10%
	C	5%			5%	20%	70%
ID	I	80%	5%	5%		10%	
	E	5%		70%		20%	5%
	R	50%	5%	5%	10%	20%	10%
	C	10%				25%	65%
String	I	55%	5%	40%			
	E			70%	15%	15%	
	R	25%		60%	5%	10%	
	C			20%	25%	55%	
IP	I	75%	20%		5%		
	E		35%	15%	20%	20%	5%
	R		40%		40%	10%	10%
	C					25%	75%
Constant	I	40%	40%	20%			
	E		55%	20%	10%	5%	10%
	R	45%	10%	30%	5%	10%	
	C	25%	5%			15%	55%
Number	I	60%	30%			5%	5%
	E	25%	50%	5%			20%
	R	5%	50%	5%	20%		20%
	C		5%			45%	55%



A) Sensitivity of different data attributes



B) Sensitivity of different data actors

Figure 12: Users' feedback on the multi-level anonymization and its analysis: A) Sensitivity of different data attributes; B) Sensitivity of different data actors.

7.3 A Study on the Satisfaction of Anonymized Data

We also investigate the understanding of different anonymization methods by data owners mainly for two reasons. First, to check whether the selected anonymization method meets the data owners' requirements and then to show the impact of the selected anonymization primitives on the privacy/utility level. For these purposes, we take a sample of OpenStack cloud data, provide it to the user study participants and ask them to anonymize it such that their privacy requirement would be satisfied. After that, we illustrate the plain and anonymized data and present it to the participant to check whether the anonymization process configured by data owners can meet their expectations.

The benefits of this selected cloud data to perform this experiment: (i) the data syntax is simple and understandable (i.e., the data consists of IP addresses, IDs, and reachability rules), and (ii) the data can be easily represented in a visualized form. Figure 13.A shows the plain data visualization and the remaining parts of the figure depict the visualization of the anonymized data using different anonymization primitives as mentioned at the bottom of each figure.

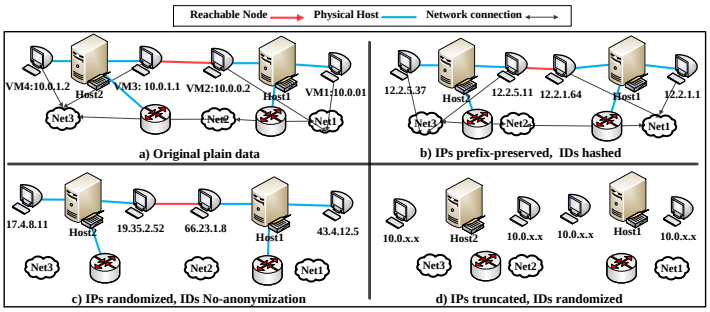


Figure 13: The visualization of plain and anonymized data using different anonymization methods.

As we can see in Figure 13, the output of the anonymized data can vary from high utility output as shown in Figure 13.B where all the properties of the original data are preserved to high privacy output as shown in Figure 13.D where all the mentioned properties are hidden. 63% of the participant has selected the anonymization method presented in Figure 13.B, and only 39% of them were satisfied with the final anonymized output. On the other hand, 73% of the participants have selected anonymization methods presented in Figures 13.B and 13.C, while 84% of them are happy about the final anonymized output. Hence this can be concluded that visualizing the data may assist the data owners to evaluate their selected anonymization primitives to evaluate their satisfaction level. We will discuss this outcome more in detail in Section 8.

8 Discussions

In this section, we make an overall discussion on various aspects of *iCAT+*.

Extensibility and Guideline for Customizing the Anonymization Primitive. We have made iCAT+ modular by making all its major components independent, as shown in Figure 2. Such modularity can allow the tool users to customize the anonymization process with less effort in

order to provide anonymization as a service. In the current version of *iCAT+*, the anonymization primitives' codes, the anonymization levels (i.e., lattices), the anonymization primitives' properties, and the NLP module can all be updated separately through configuration files. For example, in order to change the machine translation model, the user only needs to update the "MLModel.txt" configuration file with the name of the new model to use from the Huggingface AI community [32]. Moreover, in order to add a new anonymization primitive, two configuration files need to be updated as follows: i) "AddPrimiCode.txt" file with the actual code of the new anonymization primitive, and ii) "AddPrimiInfo.txt" with the name of the anonymization primitive and the privacy level it provides with respect to the existing primitives in the tool.

Protection Against Learning Phase Exploitation. An attacker may attempt to abuse the learning process of *iCAT+* by inserting records into the translation dictionary during any manual interpretation of mapping the translated requirements into a higher utility level. However, since the data owner and the data user requirements are working separately, the data user is incapable of influencing *iCAT+* to use a primitive that breaches the data owner's requirements. The *iCAT+* ensures the enforcement by; (i) during requirements translation, the ontologies for the data owner and data users, respectively, are stored and used separately; (ii) *iCAT+* does not allow the data owner to make any dataset available for sharing until a privacy level is assigned to each data attribute (by processing requirements either through NLP or manually). Consequently, if the data user tries to insert any records in the translation dictionary, it does not affect the privacy level assigned to him/her as the translation of the data owner's requirements is performed using another dictionary assigned to him/her. Moreover, a dataset will not be available for data users until an anonymization level is assigned to each data attribute.

Data Users Collaboration. *iCAT+* does not implement any countermeasure against the multi-users collaboration, i.e., two or more users may share their anonymized data among them to extract more information than what is shared by the data owner. However, the implemented PU/UD rules act as a countermeasure. These rules ensure that, even if a collaboration happens among several data users, the maximum information that they can achieve will be equal to the utility level assigned to the highest privileged user. More precisely, only the levels inside the 'utility-down' region can be extracted from the data with maximum utility without violating the data owner's privacy requirement.

Compositional Analysis. A well-known issue in anonymization is that releasing multiple views of the same data may breach privacy since an adversary can combine them. However, in our anonymization space lattice, whatever levels inside its 'privacy-up' region can be safely released because all those views contain strictly less information than the specified privacy level and hence such combining efforts do not produce any advantage. However, if the data user is mistakenly assigned different privacy levels at different times, then s/he can potentially combine those views to gain more information. However, the anonymization space lattice makes it easy for the data owner to see exactly what s/he will gain (i.e., the GLB of those levels) and take appropriate actions.

Business Potential. At present, data is becoming one of the most valuable assets, and the determiner of success in many aspects. We believe *iCAT+* can be used to provide 'data anonymization as a service' in which the data owner sets the desired privacy level for each (type of) data user, without worrying about their utility requirements. Afterward, the data users can interactively query the tool without any intervention from the data owner. The data owner can be sure that the privacy is preserved, whereas the data users can obtain as many anonymized views of the data as needed for different analyses.

Privacy Analysis. Since *iCAT+* does not propose any new anonymization primitive rather than relies on the correctness of existing primitives, the privacy/utility level provided by *iCAT+* will be the same as the anonymization primitives being used. However, *iCAT+* may mistakenly translate the data owner requirements and map them to anonymization primitives that provide lower/higher privacy levels. Hence, in our design, the data owner-side requirement translation is only intended as a suggestion, which requires further validation by the data owner to ensure the correctness of the privacy level assigned to each data attribute.

Data Linkage. We emphasize that such a limitation, deanonymizing a given dataset using publicly available data, is not due to iCAT+ as we mentioned earlier in our threat model (Section 2.2). Nonetheless, using iCAT+, the data owner will have the flexibility to assign a privacy level for each data attribute and for each data user based on their trust. Consequently, the data owner can always specify a higher privacy level for less trusted users, users with background information of the shared dataset, or for sensitive attributes (e.g., randomization, truncation, hiding, etc.). And the final anonymized data generated by iCAT+ will be more resistant to linkage attacks.

Data Visualization. The study presented in Section 7.3 indicates that data owners were not satisfied even with their own selection of anonymization methods after visualizing the anonymized data. This dissatisfaction may not occur due to a lack of understanding regarding the output of the anonymization methods rather than due to a lack of imagination on how the different anonymized attributes can be linked together. As a consequence, there might be a scope of working on the visualization of the plain and anonymized data to offer a better scenario to data owners in understanding their anonymization selection, and hence they would be able to make a comparison of the anonymized data with their inserted one.

9 Related Work

In this section, we discuss the existing works in the domain of machine learning (ML), anonymization and the domain of privacy goals mining from privacy policies. We also demonstrate the existing data anonymization tools, their limitations and provide a taxonomy based on the nature of each tool into- (i) cryptography-based anonymization tools, and (ii) replacement-based anonymization tools. Finally, we

Tool	Anonymized Fields					Anonymization Primitive					
Name	NF fields	IP	Port	Header	Payload	Pref-Pres	Hiding	Permutation	Truncation	Hashing	Shifting
AnonToo [33]	✓	✓				✓	✓	✓		✓	
CANINE [14]	✓	✓	✓			✓	✓	✓	✓		✓
CoralReef [16]	✓	✓	✓			✓	✓		✓		
Flaim [15]	✓	✓	✓			✓	✓	✓		✓	✓
IPsumdump [34]		✓		✓		✓					
NFDump [35]		✓				✓					
SCRUB [36]		✓		✓	✓		✓	✓			✓
TCPanon [13]					✓		✓				
tcpdpriv [37]		✓		✓	✓	✓	✓	✓	✓		
TCPmkpub [38]		✓		✓	✓	✓	✓		✓		
TCPurify [39]		✓			✓		✓	✓	✓		
iCAT+	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓

Table 8: Comparing existing network data anonymization tools with iCAT+.

Tool	Anonymized Fields						Anonymization Primitive					Mapping	
Name	Number	Path	ID	string	IP	Timestamp	Hiding	Substitution	Randomization	Hashing	Shifting	Look-up table	algorithm
Camouflage [10]	✓		✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Loganon [9]	✓		✓	✓	✓	✓		✓				✓	
Log-anon [9]			✓		✓			✓				✓	
Flaim [15]			✓	✓	✓	✓		✓	✓	✓	✓		✓
NLM [40]		✓	✓	✓	✓		✓		✓				✓
bsmpseu [41]	✓	✓	✓	✓	✓	✓		✓	✓		✓		✓
iCAT+	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 9: Comparing different features of existing replacement anonymization tools with *iCAT+*.

study the tools' capabilities under each taxonomy in terms of the features they provide, the fields that they cover, and the anonymization primitives they support.

In [42], the authors propose a machine learning-based model that could remove sensitive personal health information. Unlike *iCAT+*, the authors deploy an ML algorithm to act as anonymization primitive to anonymize data. However, in *iCAT+*, the ML algorithm is used to help in understanding the users' needs and translate them into the appropriate anonymization primitives only. Moreover, we link the requirements translation phase in *iCAT+* to the privacy goals extraction from the privacy policies mining field, as we are aiming the same interest in inferring requirements from the natural English text. On the other hand, in [43], [44], the authors introduce the goal-based requirements analysis method (GBRAM) and heuristics to extract goal specifications from the text. Then, they apply GBRAM to mine privacy goals from privacy policies. In [17], the authors report results from three experiments aimed at assessing the potential of crowdsourcing requirements extraction to non-experts. The authors show the cost, efficiency, and effectiveness of that task and conclude that using NLP techniques, the cost decreases, and the requirements coverage increases compared to manual extraction by trained experts. A combination of crowdsourcing and NLP is implemented in [45], while the authors introduce and evaluate a method that combined crowdsourcing and NLP to extract goals from privacy policies. Their analysis depicts that crowd workers can provide human interpretations while that are still beyond the state of the art in NLP and the NLP can provide a cost-effective and more effective goals extraction. As per our best knowledge, unlike all existing works, *iCAT+* deploys the extracted privacy goals from the privacy policy in data anonymization.

To distinguish *iCAT+*, we present our taxonomy of existing data anonymization tools. Table 8 compares those tools according to the anonymized fields (e.g., IP, header, port, etc.) and the anonymization primitives they use. As shown in Table 8, except *iCAT+*, none of those tools can support all the attributes or anonymization primitives (let alone the flexibility for customization), nor can take users' requirements to understand their privacy and utility needs. The cryptography-based anonymization tools are considered as the first taxonomy, whereas most of the existing tools under this taxonomy use cryptography-based anonymization primitives; such as prefix-preserving, shifting, hashing, and permutation. Existing tools in this category are used to anonymize network traces and mainly anonymize the TCP header. However, some of those tools support live interfaces anonymization to anonymize the data in a running-time manner. Moreover, tools under this taxonomy provide higher privacy output and are well known to be more user-friendly as the tool user does not require to have good knowledge about the anonymization primitives.

The second taxonomy is the replacement-based anonymization tools, while the existing tools in this category deal mainly with log files and anonymized data by replacing the sensitive attributes (e.g., passwords, system logs, files paths, etc.) in the log with some values predefined by the user in the so-called rule-file or generated using deterministic cryptography algorithms. The rule file contains patterns used by the tool to perform pattern matching and the conversion state of the anonymization can be stored in a look-up table. Table 9 compares these tools in terms of anonymized fields, anonymization primitives used, and how the mapping is achieved. This category of anonymization provides a higher utility output because it preserves some property of the original data (e.g., equality, format, order, etc.). However, this is also susceptible to de-anonymization attacks, known as semantic attacks (e.g., frequency analysis, injection, and shared text matching attacks). Moreover, those tools are generally not user-friendly and require knowledge about conducting tool-based search patterns and managing the conversion state of the anonymized data.

10 Conclusion

Due to a lack of understanding of the requirements as well as the non-customizability of the existing anonymization tools make this inflexible and hence inefficient to support various privacy and utility requirements of both data owners and data users. To address these issues, in this paper, we proposed an interactive and customizable data anonymization tool, namely, *iCAT+*, which takes user requirements in English, automatically processes those requirements using an NLP technique, and addresses the flexibility limitations of most existing tools by creating a customizable anonymization space. *iCAT+* can ensure the active participation of data users in making their own decisions. We leveraged a CNN-based NLP to make the requirements translation process automated. Since, due to typos, the designed NLP may fail to translate any requirement, *iCAT+* can trigger on a feedback module to accept the manual interpretation. We made an extensive analysis based on both real and synthetic data to evaluate our proposed solution and formally achieved higher effectiveness (e.g., 98% of users' requirements were correctly translated), while the decision-making time was significantly small (e.g., 64 seconds). In addition, we conducted several user surveys and obtained quite positive feedback from the tool users who participated from both industry and academia.

Future Work. As the future direction, we plan to investigate the possibility of developing a public interface for *iCAT+*, where data owners can give access and specify trust levels to users who will interact with the tool to get their anonymized data. We also intend to provide an interface to add other primitives to allow tool users to integrate additional anonymization primitives into the tool.

Acknowledgment. We thank the reviewers for their valuable comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada and Ericsson Canada under the Industrial Research Chair in SDN/NFV Security and the Canada Foundation for Innovation under JELF Project 3859.

References

[1] C. Tenopir, S. Allard, K. Douglass, A. U. Aydinoglu, L. Wu, E. Read, M. Manoff, and M. Frame, "Data sharing by scientists: practices and perceptions," *PLoS one*, 2011.

[2] Marty Swant, "People are becoming more reluctant to share personal data, survey reveals," 2021, available at: t.ly/tydm.

[3] Josh D'Addario, "New survey finds british businesses are reluctant to proactively share data," 2020, available at: https://theodi.org/article/new-survey-finds-just-27-of-british-businesses-are-sharing-data/.

[4] EU General Data Protection Regulation, "Fines and Penalties," 2018, available at: https://www.gdpreu.org/compliance/fines-and-penalties/.

[5] T. Brekne, A. Årnes, and A. Øslebø, "Anonymization of ip traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies," in *PET*. Springer, 2005, pp. 179–196.

[6] A. Majeed and S. Lee, "Anonymization techniques for privacy preserving data publishing: A comprehensive survey," *IEEE Access*, 2020.

[7] G. Cormode and D. Srivastava, "Anonymized data: generation, models, usage," in *ICMD*, 2009.

[8] W. Zhang, Y. Lin, S. Xiao, J. Wu, and S. Zhou, "Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing," *IEEE Transactions on Computers*, 2015.

[9] Sys4 consults, "a generic log anonymizer," 2018, available at: https://github.com/sys4/loganon.

[10] IMPREVA, "Camouflage data masking," 2018, available at: https://www.imperva.com/products/data-security/data-masking/.

[11] Google, "Traces from Requests Processed by Google Cluster Management System," 2019, available at: https://github.com/google/cluster-data.

[12] M. Oqaily, Y. Jarraya, M. Zhang, L. Wang, M. Pourzandi, and M. Debbabi, "icat: An interactive customizable anonymization tool," in *ESORICS*. Springer, 2019.

[13] Franceesco Gringoli, "Tcpanon tool," 2019, available at: http://netweb.ing.unibs.it/~ntw/tools/tcpanon/.

[14] Y. Li, A. Slagell, K. Luo, and W. Yurcik, "Canine: A combined conversion and anonymization tool for processing netflows for security," in *International conference on telecommunication systems modeling and analysis*, vol. 21, 2005.

[15] A. J. Slagell, K. Lakkaraju, and K. Luo, "Flaim: A multi-level anonymization framework for computer and network logs." in *LISA*, vol. 6, 2006, pp. 3–8.

[16] D. Moore, K. Keys, R. Koga, E. Lagache, and K. C. Claffy, "The coralreef software suite as a tool for system and network administrators," in *Proceedings of the 15th USENIX conference on System administration*. USENIX Association, 2001, pp. 133–144.

[17] T. D. Breaux and F. Schaub, "Scaling requirements extraction to the crowd: Experiments with privacy policies," in *IEEE IREC*. IEEE, 2014.

[18] D. E. Denning, "A lattice model of secure information flow," *Commun. ACM*, vol. 19, no. 5, pp. 236–243, May 1976. [Online]. Available: http://doi.acm.org/10.1145/360051.360056

[19] R. S. Sandhu, "Lattice-based access control models," *Computer*, 1993.

[20] T. Donnellan, *Lattice Theory*. Oxford: Pergamon press, 1968.

[21] E. D. Bell and J. L. La Padula, "Secure computer system: Unified exposition and multics interpretation," Bedford, MA, 1976. [Online]. Available: http://csrc.nist.gov/publications/history/bell76.pdf

[22] A. Majeed and S. O. Hwang, "A generic approach towards enhancing utility and privacy in person-specific data publishing based on attribute usefulness and uncertainty," *Electronics*, 2023.

[23] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[26] V. Shankar and S. Parsana, "An overview and empirical comparison of natural language processing (nlp) models and an introduction to and empirical application of autoencoder models in marketing," *Journal of the Academy of Marketing Science*, 2022.

[27] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artificial Intelligence Review*, 2020.

[28] R. Abbasi, P. Martinez, and R. Ahmad, "An ontology model to represent aquaponics 4.0 system's knowledge," *Information Processing in Agriculture*, vol. 9, no. 4, pp. 514–532, 2022.

[29] UCIMLR, "Burst Header Packet Flooding Attack on Optical Burst Switching Network Data Set," 2019, available at: https://archive.ics.uci.edu/ml/datasets/.

[30] A. Assila, H. Ezzedine *et al.*, "Standardized usability questionnaires: Features and quality focus," *eJCIST*, 2016.

[31] UCI, "Machine Learning Repository," 2020, available at: https://archive.ics.uci.edu/ml/datasets.php.

[32] H. A. community, "Huggingface ai community 2023." https://huggingface.co/.

[33] M. Foukarakis, D. Antoniades, S. Antonatos, and E. P. Markatos, "Flexible and high-performance anonymization of netflow records using anontool," in *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*. IEEE, 2007, pp. 33–38.

[34] Eddie Kohler, "ipsumdump tool," 2015, available at: https://read.seas.harvard.edu/~kohler/ipsumdump/.

[35] P. Haag, "Nfdump," *Available from World Wide Web: http://nfdump.sourceforge. net*, 2010.

[36] W. Yurcik, C. Woolam, G. Hellings, L. Khan, and B. Thuraisingham, "Scrub-tcpdump: A multi-level packet anonymizer demonstrating privacy/analysis tradeoffs," in *SecureComm 2007*. IEEE, 2007.
[37] Greg Minshall of Ipsilon Networks, "Tcpdpriv," 2005, available at: http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html.
[38] R. Pang, M. Allman, V. Paxson, and J. Lee, "The devil and packet trace anonymization," *ACM SIGCOMM*, 2006.
[39] Ethan Blanton, "Tcpurify tool," 2019, available at: https://web.archive.org/web/20140203210616/irg.cs.ohiou.edu/~eblanton/tcpurify/.
[40] Kayaalp, M., Sagan, P., Browne, A.C., McDonald, "Nlm-scrubber," 2018, available at: https://scrubber.nlm.nih.gov/files/.
[41] Konrad Rieck , "Pseudonymizer for solaris audit trails," 2018, available at: http://www.mlsec.org/bsmpseu/bsmpseu.1.
[42] G. Szarvas, R. Farkas, and R. Busa-Fekete, "State-of-the-art anonymization of medical records using an iterative machine learning framework," *Journal of the American Medical Informatics Association*, vol. 14, no. 5, pp. 574–580, 2007.
[43] R. J. Abbott, "Program design by informal english descriptions," *Communications of the ACM*, vol. 26, no. 11, pp. 882–894, 1983.
[44] A. I. Antón and J. B. Earp, "A requirements taxonomy for reducing web site privacy vulnerabilities," *Requirements engineering*, vol. 9, no. 3, pp. 169–185, 2004.
[45] J. Bhatia, T. D. Breaux, and F. Schaub, "Mining privacy goals from privacy policies using hybridized task recomposition," *TOSEM*, 2016.

Momen Oqaily Momen Oqaily is currently a Ph.D in Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, Canada. Oqaily has bachelor in network engineering and security from JUST university, Master's in information system security from Concordia university. His research mainly focuses on privacy-preserving security auditing, cloud security auditing and Network Function Virtualization (NFV) security.

Mohammad Ekramul Kabir is currently working as a Horizon postdoctoral research fellow in CIISE at Concordia University, Montreal, Canada. He has his PhD on Information and Systems Engineering from Concordia University in May' 2021. He has received the B.Sc. and M.S. degree in Applied Physics, Electronics and Communication engineering from University of Dhaka, Bangladesh. His research interests include green, smart and secure charging of electric vehicle, cloud/edge computing security and applications of artificial intelligence.

Suryadipta Majumdar Suryadipta Majumdar is currently an Assistant Professor in Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, Canada. Previously, Suryadipta was an Assistant Professor in the Information Security and Digital Forensics department at University at Albany - SUNY, USA. He received his Ph.D. on cloud security auditing from Concordia University. His research mainly focuses on cloud security, Software Defined Network (SDN) security and Internet of Things (IoT) security.

Yosr Jarraya Yosr Jarraya is currently a researcher in security at Ericsson focusing on security and privacy in cloud, SDN, and NFV. Previously, she was awarded a postdoctoral fellowship at the same company. Before that, she was a Research Associate at Concordia University, Montreal. She received a Ph.D. in Electrical and Computer Engineering from Concordia University. She has several patents granted or pending. She co-authored two books and more than 40 research papers on topics including cloud security, data anonymization, network and software security, formal verification, and SDN.

Mengyuan Zhang Dr. Mengyuan Zhang received her Ph.D. degree in Information and Systems Engineering from Concordia University, Montreal, Canada. She is currently a research assistant professor in the Department of Computing at The Hong Kong Polytechnic University. Previously, she was an experienced researcher at Ericsson Research, Montreal, Canada. Her research interests include security metrics, attack surface, cloud computing security, and applied machine learning in security. She has published more than 20 research papers and two book chapters on the aforementioned topics in peer-reviewed international journals and conferences such as IEEE Transactions on Information Forensics and Security (TIFS), IEEE Transactions on Dependable and Secure Computing (TDSC), CCS, and Esorics. Her paper on detecting common attack surface received the best paper award in 33rd Annual IFIP WG 11.3 Working Conference on Data and Application Security (DBSec). She also regularly serves as a reviewer for several major journals in information security.

Makan Pourzandi Makan Pourzandi is a research leader at Ericsson, Canada. He received his Ph.D. degree in Computer Science from University of Lyon I Claude Bernard, France and M.Sc. in parallel computing from École Normale Supérieure de Lyon, France. He is the co-inventor of 19 granted US patents and more than 65 research papers in peer-reviewed scientific journals and conferences. His current research interests include security, cloud computing, software security engineering, cluster computing, and component-based methods for secure software development.

Lingyu Wang Lingyu Wang is a Professor at the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Montreal, Canada. He holds the NSERC/Ericsson Senior Industrial Research Chair in SDN/NFV Security. He received his Ph.D. degree in Information Technology in 2006 from George Mason University. His research interests include cloud computing security, SDN/NFV security, security metrics, software security, and privacy. He has co-authored five books, two patents, and over 120 refereed conference and journal articles at reputable venues including TOPS, TIFS, TDSC, TMC, JCS, S&P, CCS, NDSS, ESORICS, PETS, ICDT, etc.

Mourad Debbabi Mourad Debbabi is a Full Professor at the Concordia Institute for Information Systems Engineering and Interim Dean at the Gina Cody School of Engineering and Computer Science. He holds the NSERC/HydroQuebec Thales Senior Industrial Research Chair in Smart Grid Security and the Concordia Research Chair Tier I in Information Systems Security. He is also the President of the National Cyber Forensics and Training Alliance (NCFTA) Canada. He is a member of CATAAlliance's Cybercrime Advisory Council. He serves/served on the boards of Canadian Police College, PROMPT Québec and Calcul Québec. He is the founder and one of the leaders of the Security Research Centre at Concordia University. Dr. Debbabi holds Ph.D. and M.Sc. degrees in computer science from Paris-XI Orsay, University, France. He published 5 books and more than 300 peer-reviewed research articles in international journals and conferences on cyber security, cyber forensics, smart grid, privacy, cryptographic protocols, threat intelligence generation, malware analysis, reverse engineering, specification and verification of safetycritical systems, programming languages and type theory. He supervised to successful completion 32 Ph.D. students, 76 Master students and 14 Postdoctoral Fellows. He served as a Senior Scientist at the Panasonic Information and Network Technologies Laboratory, Princeton, New Jersey, USA; Associate Professor at the Computer Science Department of Laval University, Canada; Senior Scientist at General Electric Research Center, New York, USA; Research Associate at the Computer Science Department of Stanford University, California, USA; and Permanent Researcher at the Bull Corporate Research Center, Paris, France.