Université de Montréal

Titre du mémoire

Towards Privacy Preserving Cooperative Cloud based Intrusion Detection Systems

par

Anirudh Mitreya Kothapalli

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures et postdoctorales en vue de l'obtention du diplôme de Maître ès en sciences (M.Sc.) Informatique

Août 2020

Université de Montréal

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Ce mémoire intitulé

Towards Privacy Preserving Cooperative Cloud based Intrusion Detection Systems

Présenté par

Anirudh Mitreya Kothapalli

A été évalué par un jury composé des personnes suivantes

Gilles Brassard
Président-rapporteur

Esma Aimeur
Directeur de recherche

Louis Salvail
Membre du jury

Résumé

Les systèmes infonuagiques deviennent de plus en plus complexes, dynamiques et vulnérables aux attaques. Par conséquent, il est de plus en plus difficile pour qu'un seul système de détection d'intrusion (IDS) basé sur le cloud puisse repérer toutes les menaces, en raison des lacunes de connaissances sur les attaques et leurs conséquences. Les études récentes dans le domaine de la cybersécurité ont démontré qu'une coopération entre les IDS d'un nuage pouvait apporter une plus grande efficacité de détection dans des systèmes informatiques aussi complexes. Grâce à cette coopération, les IDS d'un nuage peuvent se connecter et partager leurs connaissances afin d'améliorer l'exactitude de la détection et obtenir des bénéfices communs. L'anonymat des données échangées par les IDS constitue un élément crucial de l'IDS coopérative. Un IDS malveillant pourrait obtenir des informations confidentielles d'autres IDS en faisant des conclusions à partir des données observées. Pour résoudre ce problème, nous proposons un nouveau système de protection de la vie privée pour les IDS en nuage. Plus particulièrement, nous concevons un système uniforme qui intègre des techniques de protection de la vie privée dans des IDS basés sur l'apprentissage automatique pour obtenir des IDS qui respectent les informations personnelles. Ainsi, l'IDS permet de cacher des informations possédant des données confidentielles et sensibles dans les données partagées tout en améliorant ou en conservant la précision de la détection. Nous avons mis en œuvre un système basé sur plusieurs techniques d'apprentissage automatique et de protection de la vie privée. Les résultats indiquent que les IDS qui ont été étudiés peuvent détecter les intrusions sans utiliser nécessairement les données initiales. Les résultats (c'est-à-dire qu'aucune diminution significative de la précision n'a été enregistrée) peuvent être obtenus en se servant des nouvelles données générées, analogues aux données de départ sur le plan sémantique, mais pas sur le plan synthétique.

Mots-clés : Systèmes infonuagiques, Cyber-attaques, Intimité, Système de détection d'intrusion, IDS coopératif.

Abstract

Cloud systems are becoming more sophisticated, dynamic, and vulnerable to attacks. Therefore, it's becoming increasingly difficult for a single cloud-based Intrusion Detection System (IDS) to detect all attacks, because of limited and incomplete knowledge about attacks and their implications. The recent works on cybersecurity have shown that a co-operation among cloud-based IDSs can bring higher detection accuracy in such complex computer systems. Through collaboration, cloud-based IDSs can consult and share knowledge with other IDSs to enhance detection accuracy and achieve mutual benefits. One fundamental barrier within cooperative IDS is the anonymity of the data the IDS exchanges. Malicious IDS can obtain sensitive information from other IDSs by inferring from the observed data. To address this problem, we propose a new framework for achieving a privacy-preserving cooperative cloud-based IDS. Specifically, we design a unified framework that integrates privacy-preserving techniques into machine learning-based IDSs to obtain privacy-aware cooperative IDS. Therefore, this allows IDS to hide private and sensitive information in the shared data while improving or maintaining detection accuracy. The proposed framework has been implemented by considering several machine learning and privacy-preserving techniques. The results suggest that the consulted IDSs can detect intrusions without the need to use the original data. The results (i.e., no records of significant degradation in accuracy) can be achieved using the newly generated data, similar to the original data semantically but not synthetically.

Keywords: Cloud Systems, Cyber Attacks, Privacy, Intrusion Detection System, Cooperative IDS.

Table of Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

ARX	Anonymization Tool
CP	Cloud Provider
DoS/DDoS	Denial-of-Service/Distributed Denial-of-Service
DP	Differential Privacy
DT	Decision Trees
GANN	Generative Adversarial Neural Network
IDS	Intrusion Detection System
IT	Information Technology
KA	K-Anonymity
KNN	K-Nearest Neighbor
LD	L-Diversity
ML	Machine Learning
NIDS	Network Intrusion Detection System
R2L	Remote to Login
RF	Random Forest
SVM	Support Vector Machine
TC	T-Closeness
U2R	User to Root
VANet	Vehicular Ad-Hoc Networks
VM	Virtual Machine

Acknowledgments

I would like to thank my research director, Prof Esma Aïmeur, who has provided the necessary guidance to pursue this project to completion.

I would like to thank Dr. Adel Abusitta for his continued support and supervision throughout the dissertation. Thank you for believing in my potential and letting me explore Cloud Computing and Privacy. Without your guidance and support, this dissertation would not have been possible.

Furthermore, I would like to thank all my colleagues in the research lab at Université de Montreal.

I would like to thank my parents for their unconditional support throughout my MSc and for providing me with the opportunity to follow this path. Thank you for believing in me.

Chapter 1 – Introduction

In this chapter, we present the context of our research work and define the problems addressed in this thesis. We also present the corresponding research questions and finally identify the objectives of our research work. We present the background and related work in the next chapter (Chapter 2).

1.1 Problem Statement

The explosive rise[1] in cloud computing solutions and technologies, such as Amazon Web Services, Google Cloud and Microsoft Azure, is currently driven by the idea that it is economically profitable as it enables companies to streamline expenditure on infrastructure (Low, Chen, & Wu, 2011). Companies and governments are expecting to transfer their IT solutions to the cloud, if not yet (Oliveira, Thomas, & Espadanal, 2014).

With the complex, dynamic, and heterogeneous architecture of the cloud, it has become increasingly difficult for a traditional cloud-based intrusion detection system (IDS) to detect all attacks (Rowland, 2002). A single intrusion into a heterogeneous system may take various forms that are semantically but not synthetically similar.

Another problem with the heterogeneity of the cloud is that recent intrusions have evolved, becoming more sophisticated and difficult to detect. The attackers are becoming experts in creating and launching complex intrusions and concealing their malicious behavior (Herrington & Aldrich, 2013). Thus, it has become difficult for a single IDS to detect all attacks due to limited knowledge about intrusion patterns (Liao, Lin, Lin, & Tung, 2013).

A collaboration among cloud-based IDSs has proven its efficiency (i.e., accuracy) in identifying new and complex attacks (Bakshi & Dujodwala, 2010; Zargar, Takabi, & Joshi, 2011; Zhou, Leckie, & Karunasekera, 2010). By collaborating, IDSs in different locations or those

[1] https://www.forbes.com/sites/louiscolumbus/2019/04/07/public-cloud-soaring-to-331b-by-2022-according-to-gartner/#76dba21b5739

belonging to different Cloud Providers (CPs) may cooperate to use each other's expertise to cover evolving and unknown patterns of attack. To achieve that, IDSs can share knowledge and experience with others to enhance the detection accuracy and achieve mutual benefits. Recent works by (Abusitta, Bellaiche, Dagenais, & Halabi, 2019) (Fung & Zhu, 2016) show that cooperation among IDSs can enhance the detection accuracy by up to 60%.

While several methods were proposed to model the cooperation among cloud-based IDSs (Lo, Huang, & Ku, 2010), these approaches do not consider sharing the privacy of the data. Therefore, Song et al. (Song, Shi, Fischer, & Shankar, 2012) are researching personal data (i.e., data from clouds clients or users) protection measures for in-flight data. In recent times, it has become significantly challenging to avoid data leaks in cooperative cloud-based IDS, given the large amount of data shared among the IDSs to enhance detection accuracy and achieve mutual benefits (Hudis, Helman, Malka, & Barash, 2013). Thus, it reduces the motivation of cloud-based IDSs to participate in IDS communities.

The problems have led to the following research questions that we have attempted to address throughout the thesis:

- *How to preserve privacy in cooperative cloud-based IDS* while simultaneously enhancing or maintaining their detection accuracy?
- *How to evaluate the anonymity of data and decide* which data points should be shared?
- *How to find an efficient mechanism* that allows an IDS to convert the single private data points to a new form similar to the original one semantically but not synthetically, to use the newly generated data to feed other IDSs in the community and improve detection accuracy?

1.2 Research Objectives

This thesis aims to maintain and enhance the detection accuracy of cooperative cloud-based IDS, simultaneously attempting to protect the privacy of cloud-based IDSs' users and clients. More specifically, the objectives of the thesis are:

- To propose a framework that allows us to integrate privacy-preserving techniques into cooperative cloud-based IDSs.
- To devise a decision model that lets an IDS decide which data to be shared while satisfying the goals of enhancing the detection accuracy and protecting private data.
- To design g a mechanism that enables an IDS to create new data similar to the original data semantically but not synthetically to allows the IDSs to handle the generated data instead of the original data. The challenge is to maintain accuracy with the new synthetic data.

1.3 Main Contributions and Originality

The originality of our work lies in the design of privacy-preserving cooperative cloud-based IDS. The principal contributions are:

- Proposing a unified framework for integrating privacy-preserving techniques into machine learning-based IDS. The proposed framework enables us to maintain or enhance detection accuracy.
- Designing a novel algorithm that allows an IDS to evaluate or investigate the efficiency of applying privacy-preserving techniques into machine learning-based IDS. The proposed algorithm allows us to determine the data points to share and increase the detection accuracy while preserving the privacy of cloud-based IDS users and clients.
- Designing and implementing a new method that allows us to convert the IDS original data into new synthetic data. The generated data is used to feed machine-learning models and achieve a high detection accuracy for each IDS in the coalition.

1.4 Thesis Structure

The rest of the thesis is organized as follows. In Chapter 2, we discuss the related work. In Chapter 3, we present the proposed privacy-preserving cooperative cloud-based IDS. In Chapter 4, we report our empirical results to show the effectiveness of the proposed framework. Chapter 5 presents a discussion of our work. Finally, Chapter 6 concludes the thesis by summarizing our contributions and discussing some research gaps that require further investigation.

Chapter 2 – Related Work

In this chapter, we discuss recent works in the areas related to this thesis. We first present some basic definitions of the topic and discuss the related work.

2.1 Cloud Computing

Cloud computing allows cloud providers (CPs) such as Amazon Web Services, Azure, and Google Cloud to rent out space on their infrastructures, platforms, and services to consumers. Thanks to the virtualization, it is possible for the smooth migration of applications and services from node to node. Many companies, organizations, and governments have plans or already planned to transfer[2], if they have not already, all or parts of their IT solutions to the cloud. This transfer is beneficial from an economic viewpoint (Low et al., 2011) as it allows them to streamline operating workflows and reduce expenditure on infrastructure.

2.2 Intrusion Detection System

Intrusion detection is the process of monitoring performance metrics to explore attack symptoms. Some examples of IDS systems are Snort by Cisco and an open-source Zeek[3] for traffic logging and analysis**.** Such symptoms might be at an early stage or advanced enough to affect the efficiency of the system. The IDS are of two main types: Signature-based and Anomaly-based IDS. The signature-based IDS contrasts the behavior of the attack with known types and patterns. In order to ensure that this style of the IDS system is successful, the list of known signatures must to be regularly updated. To fix the vulnerabilities faced by the former, we use the anomaly-based IDS to raise alarms when irregular network traffic behavior is observed. Anomaly-based IDS also

[2] https://www2.deloitte.com/us/en/insights/industry/technology/why-organizations-are-moving-to-the-cloud.html

[3] https://zeek.org/

has the weakness of mistaking normal activity for an attack and an attack for normal activity (Spathoulas & Katsikas, 2010).

2.2.1 Cooperative Intrusion Detection System

It has become challenging for a sole IDS to detect all attacks due to limited knowledge. Collaboration among IDS has proven its efficiency in terms of the accuracy in detecting new and sophisticated attacks. See some examples of cooperative IDS by (Fung & Zhu, 2016) and Revmatch (Fung, Lam, & Boutaba, 2014).

Through collaboration, IDSs in different regions, and possibly belonging to different Cloud Providers (CPs), can cooperate in a way that they use each other's expertise to cover unknown threat patterns. By enabling IDSs to consult each other about suspicious behavior, the received feedback can help decide whether to raise the alarm.

2.2.2 Cloud-based Cooperative Intrusion Detection System

We classify attack detection approaches into three significant categories: signature-based, anomaly-based, and hybrid detection approaches. Hybrid detection efficiently combines both signature-based and anomaly-based approaches to improve detection accuracy.

Figure 1 shows an example of a cloud-based cooperative IDS. If multiple clouds, represented by the small clouds with the red circles signifying the IDS, collaborate with each other through the cloud federation's exchange of metrics and data, they can improve their performance and efficiency in detecting different types of attacks.

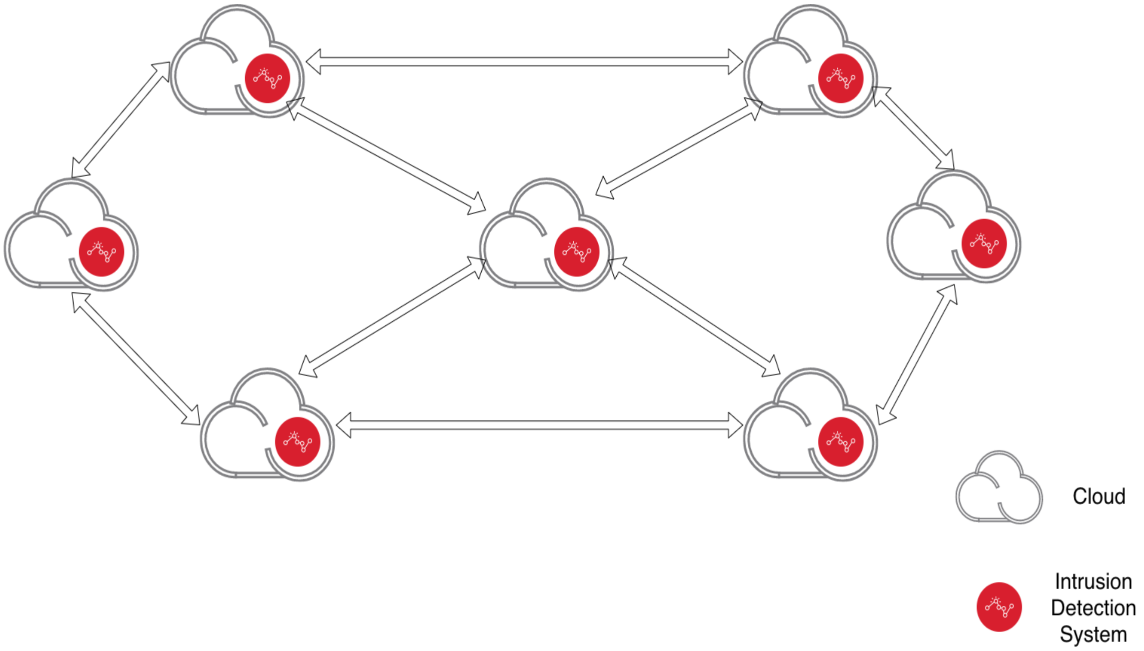


Figure 1. Cooperative Cloud Based IDS

2.2.3 Signature Based Detection System

A signature-based IDS is designed with known rules and filters for an IDS to detect and raise the alarm. An implementation by (Lonea, Popescu, & Tianfield, 2013) proposes SNORT, a signature-based detection method configured with already known rules to detect known attacks in the cloud environment. They tested their method by simulating the flooding attacks across different protocols. Another approach (Bakshi & Dujodwala, 2010) deploys SNORT on every virtual machine (VM) at its virtual interface, allowing VMs to analyze traffic in real-time. Snort-based detection methods utilize the advantage of sniffing attacker packets on known patterns and raising the alarm on the unknown packets. They are also useful in detecting Networked Denial of Service (NDoS) attacks.

Moreover, other IDSs designed by (Gupta & Kumar, 2013) use a similar attack detection approach based on VMs profile optimization. Their approach involves deploying a rule-based detection method for matching packets in flooding attacks by generating a threshold for each rule pattern.

Although signature-based detection approaches are extremely effective in detecting known attack patterns, the weakness of these approaches is apparent when the attack patterns are dynamic and ever-changing. Attackers are becoming smarter in launching attacks in such a way that they try to avoid known patterns of attack and bypass the IDS rule(s) from detecting their attack. Instead, more complex and better methods must be adopted by observing and adapting to the cloud environment they are deployed in. Fortunately, the majority of the new tools used to launch DoS attacks are available on the internet and are proactively understood (Bhuyan, Kashyap, Bhattacharyya, & Kalita, 2014) to design newer rules for a static rule set based IDS.

2.2.4 Anomaly-Based Detection System

The approach of an anomaly-based is different for detecting attacks than the signature-based approach in terms of its ability to detect unknown attack patterns. We classify the state-of-the-art anomaly-based detection techniques into two main categories: Machine Learning (ML) and Statistical approaches.

Several methods have been proposed by (Garcia-Teodoro, Diaz-Verdejo, Maciá-Fernández, & Vázquez, 2009) to use machine learning models as Bayesian Networks, Markov Models and Neural Networks to improve the anomalous detection with high detection rate results.

Also, (Garcia-Teodoro et al., 2009) have used the statistical model and knowledge-based models for designing the anomaly-based detection system. Besides, (Su, 2011) has designed a model by using the KNN Based Classifiers for online anomaly network classification.

Earlier methods used data mining techniques in intrusion detection, and a general study of their implementation by (C. C. Aggarwal & Philip, 2008) and (Barbara, Couto, Jajodia, Popyack, & Wu, 2001) show the various ways it helped in the detection of attacks by allowing the IDS to learn by itself.

The approaches elaborated so far do not consider the privacy aspect of the data traveling on the network in a cooperative cloud or between multiple standalone IDS. Several privacy-preserving intrusion detection systems address this issue. In Section 2.2.5, we explore them in further detail.

2.2.5 Privacy Preserving Intrusion Detection System

As we have observed, the anomaly detection method is better than the static signature-based approach. Most of the IDSs are built with anomaly detection features. The inherent weakness in this approach is that with the lack of datasets, the system cannot correctly classify the attacks.

In a cooperative cloud environment, multiple IDSs are strategically placed. The protection of the privacy of the data shared is a growing concern. Several methods of intrusion detection were designed and developed to address this issue while maintaining data privacy**.** (Shokri & Shmatikov, 2015) have designed a deep learning technique that enables multiple parties to learn an accurate neural-network model for an objective without sharing their input datasets. The advantage of the optimization algorithms used in modern deep learning, namely, those based on stochastic gradient descent, is that it can be parallelized and executed asynchronously. The system they have invented allows the participants to independently train their datasets and selectively share small subsets of their model's key parameters during training. This offers an attractive point in the trade-off space of utility/privacy. Participants maintain the privacy of their data while benefiting from the models of other participants and thus boosting their learning accuracy beyond what they achieve through their sole inputs**.**[4]

(T. Zhang & Zhu, 2018) have designed a privacy-preserving framework for Vehicular Ad-Hoc networks (VANets) enabling technology in modern transportation systems in helping to provide safety and valuable information and yet vulnerable to attacks from passive eavesdropping to active interference.

To this end, distributed machine learning is an appropriate framework for the design of scalable and implementable collaborative detection algorithms over VANets. One fundamental barrier to collaborative learning is the privacy concern as nodes exchange data among them. A malicious

[4] https://www.cs.cornell.edu/~shmat/shmat_ccs15.pdf

node can obtain sensitive information from other nodes by inferring from the observed data the algorithm employs the alternating direction method of multipliers to a class of empirical risk minimization problems and trains a classifier to detect the intrusions in the VANets. The authors use the concept of differential privacy to capture the privacy notation of the privacy-preserving machine-learning-based collaborative IDS (PML-CIDS) and propose a method of dual-variable perturbation to provide dynamic differential privacy.

(Niksefat, Sadeghiyan, Mohassel, & Sadeghian, 2014) introduce a client- server solution mechanism in detecting attacks based on signatures, in which the server is frequently updated with the sensitive signatures of zero-day attacks. For clients connecting to the IDS, the server compares both signatures and thus can classify it as an attack or not. In this approach, neither the server nor the client is aware of each other's data, which gives a guarantee in preserving the privacy of the shared data.

To implement this, the authors reduce privacy-preserving intrusion detection to an instance of secure two-party oblivious deterministic finite automata (ODFA) evaluation (Hromkovič & Schnitger, 2003). Then, motivated by the fact that the DFAs associated with attack signature is often sparse, the authors proposed a new and efficient ODFA protocol that takes advantage of this sparsity.

(Niksefat et al., 2014) and (Meng, Li, Kwok, & Xiang, 2013) also suggested a signature comparison for having Intrusion Detection Systems as a Service (IDSaaS). (Dara & Muralidhara, 2016) have proposed a cryptographic approach to the real-world use cases to defend against intrusion detection attempts from the availability of global data, which is a must for organizations to defend against modern advanced persistent threats (APTs). Although several implementations of use case-specific versions in privacy-preserving IDS have been examined, privacy concerns continue to be of major hindrance of cooperative intrusion detection.

As with anomaly-based intrusion detection systems, which use machine learning-based training to improve and detect outliers, we briefly introduce machine learning in intrusion detection systems in Section 2.3.

2.3 Machine Learning in Intrusion Detection Systems

Machine Learning (Michie, 1994) has been effectively utilized in calculating the accuracy of attack detection and determination. The use of machine learning has helped decide which sets of data and parameters are tantamount to an attack or false flag. The results are based on the accuracy, precision, recall, and f-score of the machine learning algorithms. Classification is a data mining method of assigning data instances into one among the few categories. Many classification algorithms have been developed to outperform one another, and they work based on mathematical techniques like decision tree, linear programming, and neural networks. These techniques analyze the available data to make predictions of an attack. Other more advanced algorithms in machine learning have also been utilized in various contexts such as deep learning, neural network, and Bayesian classifiers.

As part of our experimental study, we have considered four machine-learning algorithms, Random Forest (Liaw & Wiener, 2002), K-Nearest Neighbors (Peterson, 2009), Support Vector Machines (Suykens & Vandewalle, 1999) and Decision Trees (Safavian & Landgrebe, 1991).

Several implementations that enhance the wide usage of machine learning techniques exist, but we focus on the use case of privacy preservation, where we discuss them in Section 2.3.1.

2.3.1 Privacy Preserving Machine Learning

(Sweeney, 2002b) explain the situations where aggregate statistical information was once the reporting norm now rely heavily on the transfer of detailed information. It happens at a time when more and more historically, public information is also electronically available. When these data are linked together, they provide an electronic shadow of a person or organization that is as identifying and personal as fingerprints, even when the sources of information contains no explicit identifiers. Owners of data remove or encrypt explicit identifiers such as names, addresses, and phone numbers to protect individuals' anonymity to whom released data refer to. However, other distinctive data, which we refer to as quasi-identifiers when combined, are linked to publicly available information to re-identify individuals.

(Sweeney, 2002b) have proposed the k-anonymity approach to preserve privacy, by elaborating how they addressed the problem of releasing personal data while safeguarding anonymity**.** The approach pursued for multidimensional attributes is based on the extension of k-anonymity to multidimensional attributes, namely, Mondrian k-anonymity. A dataset is said to satisfy the k-anonymity principle for k > 1 if, for each combination of fundamental attribute values, at least k records exist in the dataset sharing that combination (Domingo-Ferrer, 2018).

In the information age, our internet searches, history of purchases, videos watched, and movie preferences are a few types of information that are being collected and stored daily. This data collection happens within every electronic gadget. Such private data is used in various applications like machine learning and statistics.

To address the privacy aspect of the collected data, (Al-Rubaie & Chang, 2019) have proposed several methods of privacy preservation to collect data on a contextual basis. The methods they have proposed vary from homomorphic encryption to secret sharing approach, where the secret is broken down into multiple parts with each entity holding a "share" of the secret**.** Individual shares are of no use on their own; however, when the shares are recombined, the secret can be reconstructed. With threshold secret sharing, not all the "shares" are required to reconstruct the secret, but only a threshold "t" of them is required to reconstruct the secret. (Al-Rubaie & Chang, 2019) have also explored differential privacy by (Friedman & Schuster, 2010) for gently perturbing the data and the different ways of achieving differentially private data for input data. They show that it is also applicable algorithmically and perturbing output data by additive noise.

Deep Learning methods have also been proposed by (Hesamifard, Takabi, & Ghasemi, 2017) by using Neural Networks as a way to allow for anomalous detection of the data while maintaining privacy. On the other hand, with the increasing growth of cloud services, several Machine Learning as a Service (MLaaS) (Ribeiro, Grolinger, & Capretz, 2015) are offered where training and deploying machine learning models are performed on cloud provider's infrastructure. They address the issue of privacy preservation for the data by developing a new technique of applying neural network algorithms to the encrypted data. They also show that the result they achieve is also encrypted, thus proving the applicability of encryption to privacy-preserving machine learning. The empirical results they have achieved show that it provides an accurate training and classification while preserving the privacy of the data. (Mendes & Vilela, 2017) have explored several approaches such as noise addition, data perturbation, generalization, and homomorphic encryption depending on the collected data or published data**.**

Major companies namely, Apple[5], Google, and IBM, have understood the growing need for privacy in the data they collect and have responded with their versions of privacy-preserving machine learning methods. Apple has positioned itself to stand up for user privacy rights and has taken a significant number of steps towards achieving that. Apple has gone further to improve the privacy of the data it collects by establishing an on-device computing version of differential privacy before the data is collected for analysis. Google has released a Tensorflow based version of its privacy-preserving machine learning algorithms such as RAPPOR (Erlingsson, Pihur, & Korolova, 2014). Similarly, IBM has released its set of libraries for Privacy-Preserving Machine Learning. Microsoft has also proposed SecureNN[6] to promote privacy-preserving machine learning by using neural networks.

Further research by (Lin, Shi, & Xue, 2018) has shown a practical approach for enhancing intrusion detection, which is an implementation of the generative adversarial network (GANN) model, IDSGAN, to generate the adversarial attacks, which can deceive and evade the intrusion detection system**.** Considering that the detection system's internal structure is unknown to attackers, adversarial attack examples perform the black-box attacks against the detection system. They utilize the fact that an IDSGAN leverages a generator to transform original traffic into trained traffic they can learn on. They have used a discriminator that classifies traffic examples and simulates the black-box detection system. Their tests on the NSL-KDD dataset have

[5] https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf

[6] https://www.microsoft.com/en-us/research/uploads/prod/2018/09/securenneprint.pdf

demonstrated excellent results in detecting various attacks with generated data similar to the original.

2.4 Summary

Generally, for a multi-cloud environment, the preservation of privacy in a cooperative IDS was yet to be addressed. Thus, in this thesis, we present a new framework that allows us to integrate privacy-preserving techniques into anomaly-based IDSs to obtain privacy-preserving cooperative IDSs. The proposed framework allows us to hide private and sensitive information in shared data while improving or maintaining the IDS detection accuracy. We choose the anomaly-based IDS since we can implement the privacy-preserving machine learning algorithms over the synthesized data and train the overall system to detect unknown attacks easily.

Chapter 3 – Privacy Preserving Framework in a Cooperative Cloud-based IDS

In this section, we present the proposed framework for achieving privacy-preserving cloud-based cooperative IDS.

3.1 Problem Definition & Challenges

Some detection methods, such as machine learning methods, based on anomalous behavior, are employed to decide whether a suspicious activity is an attack or not to improve the accuracy of IDS. However, in a cooperative setting (i.e., cooperative cloud-based IDS), data needs to be shared among IDSs to achieve a greater detection accuracy. When the data is shared among these nodes, the aspect of information privacy is not considered. Therefore, personal data breaches occur. It has become increasingly challenging to avoid data leaks in cooperative cloud-based IDS, given the vast amount of data shared among IDSs to enhance detection accuracy.

It’s becoming a challenge to protect privacy in data-sharing IDSs since we need to preserve the privacy on these systems while enhancing or maintaining their detection accuracy. In fact, most privacy-preserving algorithms are based on achieving a trade-off between privacy and accuracy (T. Li & Li, 2009; Rastogi, Suciu, & Hong, 2007; Sankar, Rajagopalan, & Poor, 2010; H. Zhang, Shu, Cheng, & Chen, 2016).

Thus, the challenge is to find an efficient mechanism that allows for data conversion into new data similar to the original one semantically but not synthetically. Then, an IDS inside the IDS community should use the generated data rather than the original data to decide whether suspicious activity is an attack or not. The mechanism would effectively allow us to achieve high accuracy without the need to share the original data points.

3.2 The Proposed Framework

To address the problems mentioned above, we propose a unified framework for integrating privacy-preserving methods into machine learning models that allows IDSs in the community to hide the private information of their users or clients while maintaining the detection accuracy. We can do that by converting the original data points to new data points similar to the original ones semantically but not synthetically. As a result, we achieve high accuracy without the need to use original data. To this end, we propose a decision-based mechanism that allows us to investigate the extent to which the generated data using privacy-preserving techniques is useful for achieving higher detection accuracy. This allows an IDS to decide whether to publish the newly generated synthetic data or not. Figure 2 shows the flowchart steps of the proposed framework.

For simplicity, we have decided to apply the following machine learning algorithms in the experiment: Random Forest (RF), K-Nearest Neighbours (KNN), Decision Trees (DT), and Support Vector Machines (SVM). Each of these machine learning algorithms can be used to train our datasets after applying privacy-preserving techniques. We have decided to use these algorithms because they are widely accepted in the literature and proved to achieve higher detection accuracy than other machine learning methods (Tsai, Hsu, Lin, & Lin, 2009).

As shown in the proposed flow chart diagram (Figure 2), we first run machine learning algorithms on the original data D_0. In particular, we used each of our selected machine learning algorithms and calculated the average accuracy as A_0. We repeated these steps for the data D_1, which results from the modifications applied on D_0 using differential privacy. We then compare the accuracy of A_0 and A_1. If the accuracy is more or less similar to each other, the algorithm will progress to the next step; otherwise, it will stop. The decision to stop this algorithm indicates that the disclosure of D_1 privacy-preserved data is not feasible due to the deterioration of the detection accuracy.

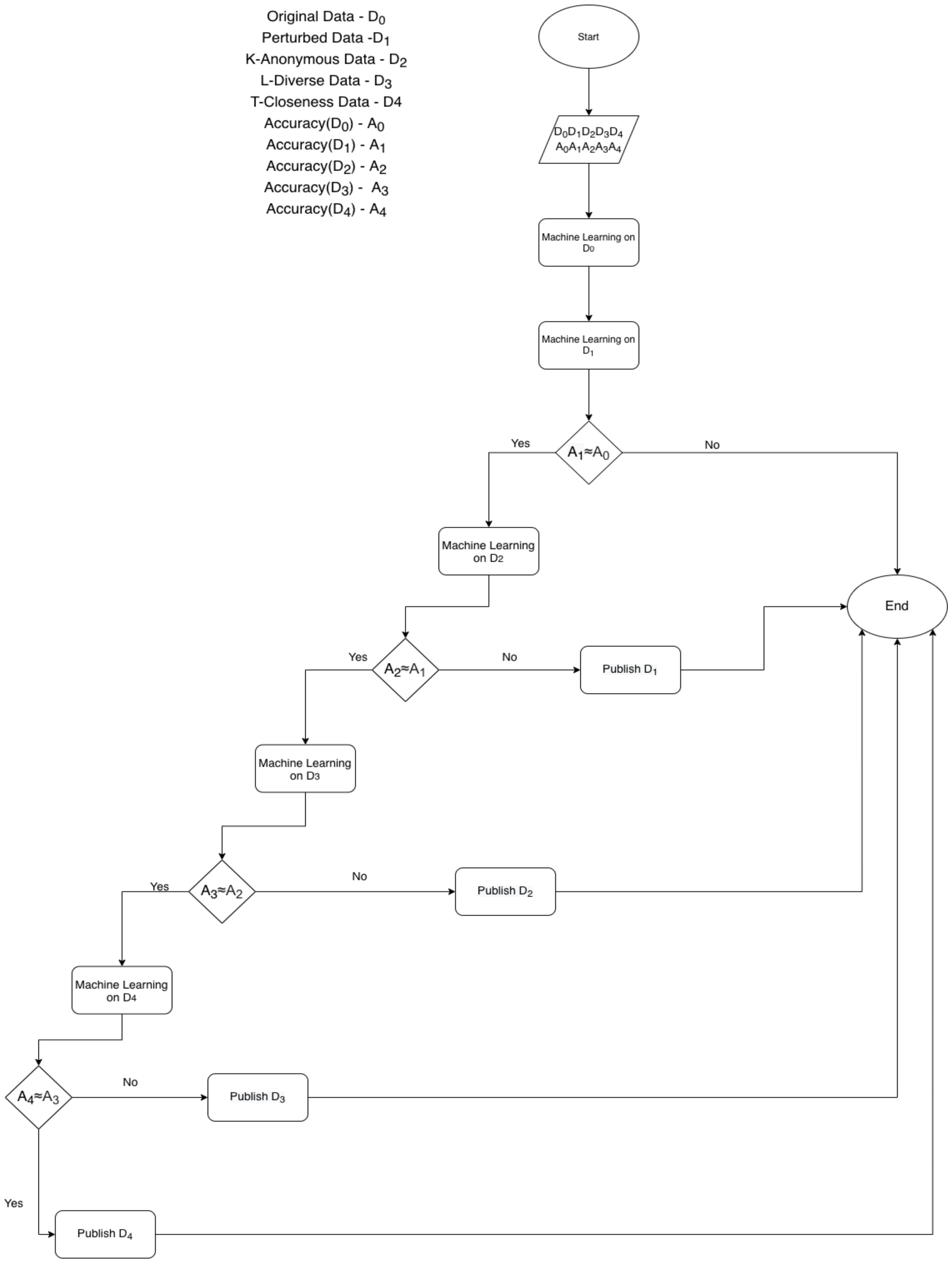


Figure 2. Flow Chart of Our Proposed Framework

When the algorithm progresses to the next phase, the average accuracy of machine learning algorithms to a new privacy-preserving technique A_2 is compared with A_1 to ensure that the accuracy will not be degraded using the new privacy-preserving technique. If the accuracies are indeed similar, then the algorithm pushes forward to the next phase. If the accuracies are not similar, the algorithm publishes the data D_1. The published data may be used by other IDSs in our cooperative system to apply a machine learning algorithm(s). However, we are still in a position to apply other privacy-another preservation technique(s) in the next phase, hoping to guarantee more data privacy. It is worth mentioning that the privacy-preserving techniques used in the proposed framework are designed to ensure that the next privacy-preserving technique provides more protection than the previous one. The same can be applied to the next phases until we reach the most protected data D_4 for the machine learning and privacy preservation techniques used.

This algorithm guarantees the anonymity of the published data over the different stages and the efficiency of its usage to train machine-learning models and achieve acceptable accuracy in collaborative IDS. Note that at each point, we argue that the framework is allowed to publish the protected data generated in the previous stage if the accuracy obtained is acceptable. However, the anonymity of the data is optimally protected after passing through all the four stages of our framework.

The dataset used to verify and evaluate the proposed flowchart is the NSL-KDD Dataset (Dhanabal & Shantharajah, 2015), widely used in the cybersecurity community. Other datasets can also be applied. However, different machine learning algorithms might be used to enhance the detection accuracy.

3.3 Selecting and Sorting Privacy Preserving Techniques

In Section 3.2, we proposed a decision-based mechanism that allows an IDS to decide whether to publish data or not. The suggested framework decides to publish the most protected data if the calculated detection accuracy for each privacy technique and machine learning model used is acceptable or high. The shortcoming of the proposed framework (Figure 2) is that the privacy-preserving listed techniques should be sorted in such a way that any shortcomings in the current phase can be addressed in the next phase. In other words, the proposed framework should work in such a way that the current phase (the current privacy-preserving technique) should cover the weaknesses and problems faced by the previous phases. To address this issue, we investigate the use of privacy-preservation techniques capable of achieving this property.

To this end, we have selected a set of approaches, namely Generalization (Bayardo & Agrawal, 2005), Suppression (Sweeney, 2002a), and Perturbation (Kargupta, Datta, Wang, & Sivakumar, 2003). In the next subsection, we show how some techniques extracted from these approaches can complement each other and achieve higher privacy protection levels.

Below, we briefly introduce the privacy-preserving approaches. Then, we describe the techniques selected from these approaches to achieve complementary properties as described above. In the next section, we review some techniques such as k-anonymity, l-diversity, t-closeness.

1. **Generalization**: This method of anonymization involves the replacement of a value by a more general one. Numerical data may be specified by intervals (e.g., an age of 53 may be specified as an interval in the form of [50, 55]), whereas categorical attributes require the definition of a hierarchy. An excellent example of a hierarchy is to generalize engineers and artists' values from occupation and attribute them to a professional. Another possibility would be to have students' parent values represent all types of students in the same occupation attribute.
2. **Suppression**: This method of data anonymization involves the extraction of some attribute values to prevent information disclosure. This operation can also be performed column-wise, in a data set (removes all values of an attribute), or row-wise (removes an entry).
3. **Perturbation**: In this data anonymization method, the original data is replaced by synthetic values with the same statistical information. The randomization methods are mostly additive and multiplicative noise. Data swapping and synthetic data generation are also perturbation techniques. Differential privacy is one commonly used method of data perturbation.

4. **Data Swapping**: The swapping of data has been proposed by (Dalenius & Reiss, 1982). In this method, sensitive attributes are exchanged between different entries of the dataset to prevent the linkage of records to identities.

From the four methods mentioned above, we combine the methods of perturbation and generalization in such a way that one method can improve the weaknesses perceived by the other. We have chosen perturbation and generalization specifically since they prevent the re-identification of the users in the data in the best manner while maintaining the trade-off between privacy and utility.

Techniques such as suppression and data swapping can also be used depending on the datasets. We propose using machine learning to determine accuracy, suppressing the data is not constructive, and data swapping can be easily undone any adversary who has background knowledge of the dataset.

In Figure 3, we show the selected privacy-preserving methods. As represented in Figure 3, the initial perturbation of data by differential privacy improves the weakness of prior knowledge attacks faced by k-anonymity. In this method of synthesizing information, a mathematical model is constructed with the original data and is applied discreetly so that the attacker cannot predict that the synthetic data was added to the normal data set. This is simpler than the data swap procedure, where the data is swapped randomly between columns, and the attacker can quickly revert the changes made.

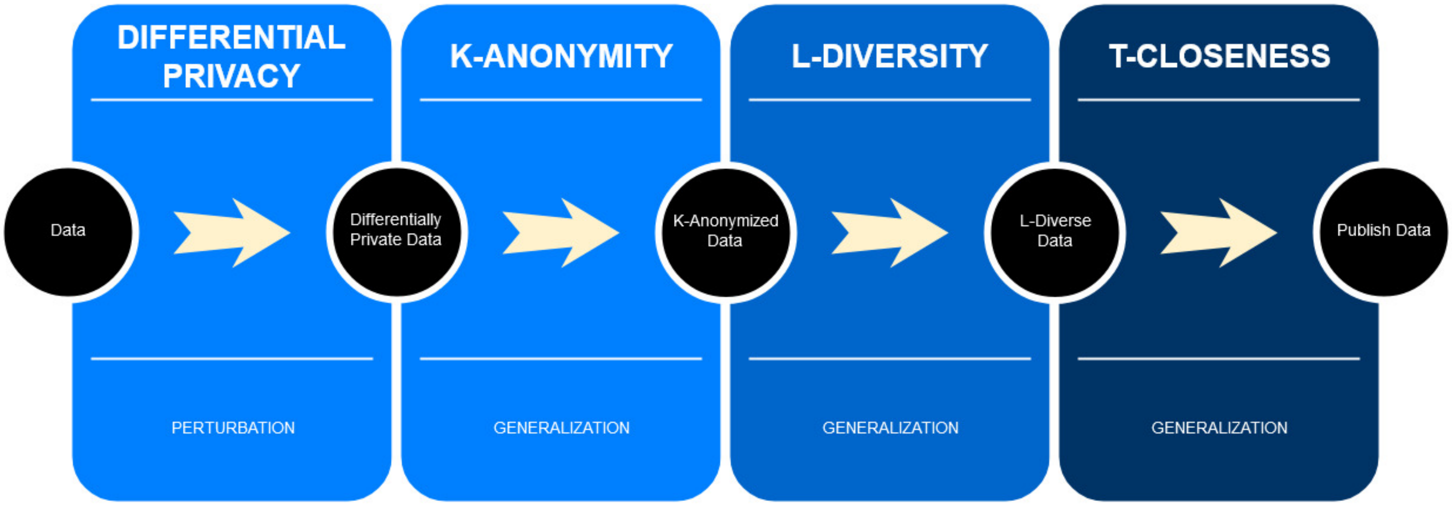


Figure 3. Our Proposed Framework

The technique of using Mondrian k-anonymity for multidimensional data is insufficient. An adversary who knows that a person is in the k-anonymous group can still learn the sensitive attribute of a person with absolute certainty. An extension of k-anonymity solves this issue, l-diversity ensures that each k-anonymous group contains at least l different values for the sensitive attribute. Finally, the weakness of l-diversity has overcome by t-closeness. The weakness of l-diversity arises as it generates partitions that contain a massive number of entries for one sensitive attribute and only a few entries for the other sensitive attributes. It is fixed by extending the l-diverse data to t-closeness, which specifies that the statistical distribution of the sensitive attribute in each l-diverse group is similar to the overall distribution. Section 3.3.2, 3.3.3, and 3.3.4 go into greater detail.

Figure 3 lists our proposed data anonymization strategy that can be applied to other datasets beyond those related to network intrusion, including telemetry and analytical data. In the next section, we briefly introduce the privacy-preserving techniques. Moreover, we explain how each technique overcomes the shortcoming of the previous technique. The perturbed data is an improvement over using the actual data as it covers the weakness of prior knowledge attack on the next approach in the framework, namely k-anonymity.

3.3.1 Differential Privacy

An algorithm is said to be differentially private if an observer is unable to tell whether an individual's information was used in the computation by seeing its output. Differential privacy is often used in situations where sensitive and recognizable personal data is stored in a database. Differentially, secret algorithms do not directly refer to attacks for identification and re-identification because they are resistant to such attacks. Examples of implementing the differential privacy for real-time data are by Apple[7] and Google[8] collecting data from their users in a manner where personally identifiable data is not easily done.

The research done by (Dwork, 2011) introduces the concept of ε-differential privacy, a quantitative description of privacy loss associated with any release of information from a statistical database. In describing the statistical database concept, we may expand the fact that a collection of information is obtained in compliance with the guarantee of confidentiality for producing statistics. The rationale for the concept of ε-differential privacy is that the privacy of an individual is not violated by a statistical release, even if their information is in the database.

The purpose of using differential privacy is mainly to give each person the same privacy as that which would result from the deletion of their information. In other words, the statistical functions performed on the data should not be overly dependent on the data of any entity and, therefore, on our case of multi-attribute data**.** We test the definition of ε-differential privacy to a combination of several attributes we have defined as quasi-identifiers (QIs).

3.3.1.1 Implementing ε-differential privacy

We apply the concept of additive noise; one such example is the Laplacian mechanism allowing us to add controlled noise into the data to prevent identification or re-identification of published data. Additive Noise Mechanism works in the way that for any sample data in a table named X and the noise generated through a chosen noise function is Y.

[7] https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/

[8] https://www.macobserver.com/analysis/google-apple-differential-privacy/

The mechanism of additive noise is explained in equation 1, where X is the data, and Y is the controlled and random noise generated via the Laplacian Mechanism.

$$X = X + Y \qquad \text{---(1)}$$

Equation 2 describes the working of Laplacian noise generation. Let D be a collection of data points and $f: D \rightarrow R$ be a real-valued function, the sensitivity $\Delta f = max|f(x) - f(y)|$, where the maximum is over all pairs of datasets x and y in D, where μ is the expected value of the Laplace distribution and b is the scale parameter.

$$\mathrm{M}_{\mathrm{Lap}}(x, f, \epsilon) = f(x) + Lap(\mu = 0, b = \frac{\Delta f}{\epsilon}) \qquad \text{---(2)}$$

Laplacian noise is applied to the data with a low expected value and scale so that the resulting skewed data is intentionally statistically close to the original values in order that as little information as possible can be leaked. Heavy noise generation can drastically affect the utility of the data for which the intrusion detection system needs to learn and fine-tune itself**.** Following the initial phase of data disruption, we explain in Section 3.3.2 how k-anonymization benefits from the perturbed data.

3.3.2 K-Anonymity

K-anonymity is a lexical transformation on attributes. It is a mapping from the universe of all possible data to those who respect the k-anonymity principle (Domingo-Ferrer, 2018). The values of k can significantly affect the anonymity achieved. There exists the possibility of leakage of private data when poor values of k are chosen or when the distance between the original data and the mapped data is small. On the other hand, the data is mostly useless when the distance between the original and mapped values is large. One of the best use cases for k-anonymity is password leak validation; one such implementation is by Cloudflare[9].

The argument against k-anonymity is that it does not work well with prior knowledge. The possibility of information exposure is significant if attackers have outside knowledge of the data

[9] https://blog.cloudflare.com/validating-leaked-passwords-with-k-anonymity/

set. This attack is further explained as if the attacker has a background knowledge of the data and can leverage the association between one or more quasi-identifier (X. Zhang, Liu, Nepal, & Chen, 2013) attributes with the sensitive attribute to reduce the set of possible values for de-identification.

We choose differentially private data as input for the next step in our framework to cover the weakness of the pre-existing knowledge attack in k-anonymity. Our previous method of using differential privacy is intended to prevent the background knowledge attack from happening. Since the responses are statistically close, no matter what prior knowledge the attacker has on the dataset, not much information is exposed in the prior knowledge attack (Smith, 2012).

The k-anonymity algorithm exists in various forms, namely, alpha k-anonymity (Wong, Li, Fu, & Wang, 2006) and incognito full-domain k-anonymity (LeFevre, DeWitt, & Ramakrishnan, 2005). It indicates that multiple approaches for different use cases exist. We utilize the Mondrian multidimensional k-anonymity approach as IDS traffic contains multiple attributes. K-anonymity is of two types: generalization and suppression. We used the generalization hierarchy to achieve k-anonymized data.

K-anonymity protects the privacy of the entity by pooling attributes into groups of at least k entities. These attributes, known as quasi-identifiers, combined several of them into a single identifier, can often uniquely identify a person even in large datasets. For example, the combination of gender, age, and zip code might be so specific that only a single person in a dataset has a given combination.

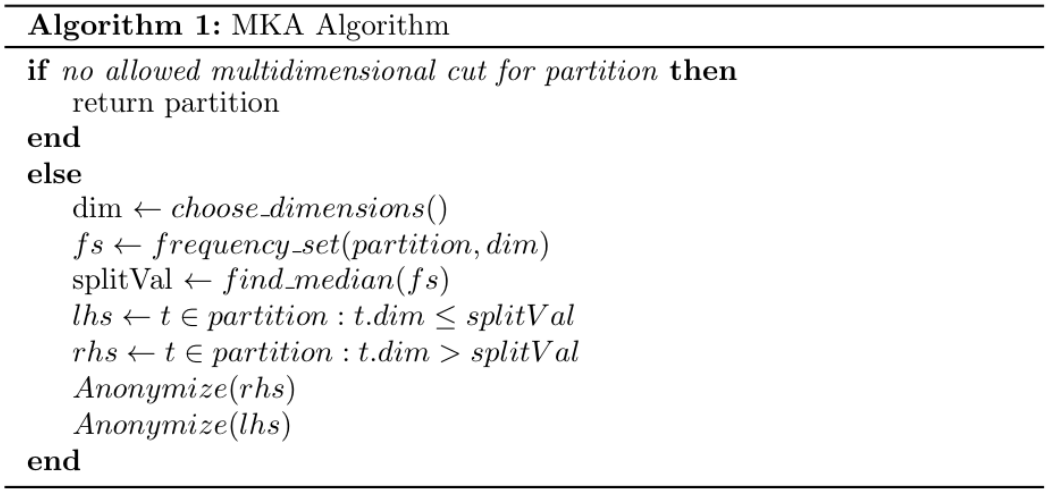
Now, k-anonymity requires that the rows in our dataset be grouped into groups of at least k rows and replace the quasi-identifying attributes of these rows with aggregate values no longer make it possible to read the individual rows. This protects the data anonymity by ensuring that an opponent who knows all the values of the quasi-identifying attributes of a target can only find out which group that target might belong to. However, the adversary might not know if the target is really in the dataset.

Making a dataset k-anonymous is a complex problem, and finding the optimal partition in k-anonymous groups is an NP-hard problem (Meyerson & Williams, 2004). Fortunately, practical algorithms that often deliver optimum results by using greedy search techniques exist.

We specifically utilize the form of k-anonymity used for multidimensional data, the Mondrian k-anonymity algorithm. This approach uses a greedy search algorithm to partition the original data into smaller and smaller groups. The algorithm assumes that we have converted all attributes into numerical or categorical values to measure the span of a given attribute.

3.3.2.1 Mondrian K-Anonymity Generalization Approach

The Mondrian Multidimensional K-Anonymity (LeFevre, DeWitt, & Ramakrishnan, 2006) technique of anonymizing the data is shown in Figure 4.



```
Algorithm 1: MKA Algorithm
if no allowed multidimensional cut for partition then
    return partition
end
else
    dim ← $choose\_dimensions()$
    $fs \leftarrow frequency\_set(partition, dim)$
    splitVal ← $find\_median(fs)$
    $lhs \leftarrow t \in partition : t.dim \leq splitVal$
    $rhs \leftarrow t \in partition : t.dim > splitVal$
    $Anonymize(rhs)$
    $Anonymize(lhs)$
end
```

Figure 4. Mondrian K-Anonymity Algorithm

We will show how the Mondrian k-anonymity algorithm, presented in Figure 4, works. First, we determine the relative spans of all columns in the partition, sort, and iterate the resulting columns by span (in descending order). Second, for each column, we try to split the partition along the column using the median column values as the split point and test if the resulting partitions satisfy the k-anonymity criterion. If the two new partitions are valid, attach them to the working set and break out of the loop. If no column produces a valid partition, we add the original partition to the complete partition set. Third, we aggregate quasi-identifiers' values and the sensitive attributes in each k-anonymous group after obtaining the partitions.

The approach can be extended to datasets with more than one sensitive attribute or datasets, where quasi-identifiers and sensitive attributes are not clearly differentiated. Because our dataset is multidimensional and has several sensitive attributes, we find this approach helpful.

The k-anonymity technique of our framework has its weaknesses. The weaknesses, such as the homogeneity attack on k-anonymous data that do not provide complete anonymity if the equivalence classes (i.e., a set of records that are indistinguishable for certain identifying attributes) lack diversity. We would better address the weakness of sensitive data stored, such as zip code and age data, if the zip code has common digits or age groups fall within a range of 20-30 or 30-40. The data for zip codes would be stored as 123***, and the age data would be stored as 2* or 3* for that homogenous dataset. The other weakness of the k-anonymous dataset is that an attacker may have an idea of how the dataset is organized, also known as background knowledge attack[10].

Whereas the benefits of using k-anonymity for distorting data from perturbation, Section 3.3.3 extends the framework to address the drawbacks of using k-anonymity technique.

3.3.3 L-Diversity

We turn to the concept of l-diversity that addresses the problem faced by the weakness of k-anonymity (Machanavajjhala, Gehrke, Kifer, & Venkitasubramaniam, 2006). An adversary who knows that a person is in the k-anonymous group can still learn the sensitive attribute of a person with absolute certainty. Using l-diversity, an extension of k-anonymity can solve this problem. L-diversity ensures that each k-anonymous group contains at least l different values for the sensitive attribute. Therefore, even if an adversary can identify the group of a person with a high degree of certainty, he/she would not be able to find the sensitive attribute(s).

L-diversity is a group-based anonymization used to preserve privacy in data sets by reducing the granularity of a data representation. The reduction is a trade-off that results in some loss of effectiveness of data management or mining algorithms to gain some privacy.[11]

[10] http://www.cs.cornell.edu/~shmat/courses/cs5436/anonymization.pdf

[11] https://en.wikipedia.org/wiki/L-diversity

The l-diversity model handles the k-anonymity model's weaknesses, where protecting identities to the level of k-individuals is not equivalent to protecting the corresponding sensitive values that were generalized or suppressed. The problem arises when the sensitive values within a group exhibit homogeneity. The l-diversity model adds to the anonymization process by fostering intra-group heterogeneity of sensitive attribute values.

To obtain l-diversity in our data, we check each partition's size while ensuring that the values of the sensitive attribute in the partition are sufficiently diverse by modifying the split functions for diverse splits.

Like with l-diversity, which addresses the weaknesses in k-anonymity, it also has weaknesses and limitations. It is neither necessary nor sufficient to reduce attribute disclosure. Even when using l-diversity, the adversary could still learn some details about an individual's sensitive attribute through probabilistic reasoning. For example, if four out of five individuals in a 5-anonymous group have a specific value of a sensitive attribute, the perpetrator may probabilistically reason that the person he/she knows is part of a group that has that value and narrow down his scope of attacks.

Section 3.3.4 further elaborates on the approach to address the drawbacks of using l-diversity.

3.3.4 T-Closeness

For a dataset with multiple sensitive attributes, l-diversity generates partitions that contain a vast number of entries for one sensitive attribute and only a few entries for the other sensitive attributes. That problem is overcome by extending the l-diverse data to t-closeness, which specifies that the statistical distribution of the sensitive attribute in each l-diverse group is similar to its overall distribution.

(N. Li, Li, & Venkatasubramanian, 2007) define an equivalence class (i.e., a set of records that are indistinguishable from each other with respect to certain identifying attributes) to be t-closeness if the Kullback-Leibler distance between the distribution of the sensitive attribute in this class and the distribution of the attribute in the whole table is not more than a threshold t. A table satisfies t-closeness property if all equivalence classes have t-closeness or satisfy the threshold t. T-closeness demands that the statistical distribution of the sensitive attribute values in each l-diverse group is close to the distribution of that attribute in the entire dataset. The reduction is a trade-off that results in some loss of effectiveness of data utility to gain some privacy.

To achieve the principle of t-closeness over our data, we test a function that returns True if the partition is diverse enough and False otherwise. To measure diversity, we calculate the Kolmogorov-Smirnov (Massey Jr, 1951) distance between the empirical probability distribution of the sensitive attribute over the entire dataset vs. the distribution over the partition.

Since, k-anonymity, l-diversity, and t-closeness limit the information that a legitimate user can also learn from the data we need to find a balance between preservation of privacy against the utility of the resulting data.

We conclude the framework that we have sought to anonymize the data for each level of the framework complements the other. Section 3.4 examines the issues that arise from anonymizing cloud/network data.

3.4 Challenge of Selecting Sensitive Attributes

In Section 3.3, we explained the framework that provides anonymity in datasets. The described framework applies generically to any data sets taken into consideration to be specific to our test network dataset, namely NSL-KDD, which consists of data in granular form or known as “microdata”.

The challenge of publishing anonymized network data, which is a form of microdata (Coull, Monrose, Reiter, & Bailey, 2009), has been the focus of study for many research groups. It continues to be an area of interest, leading to the rise of several methods that assist with microdata anonymization.

In our research, we recognize the different aspects of sharing network data in. The problem of sharing network data is the inherently complex nature of each member of the cloud network makes it virtually impossible to enforce a single standardized privacy policy across the network. Besides, it is difficult to identify the appropriate types of microdata that need to be anonymized.

Microdata anonymization works in such a way that it is quantifiable and difficult to infer potentially sensitive details about data entities such as protocol_type, service, flag, num_failed_logins, etc. If we share data between the different cloud users, we would like it to be difficult for the members of each cloud-based entity to recognize the data of the members in another cloud. Key attributes of the microdata are generally used to make inferences on the row's identity from external sources of data. They do not explicitly identify a row, but the unique attribute values may be used to connect rows in the anonymized microdata with other databases that have the identifying information. For example, if a row in the microdata had a unique combination of key attributes, then an adversary could use these attributes to look up the row's identity in an auxiliary database that has the data he can correlate. Finally, sensitive attributes are classified in that manner as they aren't found in any other databases for the adversary to link to specific identities (Coull et al., 2009). To accomplish the objective of anonymization, the data publisher eliminates attributes and uses one or more anonymization methods to modify the relationship between key attributes and sensitive attributes to ensure that such inferences are unlikely to occur. The resulting sanitized microdata is evaluated to quantify its level of privacy and utility.

For our use case, the cloud administrator is interested in protecting the confidentiality of several entities; clients, security protocols, and hosts running on the cloud. The fact that cloud information is multifaceted and of different types makes it difficult to define an entity's sensitivity. However, a single data record may have an impact on the privacy of different cloud entities.

Additionally, the confidentiality of these individuals does not depend on a single row of data, but on several rows characterizing their behavior over time. Data traveling on a cloud has a tremendous amount of information encoded within it. To enhance the detection and preserve the privacy of the data traveling, we need to not consider every single entity. Instead, we select the right entities that researchers can make the most out of, to improve the detection metrics, and analyze the traffic.

As we move forward, we must decide which entities in the cloud are considered sensitive. The solution to this is relatively simple: any feature in the data that is likely to be inaccessible for an adversary should be marked sensitive. For example, the process data and the generated logs in between clouds are not available to an external purveyor.

The inherent difficulties in defining the sensitivity of data may lead to useless data. For example, if any unique information in the cloud is considered sensitive, it is easy to imagine a scenario in which the data exhibits a homogeneous behavior.

This homogeneous data would be of no use to researchers interested in investigating anomalous behavior or seeking an accurate estimate of cloud data characteristics. In general, a measure of utility is derived by comparing the results of anonymized data with non-anonymized data. Section 3.5 elaborates on the solution to selecting the best quasi-identifiers/sensitive attributes in detail.

3.5 Alpha Distinction & Beta Separation in Attributes

In Section 3.4, we described the challenges of anonymizing network data. More specifically, the problem arises from the IDS attributes when we have to choose the best combination of attributes to reduce the risk to identify the user in the respective IDS region. To address that, we use the principle of alpha distinction and beta separation between key-attributes in network/cloud data or data in general. The said principle of alpha distinction and beta separation helps us in selecting the best Quasi Identifiers and sensitive attributes.

According to (Motwani & Xu, 2007), the α-distinction between attributes is defined as a subset of attributes that become the key in the table after removing of at most the 1−α fraction of attributes in the original table.

β-Separation is when we delete a minimum number of attributes such that there is no quasi-identifier with separation or a distinct ratio greater than β in the remaining attributes. Separation defines the degree to which combinations of attributes separate the records from each other, and distinction defines to which degree the attributes make records distinct. As Motwani explains, the separation between quasi-identifiers is always higher than the distinction between quasi-identifiers. For example, in the case of a table containing age, gender, and status as quasi-identifiers, if a combination of two of the three is chosen and the third is deleted, the combination is separate from other combinations.

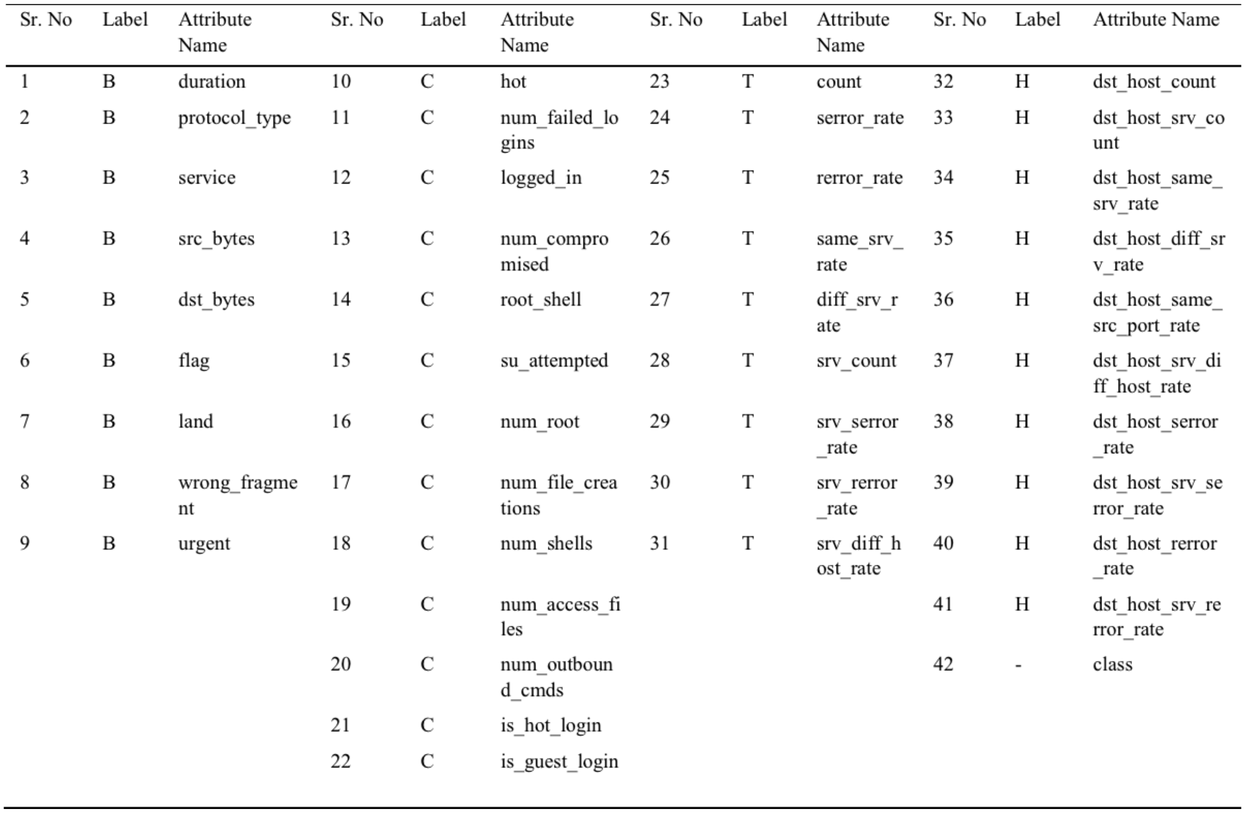
We utilize the ARX (Prasser & Kohlmayer, 2015) De-identification tool. ARX is a comprehensive open-source software for anonymizing sensitive personal data. It supports a wide variety of privacy and risk models, methods for transforming data, and methods for analyzing the usefulness of output data. The tool is used in various contexts, including industrially large data analytics platforms, research projects, data sharing in clinical trials, and training purposes. By using the ARX Anonymization tool[12], we have found a way to identify the combination of attributes in the NSL-KDD Dataset with the best separation between them and the best distinction. It has allowed us to select the attributes for use in data anonymization methods.

(P. Aggarwal & Sharma, 2015) explore the list of attributes present in the NSL-KDD dataset, where it has different attributes classified into four categories:

- Basic (B) Features are the attributes of individual TCP connections.
- Content (C) features are the attributes within a connection suggested by the domain knowledge.
- Traffic (T) features are the attributes computed using a two-second time window.
- Host (H) features are the attributes designed to assess attacks that last for more than two seconds.

Figure 5 below shows the different attributes.

[12] https://arx.deidentifier.org/



Sr. No	Label	Attribute Name	Sr. No	Label	Attribute Name	Sr. No	Label	Attribute Name	Sr. No	Label	Attribute Name
1	B	duration	10	C	hot	23	T	count	32	H	dst_host_count
2	B	protocol_type	11	C	num_failed_logins	24	T	serror_rate	33	H	dst_host_srv_count
3	B	service	12	C	logged_in	25	T	rerror_rate	34	H	dst_host_same_srv_rate
4	B	src_bytes	13	C	num_compromised	26	T	same_srv_rate	35	H	dst_host_diff_srv_rate
5	B	dst_bytes	14	C	root_shell	27	T	diff_srv_rate	36	H	dst_host_same_src_port_rate
6	B	flag	15	C	su_attempted	28	T	srv_count	37	H	dst_host_srv_diff_host_rate
7	B	land	16	C	num_root	29	T	srv_serror_rate	38	H	dst_host_serror_rate
8	B	wrong_fragment	17	C	num_file_creations	30	T	srv_rerror_rate	39	H	dst_host_srv_serror_rate
9	B	urgent	18	C	num_shells	31	T	srv_diff_host_rate	40	H	dst_host_rerror_rate
			19	C	num_access_files				41	H	dst_host_srv_rerror_rate
			20	C	num_outbound_cmds				42	-	class
			21	C	is_hot_login						
			22	C	is_guest_login						

Figure 5. List of Features in the NSL-KDD Dataset attributes

We run ARX tool on the NSL-KDD dataset and select the following attributes that have the largest distinction and separation ratios of 36.7293% and 98.75901%, respectively, therefore, we choose the combination of the following attributes in the NSL-KDD Dataset from the ARX Anonymization Tool as our quasi-identifiers (QI) and sensitive attributes and protocol_type, service and flag are categorical attributes:

- duration
- protocol_type (C)
- service (C)
- flag (C)
- src_bytes
- dst_bytes
- land
- wrong_fragment
- dst_host_rerror_rate

- dst_host_srv_rerror_rate

3.6 Data Publishing

After passing the original unmodified data through the four stages of our framework and verifying that the accuracy is more or less the same before and after anonymization. The data can be shared with other cooperative cloud administrators who wish to improve their IDS performance.

Processing the data over all four privacy-preserving strategies outlined is an optimal solution to protect the data in the best way. The information publisher decides to set the degree of anonymization and disclose it in compliance with its requirements. We, therefore, conclude our framework and demonstrate a method of sharing data between members in such a way that we maintain optimum accuracy, while at the same time preserving the privacy of user data.

Chapter 4 – Experiment and Results

4.1 Implementation

The dataset we used for our experimental study is the NSL-KDD dataset (Dhanabal & Shantharajah, 2015), an improvement over the existing KDD99 Cup dataset (Kayacik, Zincir-Heywood, & Heywood, 2005). To run our experiment for the large data set, we used an Anaconda Project configuration of Jupyter Notebooks powered by a 64-bit Windows 10 Desktop and supported by an x64 Intel ® Core™ i7-8700 CPU to implement the proposed framework. We used up to 32 GB RAM to help in our heavy calculations.

We used four methods to support our machine learning approaches: Random Forest, K-Nearest Neighbors, Support Vector Machines (SVMs), and Decision Trees.

4.1.1 Experiment Methodology

The accuracy and performance metrics can be considerably improved by selecting relevant features for an intrusion detection system. It is not feasible and practical to select all the attributes in the detection system, as we have seen for the NSL-KDD dataset.

However, in a large dataset, not all features represent the traffic. Therefore, reducing and selecting adequate features may improve the speed and accuracy of the intrusion detection system. In our approach, a feature selection-mechanism is used to eliminate non-relevant features and identify the features that contribute to improvement in the detection rate, based on the score of each feature established during the selection process. The resulting features are used in the chosen machine learning approaches. Section 4.1.2, we elaborate on the steps we have followed in the analysis of our framework.

4.1.2 Steps

The experiment follows several steps to clean and parse the data used in this thesis. We elaborate on:

Data Preprocessing: Because of numerical and non-numeric instances in the data set, the NSL-KDD dataset has been pre-processed. The classifier defined in scikit-learn (Pedregosa et al., 2011) works well with numerical data, as our dataset has numerical and categorical data, we use one-of-K or one-hot encoding system. This technique converts each categorical feature into binary**.**

Feature Scaling: A common requirement in machine learning to avoid features with large values that weigh too much on the performance. Scaling calculates the mean for each feature, subtracting the mean value from the value of that feature, and finally dividing the result by its standard deviation. As a result, features with uniform and equal weight on the performance are used.

Feature Selection: This step eliminates redundant and irrelevant data. It is a method where a subset of relevant features that fully represents the problem is chosen. It is ideal to restrict the number of features for the following reasons: First, there is a possibility that irrelevant features could suggest correlations between features and target classes that arise just by chance and do not correctly model it. This aspect is related to overfitting, usually in a decision tree classifier. Second, a vast number of features could significantly increase the computation time without a corresponding classifier improvement.

Once the best subset of features is found, a recursive feature elimination was applied to repeatedly build a model, placing the feature aside and repeating the process with the remaining features until all features in the dataset are exhausted.

Prediction: The recursive feature elimination chooses the features each class of attacks, DoS, Probe, R2L, and U2R.We fit our machine learning classifiers to each class of attack, and finally allow it to calculate the accuracy.

Table 1. Selected Features After Recursive Feature Elimination

Attack Class	Features Selected
Denial of Service	“src_bytes”, “dst_bytes”, “wrong_fragment”, “count”, “srv_count”, “same_srv_rate”, “dst_host_diff_srv_rate”, “dst_host_same_src_port_rate”, “dst_host_serror_rate”, “Protocol_type_icmp”, “service_ecr_i”,’flag_S0’, “flag_SF.”
Probe	'src_bytes', 'dst_bytes', 'count', 'rerror_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'dst_host_rerror_rate', 'service_eco_i', 'service_private'.
R2L	'duration','src_bytes', 'dst_bytes', 'hot', 'num_failed_logins', 'is_guest_login', 'srv_count', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'dst_host_serror_rate', 'service_ftp_data'
U2R	'duration', 'src_bytes', 'dst_bytes', 'hot', 'num_compromised', 'root_shell', 'num_file_creations', 'count', 'dst_host_count', 'dst_host_srv_count', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate'

4.2 Results

The accuracy and precision as the evaluation metrics give us the best way to compare the detection in the case of Intrusion Detection Systems. The accuracy score shows the number of correct predictions divided by the total number of predictions**.** The precision score attempts to answer the question as to what proportion of predictions were actually correct.

Table 2. Normal NSL-KDD Dataset

DoS		Accuracy	Precision	Recall	F-Score
	RF	0.99802	0.99799	0.99678	0.99779
	KNN	0.99715	0.99678	0.99665	0.99672
	SVM	0.99371	0.99107	0.99450	0.99278
	DT	0.99639	0.99505	0.99665	0.99585
Probe		Accuracy	Precision	Recall	F-Score
	RF	0.99654	0.99509	0.99262	0.99469
	KNN	0.99077	0.98606	0.98508	0.98553
	SVM	0.98450	0.96907	0.98365	0.97613
	DT	0.99571	0.99391	0.99267	0.99329
R2L		Accuracy	Precision	Recall	F-Score
	RF	0.97968	0.97436	0.96999	0.97213
	KNN	0.96737	0.95311	0.95484	0.95389
	SVM	0.96793	0.94584	0.96264	0.95529
	DT	0.97920	0.97150	0.96957	0.97050
U2R		Accuracy	Precision	Recall	F-Score
	RF	0.99755	0.94916	0.86641	0.89226
	KNN	0.99703	0.93282	0.84835	0.87754
	SVM	0.99652	0.91988	0.83931	0.85918
	DT	0.99663	0.86841	0.91672	0.88628

Table 3. Differential Privacy Perturbation

DoS		Accuracy	Precision	Recall	F-Score
	RF	0.99796	0.99839	0.99611	0.99738
	KNN	0.99377	0.99185	0.99383	0.99284
	SVM	0.98975	0.98484	0.99169	0.98824
	DT	0.99668	0.99559	0.99678	0.99618
Probe		Accuracy	Precision	Recall	F-Score
	RF	0.99596	0.99566	0.99184	0.99417
	KNN	0.98937	0.98234	0.98452	0.98340
	SVM	0.98368	0.96674	0.98376	0.97493
	DT	0.99522	0.99300	0.99205	0.99251
R2L		Accuracy	Precision	Recall	F-Score
	RF	0.98230	0.97560	0.97445	0.97467
	KNN	0.96912	0.95615	0.95671	0.95631
	SVM	0.96626	0.94732	0.95887	0.95284
	DT	0.98229	0.97576	0.97402	0.97488
U2R		Accuracy	Precision	Recall	F-Score
	RF	0.99755	0.94929	0.84146	0.88454
	KNN	0.99673	0.94217	0.81154	0.84776
	SVM	0.99652	0.91988	0.83981	0.85918
	DT	0.99724	0.89882	0.91465	0.89795

Table 4. K-Anonymity Generalization

DoS		Accuracy	Precision	Recall	F-Score
	RF	0.98975	0.99737	0.96086	0.98207
	KNN	0.98766	0.98787	0.98338	0.98548
	SVM	0.98567	0.97803	0.98928	0.98361
	DT	0.96455	0.97296	0.93378	0.94603
Probe		Accuracy	Precision	Recall	F-Score
	RF	0.98584	0.98955	0.97801	0.96921
	KNN	0.98459	0.9772	0.97351	0.97536
	SVM	0.98154	0.96605	0.97733	0.97127
	DT	0.97637	0.96791	0.94740	0.94761
R2L		Accuracy	Precision	Recall	F-Score
	RF	0.96691	0.96699	0.93997	0.94293
	KNN	0.96174	0.94990	0.93989	0.94309
	SVM	0.95817	0.94319	0.93892	0.93719
	DT	0.96873	0.95894	0.95002	0.95357
U2R		Accuracy	Precision	Recall	F-Score
	RF	0.99755	0.95329	0.84488	0.87567
	KNN	0.99683	0.89667	0.87071	0.87525
	SVM	0.99673	0.91761	0.83991	0.86317
	DT	0.99509	0.85341	0.88050	0.85699

Table 5. L-Diversity Generalization

DoS		Accuracy	Precision	Recall	F-Score
	RF	0.99627	0.99864	0.99223	0.99553
	KNN	0.99196	0.98751	0.99424	0.99083
	SVM	0.98260	0.97801	0.99062	0.98424
	DT	0.99266	0.99201	0.99102	0.99149
Probe		Accuracy	Precision	Recall	F-Score
	RF	0.99613	0.99597	0.99154	0.99379
	KNN	0.98797	0.98037	0.98287	0.98139
	SVM	0.98360	0.96655	0.98371	0.97481
	DT	0.99456	0.99226	0.99072	0.99147
R2L		Accuracy	Precision	Recall	F-Score
	RF	0.98166	0.97689	0.97193	0.97374
	KNN	0.97031	0.95817	0.95796	0.95800
	SVM	0.96602	0.94726	0.95811	0.95245
	DT	0.97991	0.97286	0.97016	0.97148
U2R		Accuracy	Precision	Recall	F-Score
	RF	0.99795	0.97178	0.82733	0.89803
	KNN	0.99703	0.91444	0.87081	0.88106
	SVM	0.99683	0.91586	0.84705	0.86701
	DT	0.99652	0.88166	0.87293	0.87214

Table 6. T-Closeness Generalization

DoS		Accuracy	Precision	Recall	F-Score
	RF	0.99779	0.99893	0.99678	0.99792
	KNN	0.99447	0.99146	0.99584	0.99365
	SVM	0.98794	0.98130	0.99115	0.98619
	DT	0.99651	0.99558	0.99638	0.99598
Probe		Accuracy	Precision	Recall	F-Score
	RF	0.99629	0.99592	0.99257	0.99391
	KNN	0.99044	0.98522	0.98488	0.98502
	SVM	0.98343	0.96667	0.98299	0.97453
	DT	0.99522	0.99330	0.99715	0.99251
R2L		Accuracy	Precision	Recall	F-Score
	RF	0.98206	0.97581	0.97274	0.97495
	KNN	0.97190	0.96075	0.95972	0.96017
	SVM	0.96697	0.94839	0.95970	0.95382
	DT	0.97592	0.97279	0.96905	0.97088
U2R		Accuracy	Precision	Recall	F-Score
	RF	0.99724	0.94942	0.85554	0.86293
	KNN	0.99703	0.91444	0.87081	0.88106
	SVM	0.99673	0.91761	0.83991	0.86317
	DT	0.99673	0.87910	0.89550	0.88642

We now provide a visual comparison of the accuracy achieved before and after applying the proposed framework. We have structured accuracy scores and precision scores for every class of attack, on each class of privacy protection technique in a chronological order.

4.2.1 Differential Privacy Perturbation

In Figures 6 to 13, we show the accuracy and precision obtained by the proposed model considering different attacks and different machine learning models (i.e., RF, KNN, SVM, and DT). The results reveal that our framework is resilient to the modifications applied to data using DP perturbation. The results show that the percentage of accuracy that is preserved is 99.6% for the adjustments on data. By using our model, the accuracy decreases by the differential privacy perturbation, which is only 0.04%, has no major impact, and is disregarded. These results suggest that the consulted IDSs can detect intrusions without the need to use the original data. The same results (i.e., no significant degradation has been recorded) can be achieved using the newly generated data, which is similar to the original data semantically but not synthetically.

We elaborate on the results in detail. The first phase of differentially perturbing our data shows a slight decrease in accuracy and precision scores when there is noise (Kalapanidas, Avouris, Craciun, & Neagu).

As we observe in Figures 6 and 7, the DoS attack class accuracy scores show negligible change when differentially private data is used for the various machine learning algorithms. The variation can be as low as 0.01% and 0.02%. The maximum change shown in KNN is just 0.396%. The precision score for KNN shows a maximum change of 0.623%, and the other machine learning methods mirror the accuracy score.

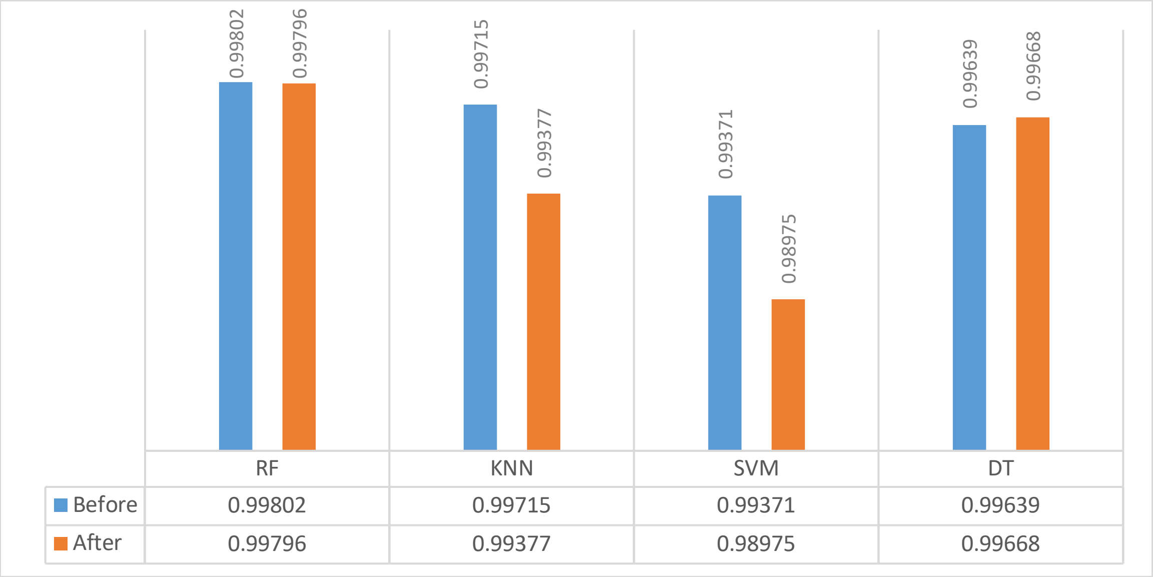


Figure 6. Accuracy Scores for DoS Perturbation

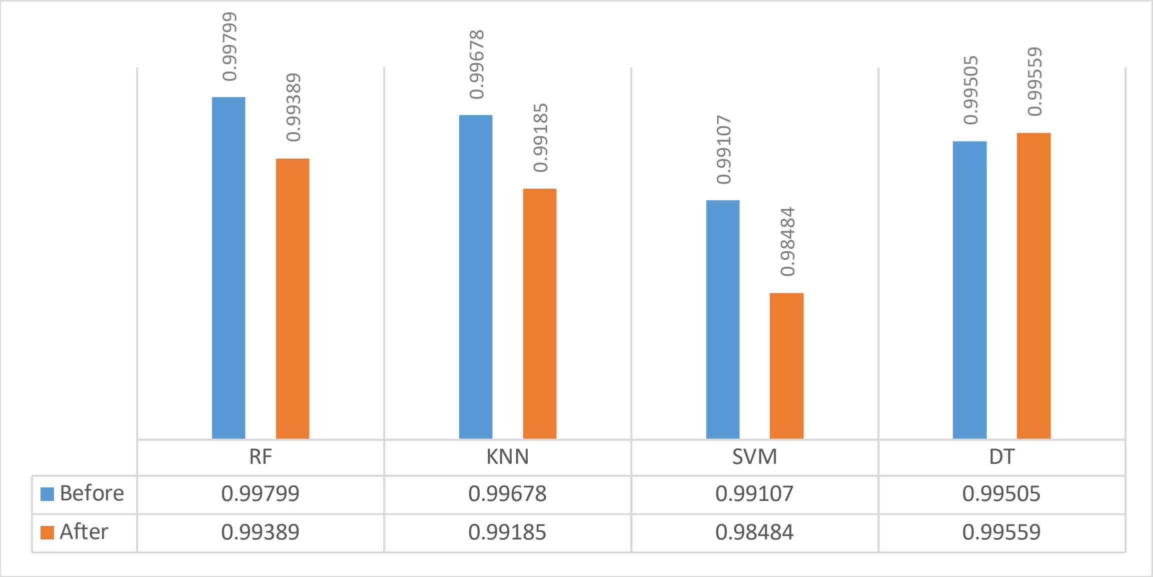
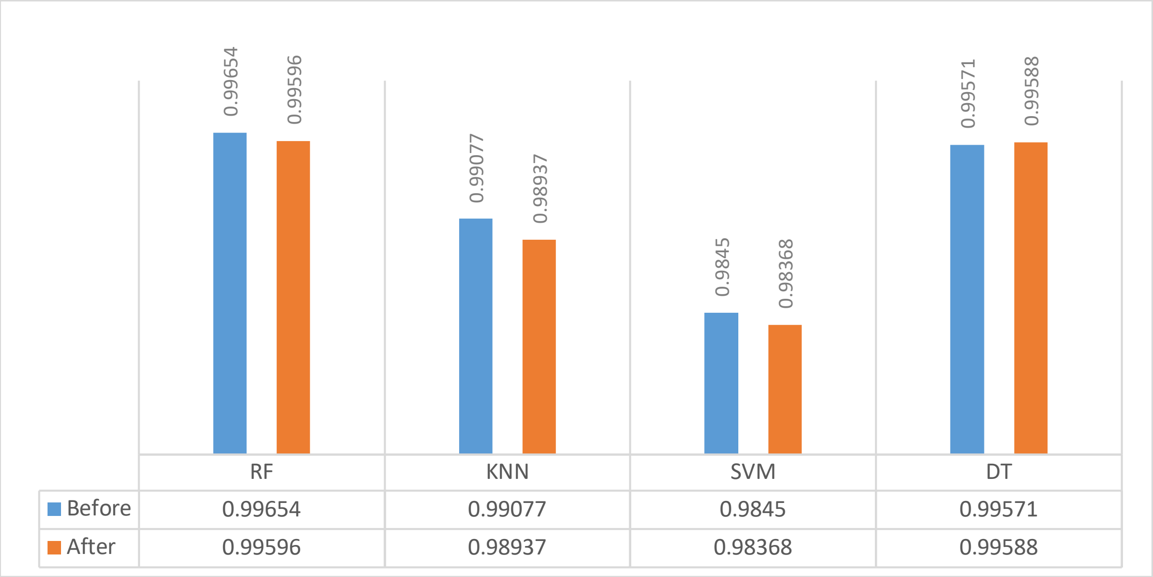


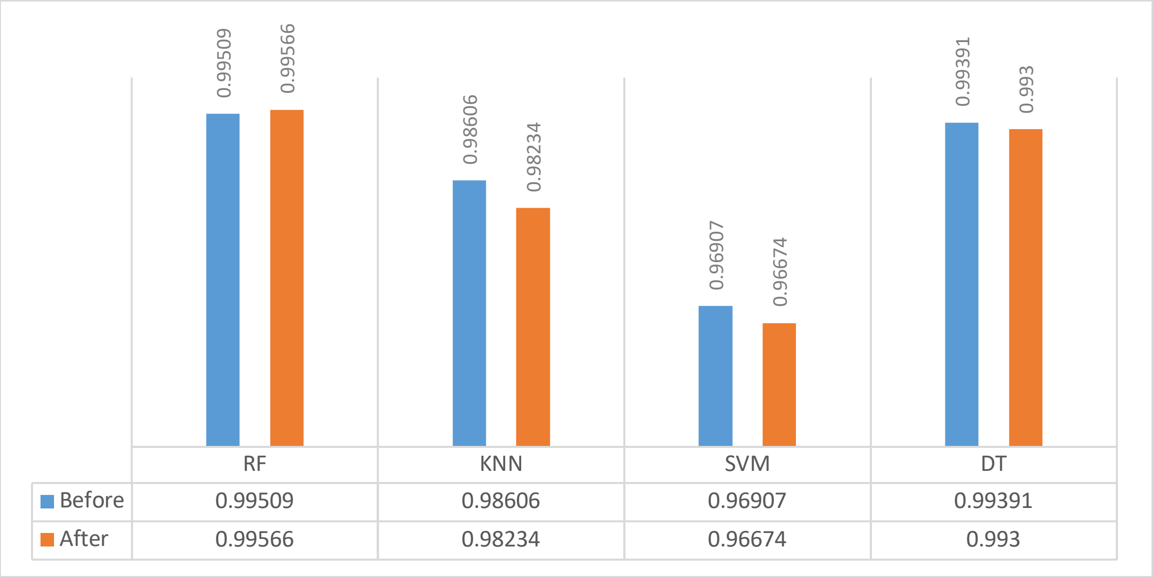
Figure 7. Precision Scores for DoS Perturbation

As we observe in Figures 8 and 9, there is a negligible change in the accuracy and precision scores when differentially private data is used for probe attack class in the various machine learning algorithms. The change we observe can be as low as 0.01% and 0.02%. The precision score for all the machine learning techniques is more or less the same.



	RF	KNN	SVM	DT
🟦 Before	0.99654	0.99077	0.9845	0.99571
🟧 After	0.99596	0.98937	0.98368	0.99588

Figure 8. Accuracy Scores for Probe Perturbation



	RF	KNN	SVM	DT
🟦 Before	0.99509	0.98606	0.96907	0.99391
🟧 After	0.99566	0.98234	0.96674	0.993

Figure 9. Precision Scores Probe Perturbation

In Figure 10 and 11, there is a negligible variation in the accuracy scores for differentially private data used in the R2L attack class for the various machine learning algorithms. The change can be as low as 0.01% and 0.02%. The accuracy is improved for KNN, Decision Trees, and Random Forest by a small amount. The precision score for DT shows an improvement of 0.06%.

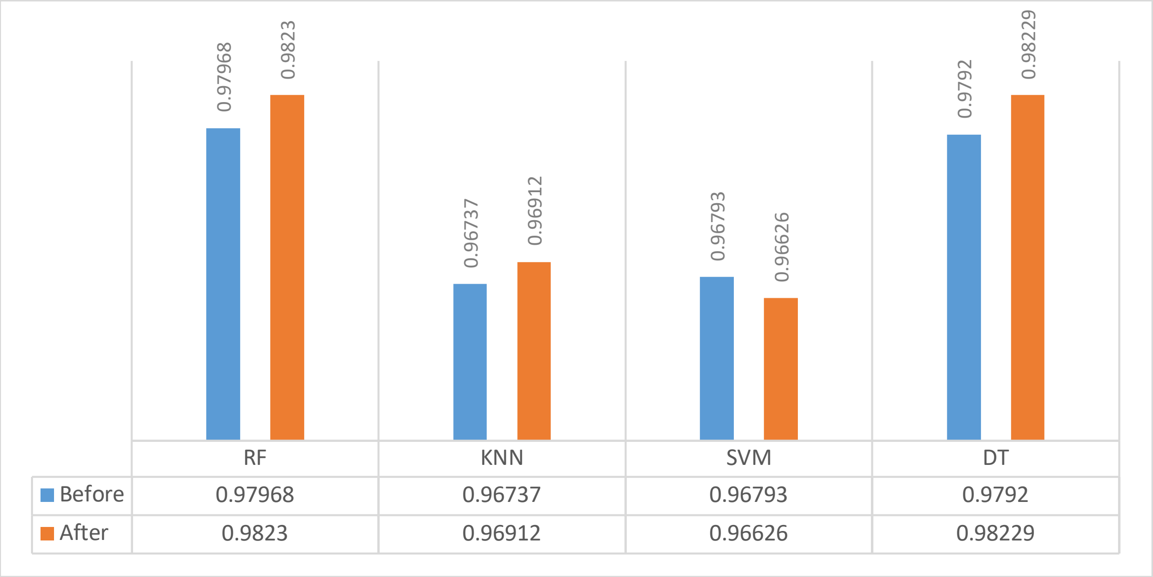


Figure 10. Accuracy Scores for R2L Perturbation

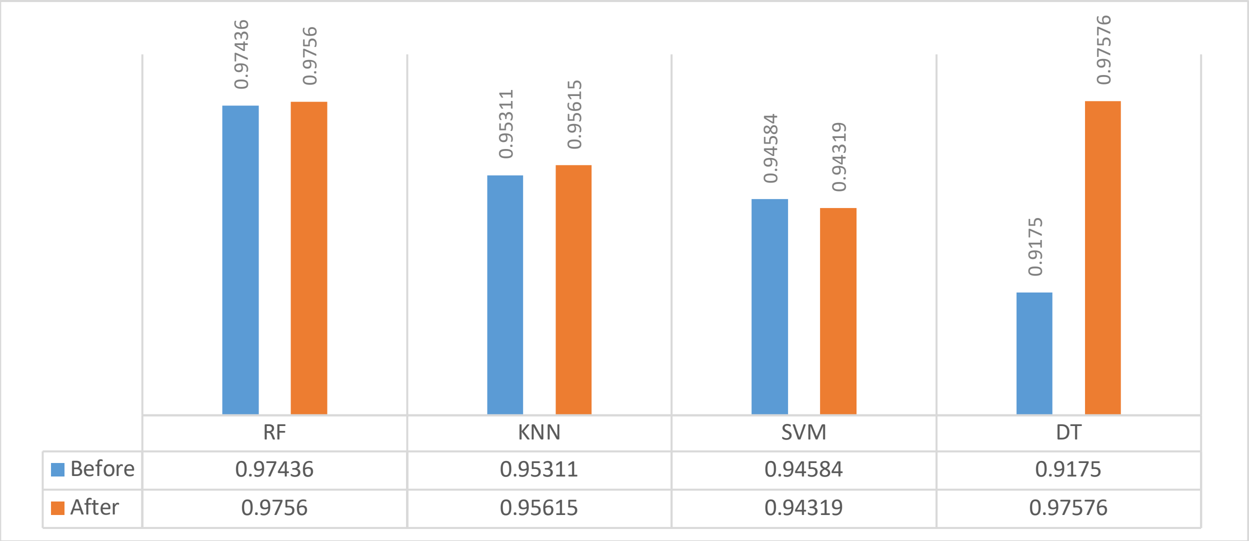
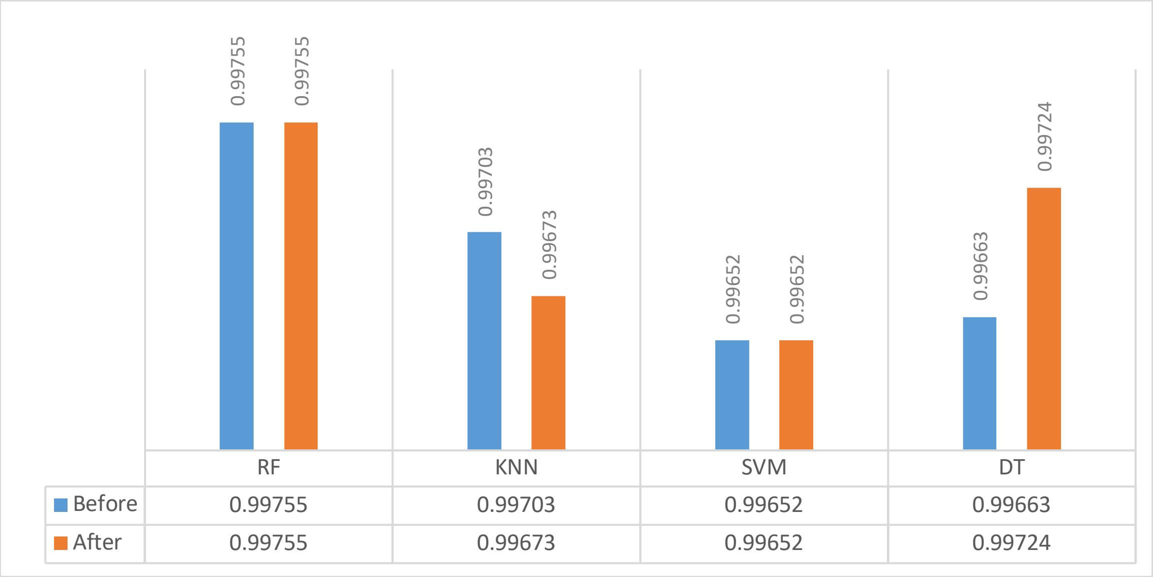


Figure 11. Precision Scores for R2L Perturbation

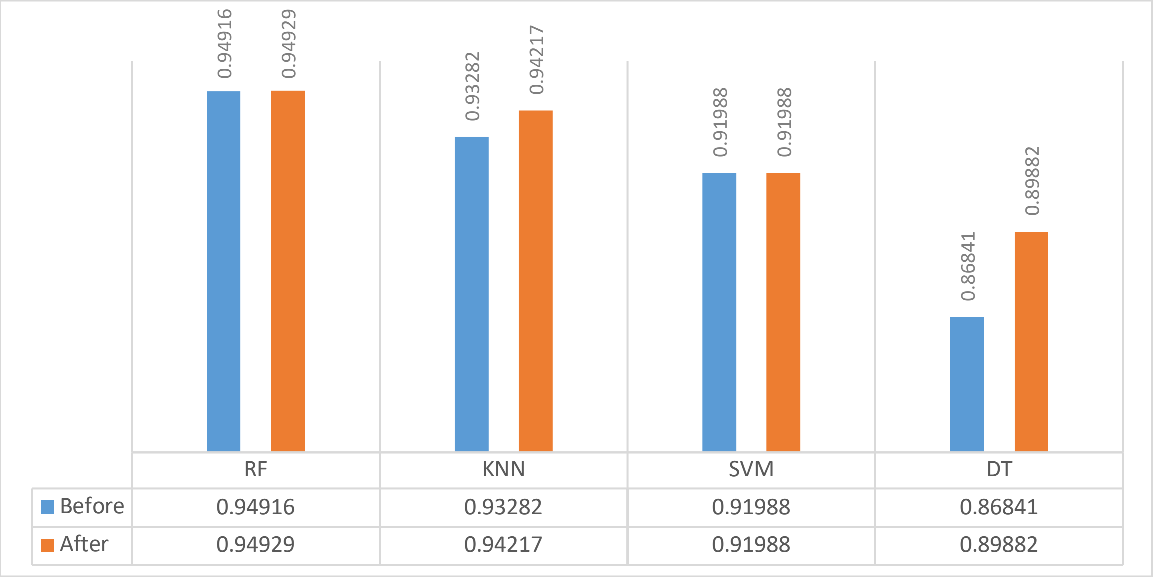
In Figure 11, there is negligible or no change in the accuracy scores for differentially private data used in the U2R attack class for different machine learning algorithms. We see that the change is as low as 0.01% and 0.02%, respectively. The accuracy in Decision Trees is improved by a small amount.



	RF	KNN	SVM	DT
🟦 Before	0.99755	0.99703	0.99652	0.99663
🟧 After	0.99755	0.99673	0.99652	0.99724

Figure 12. Accuracy Scores for U2R Perturbation

In Figure 13, there is also a negligible or no change in the precision scores for differentially private data used in the U2R attack class for the different machine learning algorithms. We notice that the shift is as small as 0.01% and 0.02%. For RF and KNN. The precision score is increased by a small amount, and for DT, there is a 0.03% improvement.



	RF	KNN	SVM	DT
🟦 Before	0.94916	0.93282	0.91988	0.86841
🟧 After	0.94929	0.94217	0.91988	0.89882

Figure 13. Precision Scores U2R Perturbation

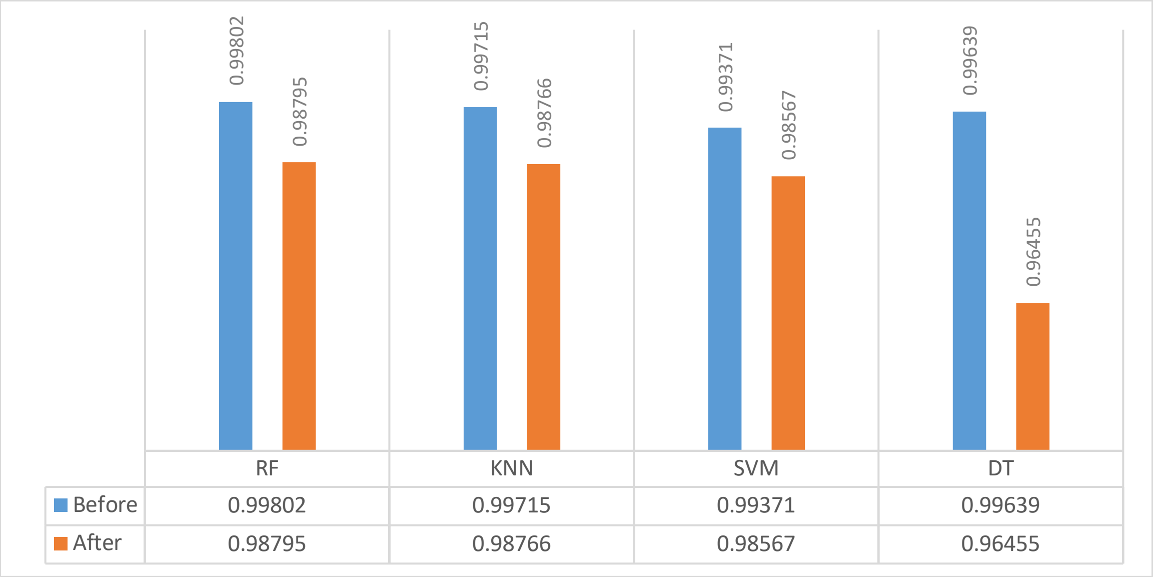
As we have seen, accuracies are similar when the data is gently perturbed and passed to the chosen machine learning algorithms. It means that the accuracies are not drastically changing but are staying close to the accuracy before perturbation.

4.2.2 K-Anonymization Generalization

We now present our findings for the k-anonymity method of anonymizing data. Figures 14 to 21 show the average accuracy and precision obtained by the proposed model considering different attacks and different machine learning models. The results reveal that our framework is also resilient to the modifications applied to data using k-anonymization generalization. The results also show that the percentage of accuracy preserved when the data is adjusted is 99.7%. This means that, by using our model, the decrease of accuracy by the k-anonymity perturbation is only 0.03%, which has no significant impact and can be disregarded. These results suggest that the consulted IDSs detect intrusions without the need to use the original data. There was no record of significant degradation using the newly generated data, which is similar to the original data.

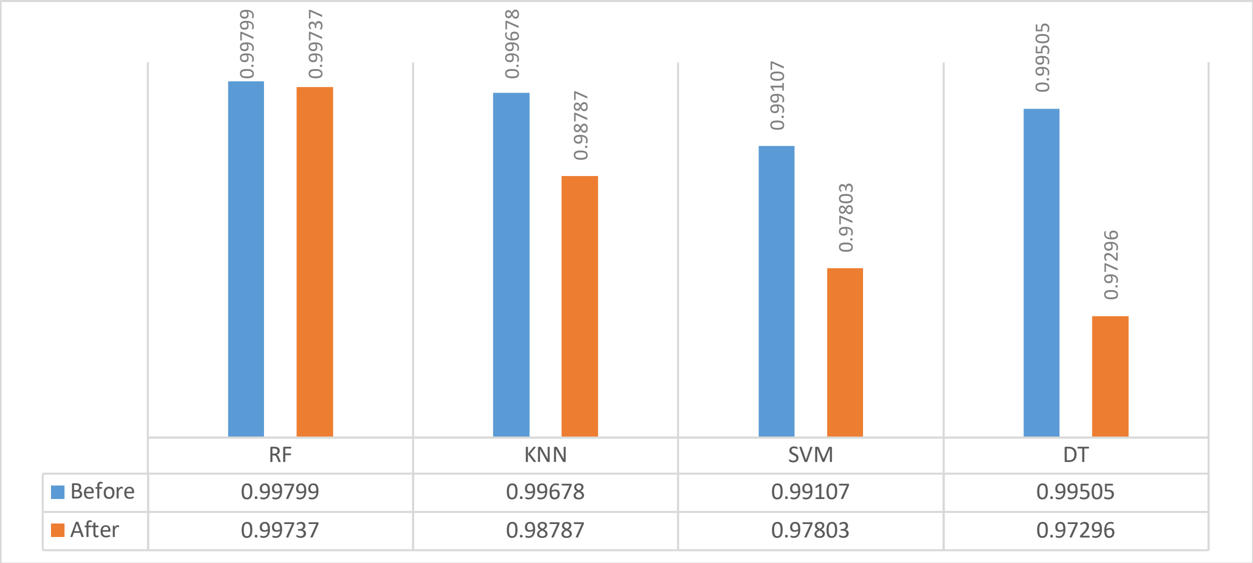
In the second phase of our framework, we considered using the k-anonymization generalization method on perturbed data. We then noted that the noise we added to the dataset was generalized. Furthermore, we can see that generalization can be extended to noise (Kodratoff, Manago, & Blythe, 1987), and thus, we see some cases where the accuracy and precision metrics are showing a minor improvement/negligible decrease.

As shown in Figures 14 and 15, there is a negligible change in the accuracy scores for the k-anonymized data used by various machine learning algorithms in the DoS attack class. The change we observe is as low as 0.01% and 0.02%, respectively. The precision score shows a slight decrease in accuracy for almost all machine learning methods. There is also a slight decrease in the precision score.



	RF	KNN	SVM	DT
🟦 Before	0.99802	0.99715	0.99371	0.99639
🟧 After	0.98795	0.98766	0.98567	0.96455

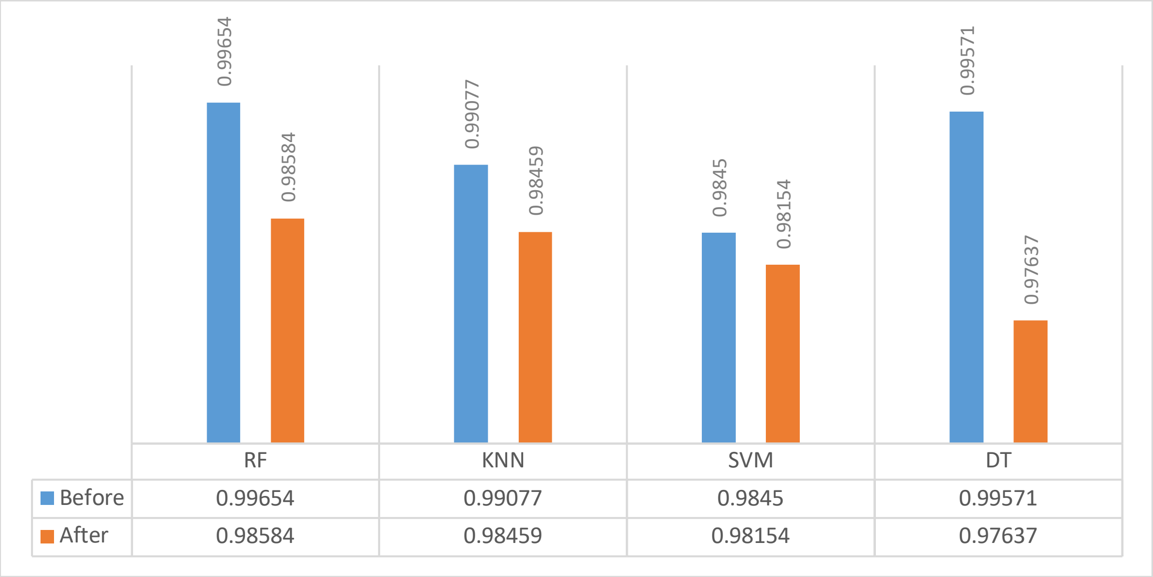
Figure 14. Accuracy Scores for DoS K-Anonymization



	RF	KNN	SVM	DT
🟦 Before	0.99799	0.99678	0.99107	0.99505
🟧 After	0.99737	0.98787	0.97803	0.97296

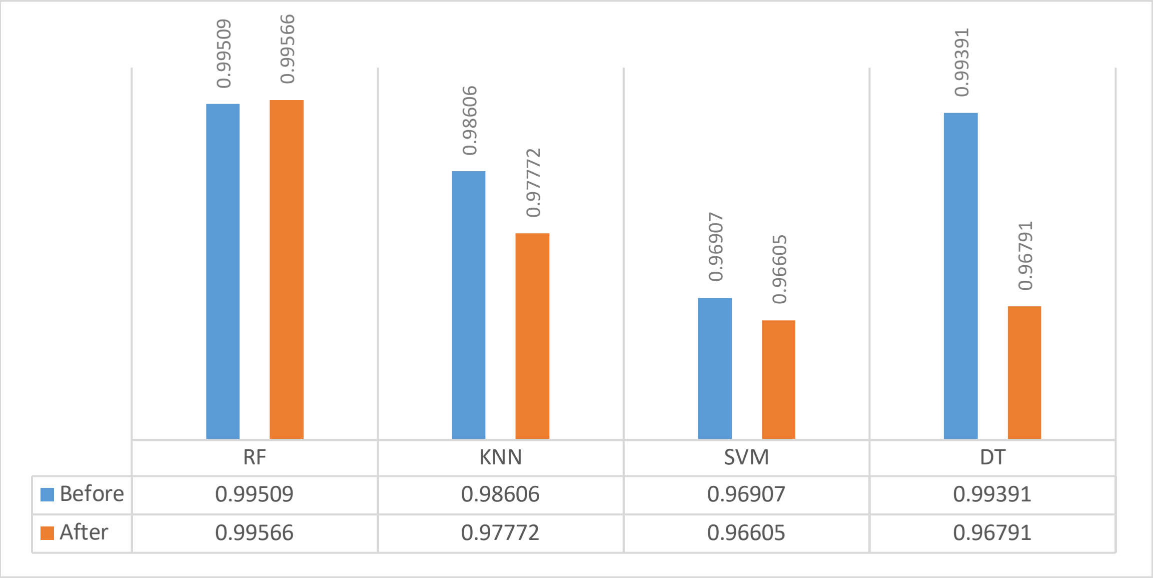
Figure 15. Precision Scores for DoS K-Anonymization

As shown in Figures 16 and 17, there is a small variation in the accuracy scores for the k-anonymized data used in the probe attack class for the various machine learning algorithms. The change we observe can be too low, at 0.01% and 0.02%. The precision score also shows the same behavior, while the RF score increases minimally.



	RF	KNN	SVM	DT
🟦 Before	0.99654	0.99077	0.9845	0.99571
🟧 After	0.98584	0.98459	0.98154	0.97637

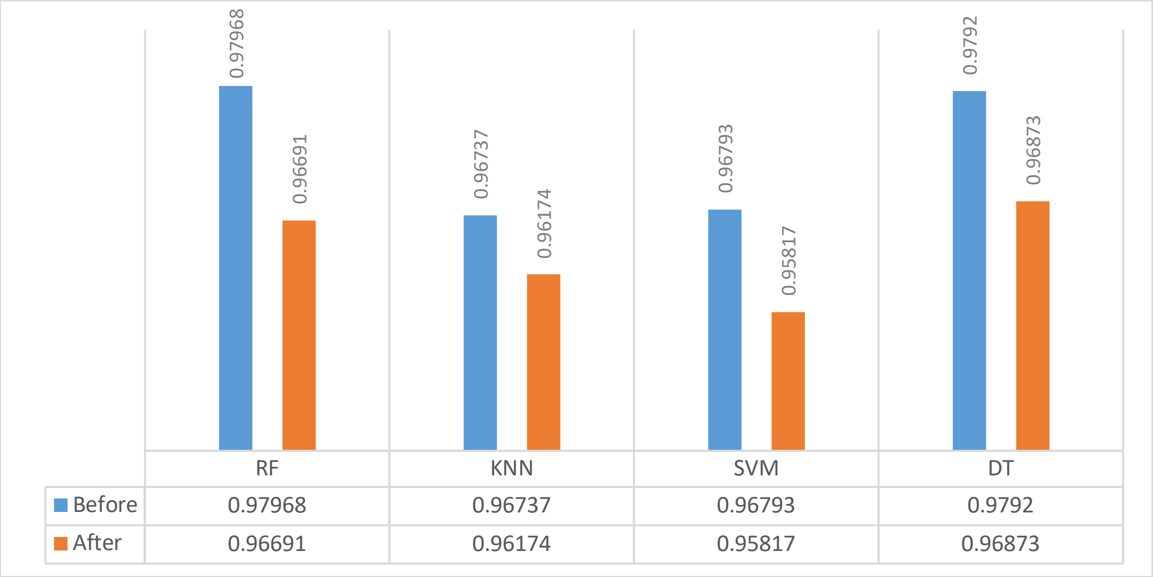
Figure 16. Accuracy Scores for Probe K-Anonymization



	RF	KNN	SVM	DT
🟦 Before	0.99509	0.98606	0.96907	0.99391
🟧 After	0.99566	0.97772	0.96605	0.96791

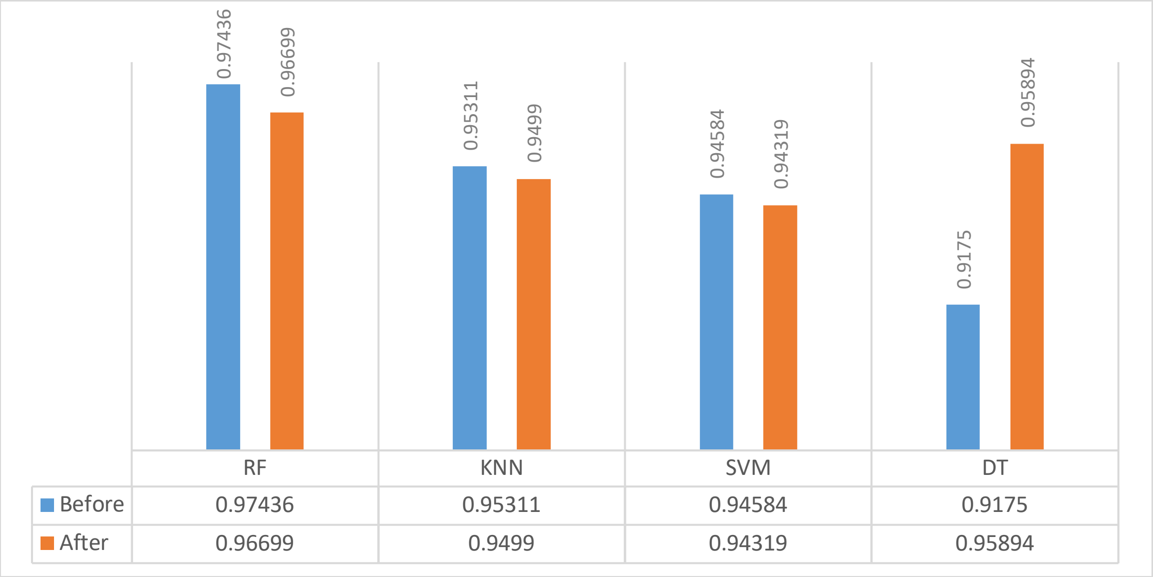
Figure 17. Precision Scores for Probe K-Anonymization

In Figures 18 and 19, there is a negligible variation in the accuracy scores for k-anonymized data used in the R2L attack class for the various machine learning algorithms. We observe a minimal change of 0.01% and 0.02%. The precision score is negligibly changing, but for DT, it increases by 0.04%.



	RF	KNN	SVM	DT
🟦 Before	0.97968	0.96737	0.96793	0.9792
🟧 After	0.96691	0.96174	0.95817	0.96873

Figure 18. Accuracy Scores for R2L K-Anonymization



	RF	KNN	SVM	DT
🟦 Before	0.97436	0.95311	0.94584	0.9175
🟧 After	0.96699	0.9499	0.94319	0.95894

Figure 19. Precision Scores for R2L K-Anonymization

In Figures 20 and 21, the accuracy scores for the k-anonymized data used in the U2R attack class for the various machine learning algorithms varies negligibly. We see small changes of 0.01% and 0.02%, respectively. The precision score changes negligibly, while the RF score increases minimally.

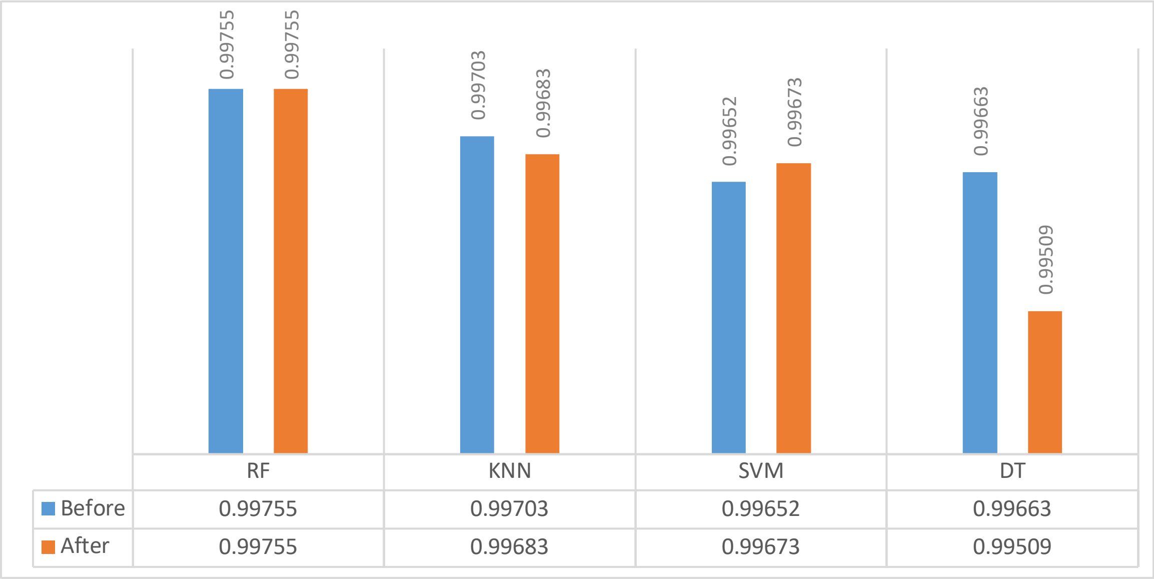


Figure 20. Accuracy Scores for U2R K-Anonymization

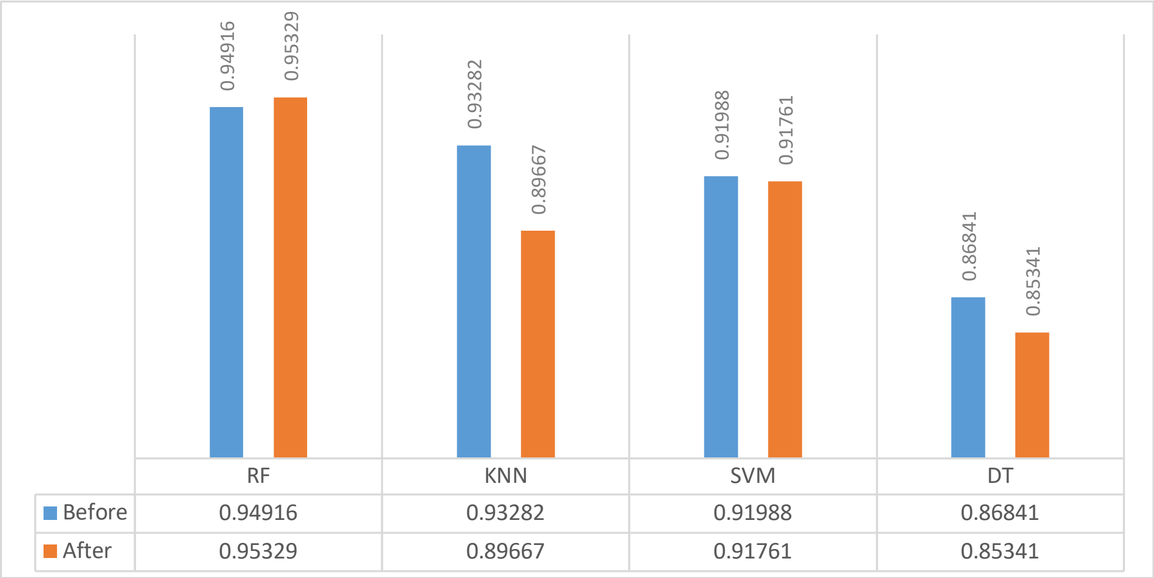
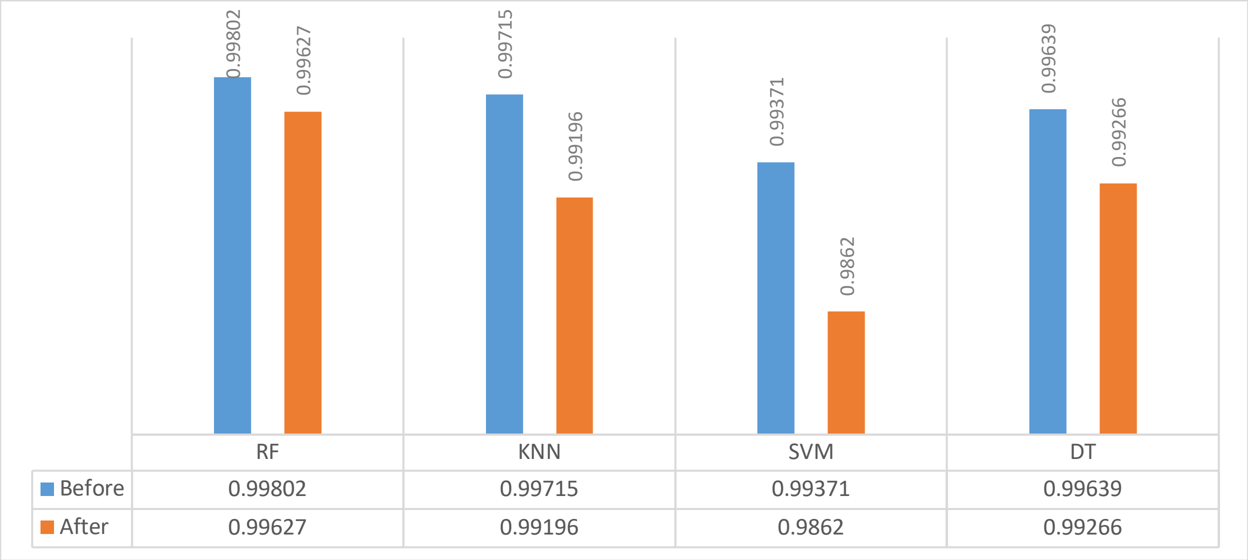


Figure 21. Precision Scores for U2R K-Anonymization

4.2.3 L-Diversity Generalization

We now present our findings for the L-Diversity Method of Anonymizing Data. Figures 22 to 29 show the accuracy and precision obtained by the proposed model considering different attacks and different machine learning models. The results reveal that our framework is also resilient to the modifications applied to data using L-diversity generalization. The results also show that the percentage of accuracy preserved for the adjustments on data is 99.7%. By using our model, the decrease of accuracy by the L-diversity generalization is only 0.03%, which has no significant impact and is disregarded. These results suggest that the consulted IDSs detect intrusions without needing to use the original data. The same results can be achieved using the newly generated data, which is similar to the original data semantically but not synthetically.

In the third phase of our framework we have the input data accrued from k-anonymized data. As shown in Figures 22 and 23, the accuracy scores for the L-Diverse data used in the Probe Attack class show a negligible change in the various machine learning algorithms. The change we observe can be very low at 0.01% and 0.02%. The precision score has been negligibly changed and slightly increases for RF.



	RF	KNN	SVM	DT
🟦 Before	0.99802	0.99715	0.99371	0.99639
🟧 After	0.99627	0.99196	0.9862	0.99266

Figure 22. Accuracy Scores for DoS L-Diversity

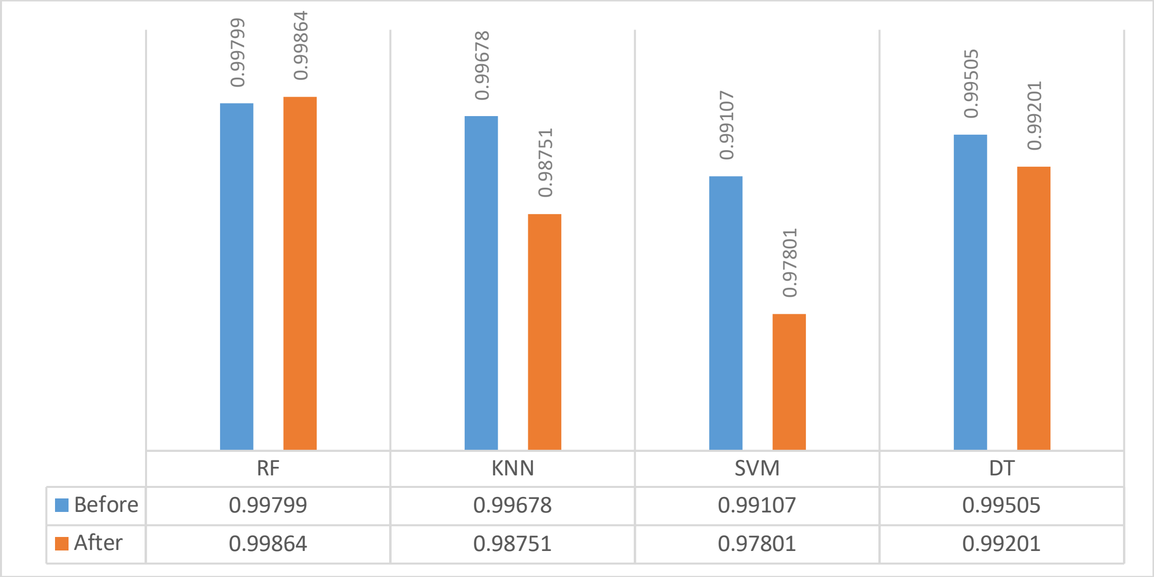


Figure 23. Precision Scores for DoS L-Diversity

In Figures 24 and 25, there is a negligible change in the accuracy scores for L-Diverse data utilized in the Probe attack class, the various machine learning algorithms. The change can be as low as 0.01% and 0.02%. The precision score is negligibly changing and slightly increases for RF.

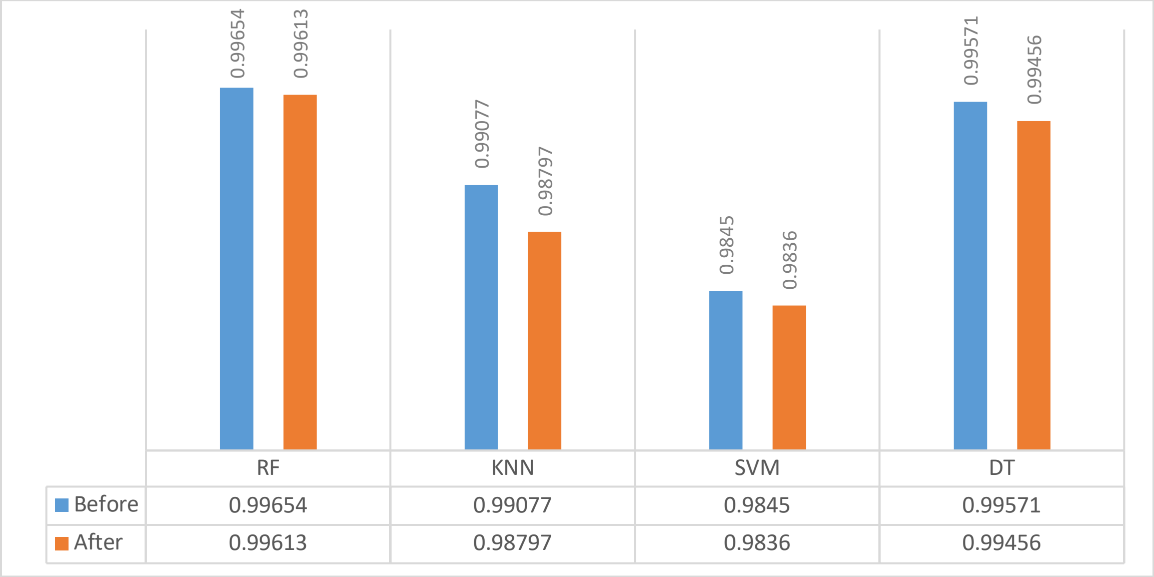


Figure 24. Accuracy Scores for Probe L-Diversity

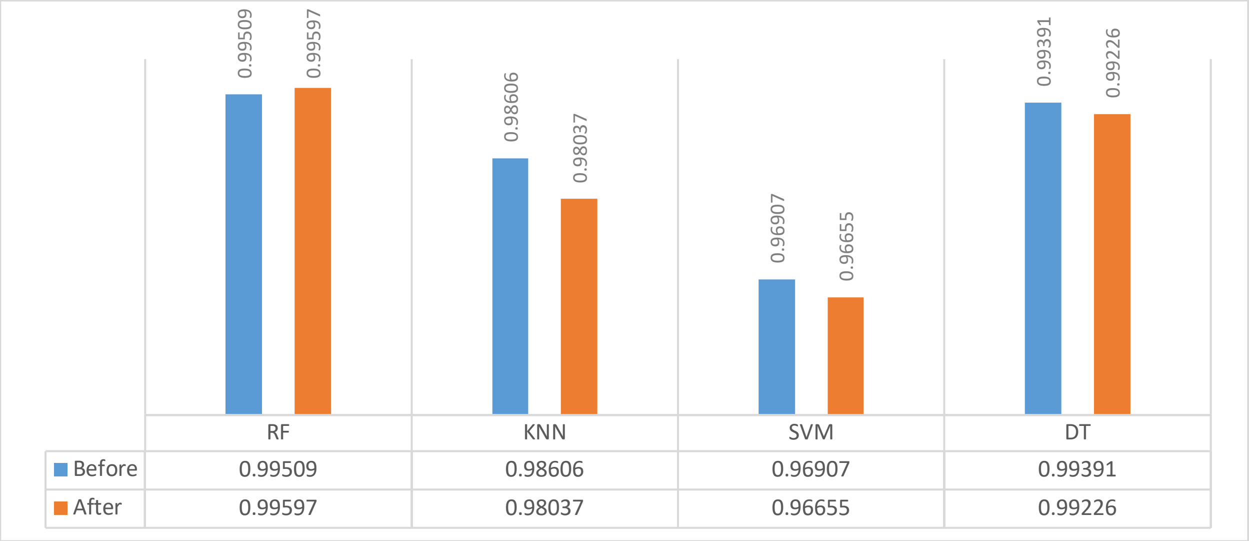


Figure 25. Precision Scores for Probe L-Diversity

In Figures 26 and 27, the accuracy scores for L-Diverse data utilized in the R2L attack class, the various machine learning algorithms show a slight improvement except for the SVM technique where it slightly reduces. The difference we observe can be as minor as 0.01% and 0.02%. The precision score increases minimally for RF, KNN, SVM while it shows a 0.06% increase for DT.

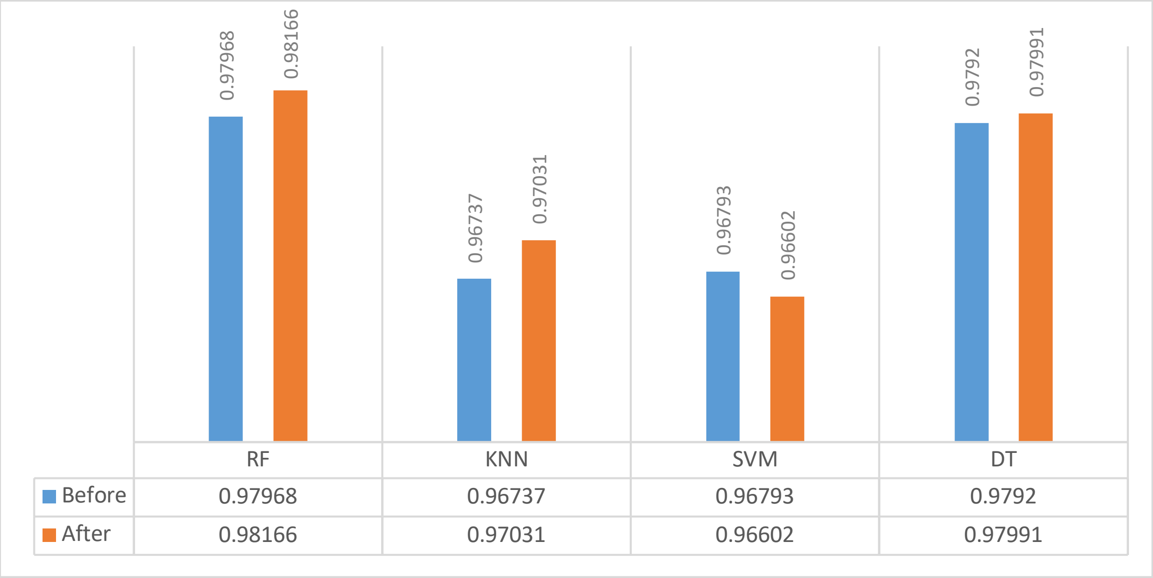
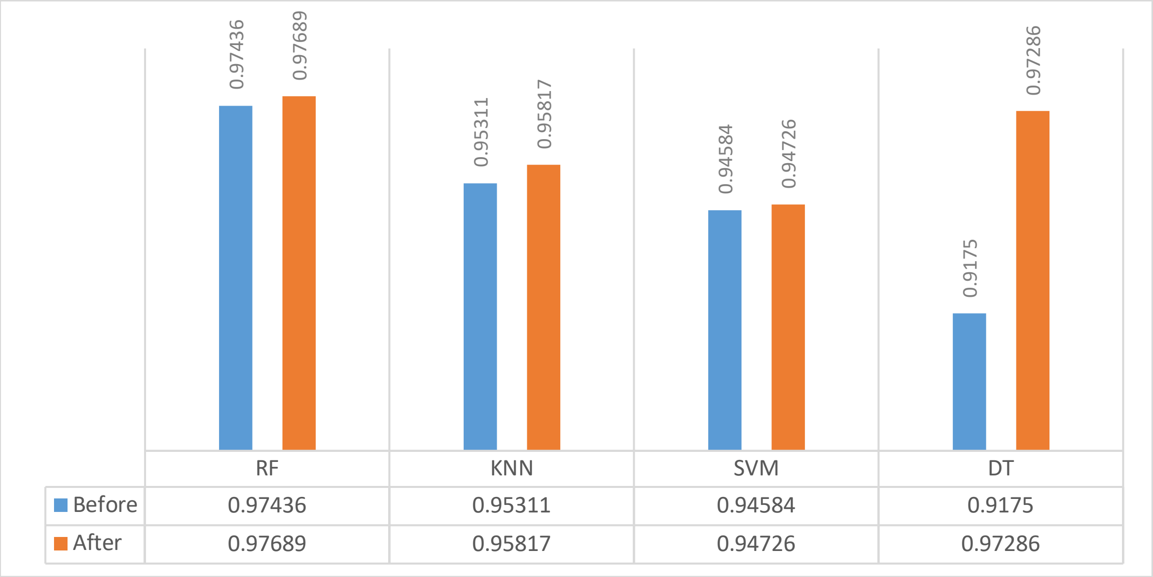


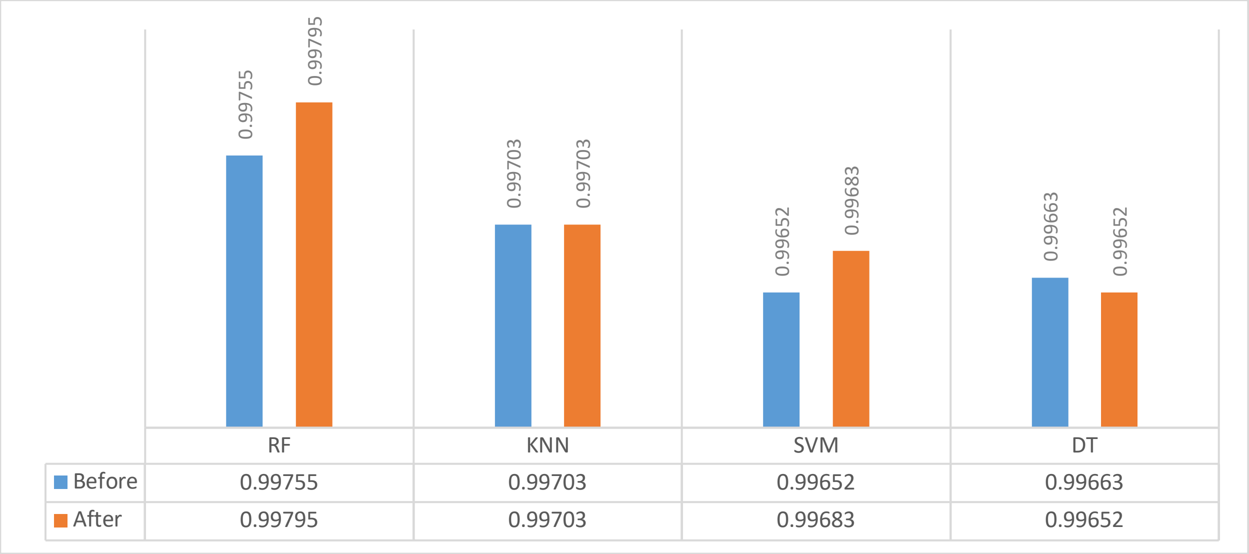
Figure 26. Accuracy Scores for R2L L-Diversity



	RF	KNN	SVM	DT
🟦 Before	0.97436	0.95311	0.94584	0.9175
🟧 After	0.97689	0.95817	0.94726	0.97286

Figure 27. Precision Scores for R2L L-Diversity

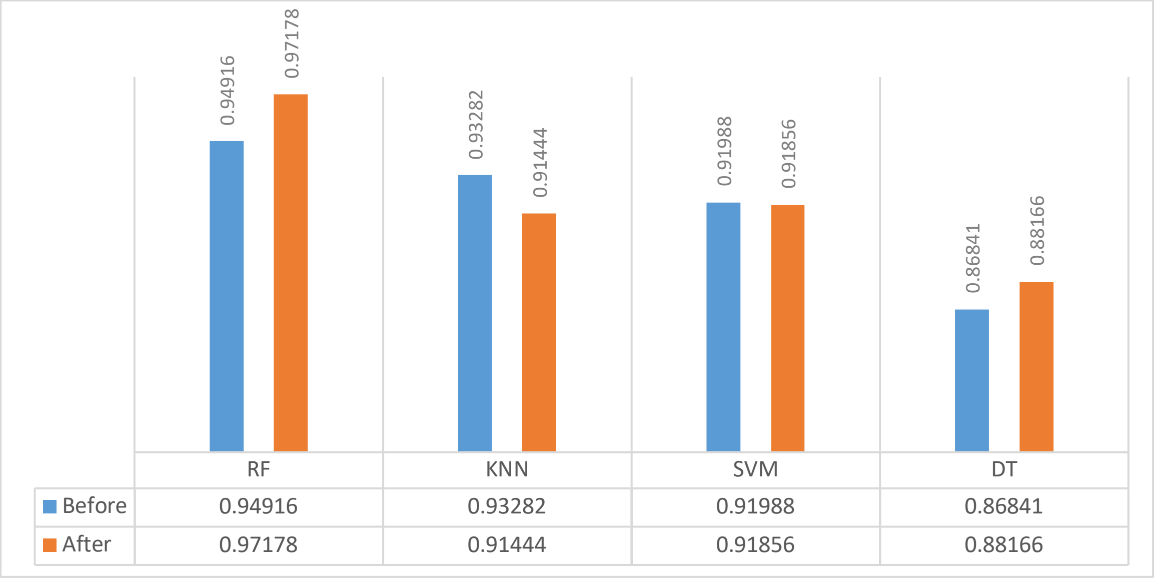
As shown in Figure 28, the accuracy scores for L-Diverse data used in the U2R attack class, the various machine learning algorithms show a slight improvement or remain the same, except for the DT technique where it reduces slightly. The difference can be less than 0.01% and 0.02%.



	RF	KNN	SVM	DT
🟦 Before	0.99755	0.99703	0.99652	0.99663
🟧 After	0.99795	0.99703	0.99683	0.99652

Figure 28. Accuracy Scores for U2R L-Diversity

Figure 29 shows the improved precision scores for RF and DT techniques while for SVM it remains the same and decreases slightly for KNN, and the observed change can be less than 0.01% or 0.02%.



	RF	KNN	SVM	DT
🟦 Before	0.94916	0.93282	0.91988	0.86841
🟧 After	0.97178	0.91444	0.91856	0.88166

Figure 29. Precision Scores for U2R L-Diversity

We have seen that the accuracies are similar to the improved data over the k-anonymization technique. L-Diversity has its weaknesses which are addressed in the last phase of our framework.

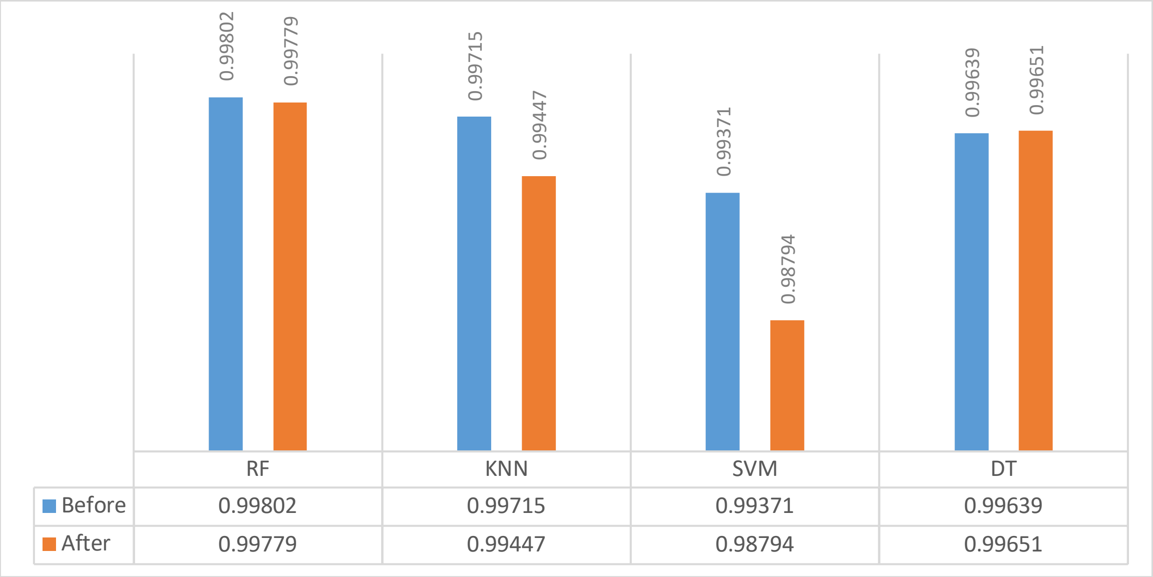
4.2.4 T-Closeness Generalization

In this section, we present our findings on our framework's final phase, namely, the t-closeness method of preserving privacy; t-closeness is an improvement over the l-Diversity Method of Anonymizing Data.

The final phase of our framework has the input data accrued from l-diverse data, t-closeness evaluation mirrors the performance metrics of the previous phase, as this is generalized data for the machine learning models to learn.

In Figures 30 to 37, we see the average accuracy and precision obtained by the proposed model considering different attacks and different machine learning models (i.e., RF, KNN, SVM, and DT). The results reveal that our framework is resilient to the modifications applied to data using t-closeness generalization. The results also show that the percentage of accuracy preserved is 99.7% for the adjusted data. By using our model, the accuracy decrease by the T-Closeness Generalization is as small as 0.03%. These results suggest that the consulted IDSs detect intrusions without the need to use the original data. We can obtain the same results (i.e., no significant degradation has been recorded) using the newly generated data, which is semantically similar to the original data but not synthetically.

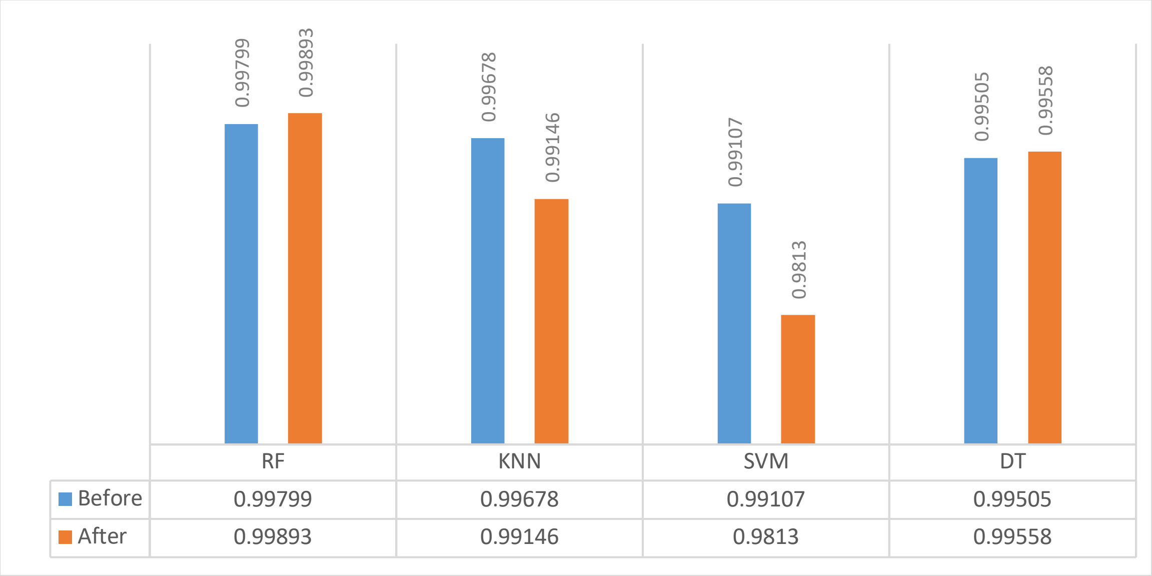
As shown in Figure 30, the accuracy scores for the t-closeness data used in the DoS attack class show a negligible change in the various machine learning algorithms. The change we observe is as low as 0.01%and 0.02%.



	RF	KNN	SVM	DT
🟦 Before	0.99802	0.99715	0.99371	0.99639
🟧 After	0.99779	0.99447	0.98794	0.99651

Figure 30. Accuracy Scores for DoS T-Closeness

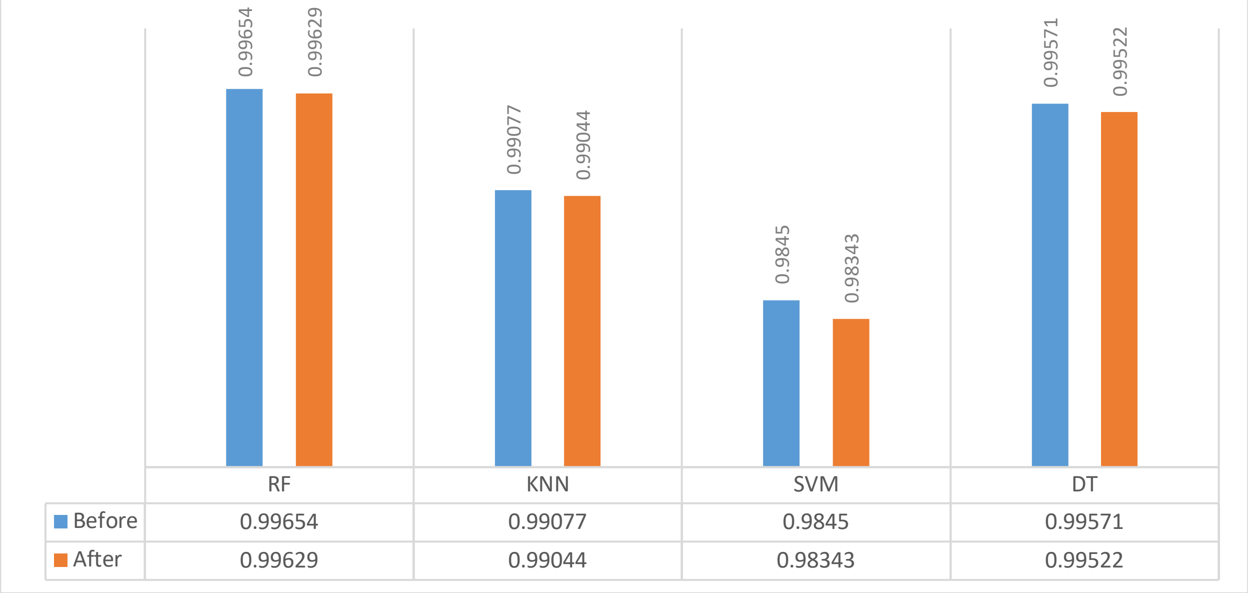
As we observe in Figure 30. Accuracy Scores for DoS T-Closeness , the precision score is slightly increasing for RF and DT techniques while it decreases negligibly for the KNN and SVM techniques.



	RF	KNN	SVM	DT
🟦 Before	0.99799	0.99678	0.99107	0.99505
🟧 After	0.99893	0.99146	0.9813	0.99558

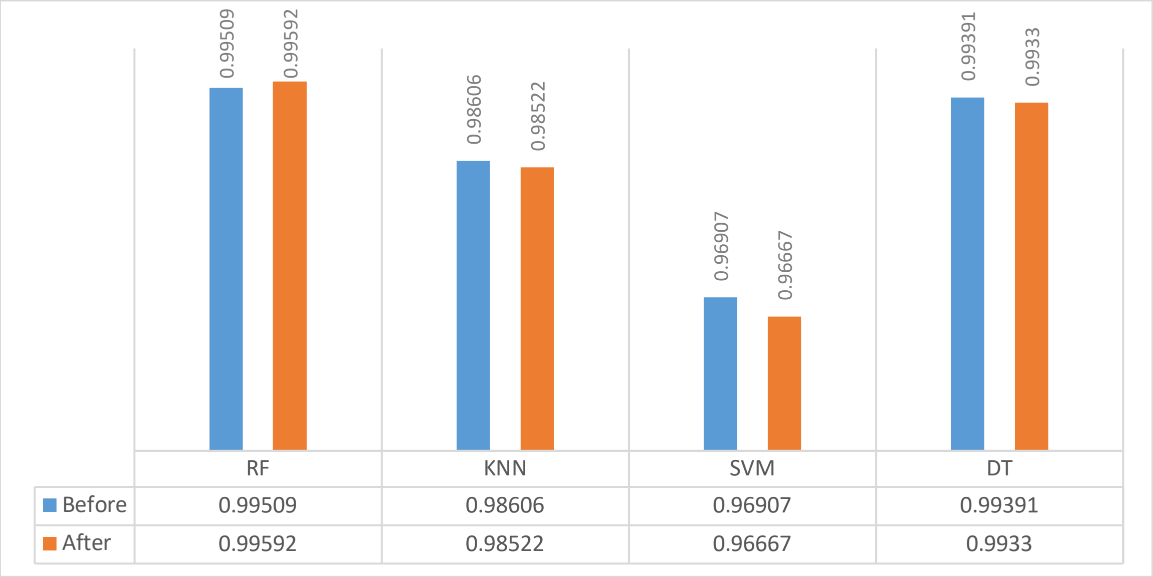
Figure 31. Precision Scores for DoS T-Closeness

As shown in Figures 32 and 33, the accuracy scores for the t-closeness data used in the Probe Attack class show a negligible change in the various machine learning algorithms. The change, as we have seen, is less than 0.01% and 0.02%. The precision score for RF shows a slight improvement while showing a negligible decrease in performance for KNN, SVM, and DT.



	RF	KNN	SVM	DT
🟦 Before	0.99654	0.99077	0.9845	0.99571
🟧 After	0.99629	0.99044	0.98343	0.99522

Figure 32. Accuracy Scores for Probe T-Closeness

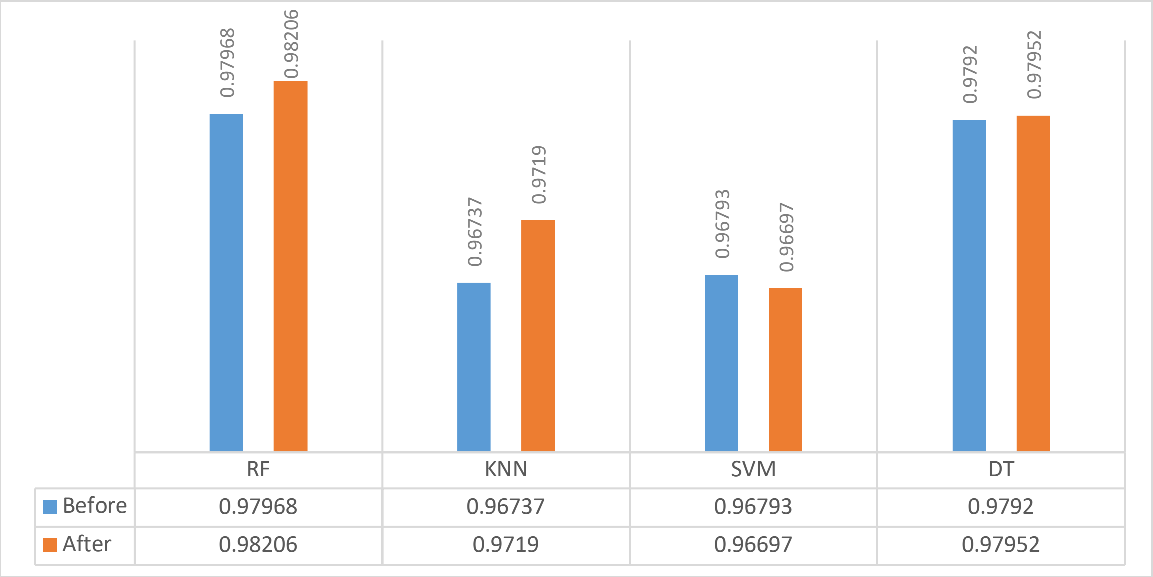


	RF	KNN	SVM	DT
🟦 Before	0.99509	0.98606	0.96907	0.99391
🟧 After	0.99592	0.98522	0.96667	0.9933

Figure 33. Precision Scores for Probe T-Closeness

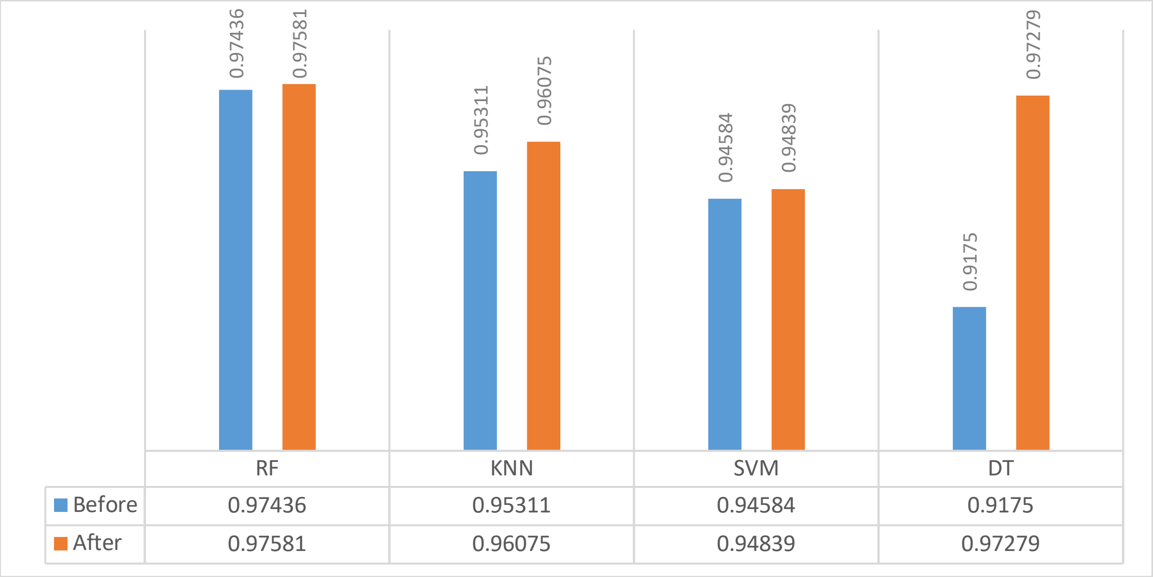
As shown in Figure 33, the precision scores for the t-closeness data used in the R2L attack class, the various machine learning algorithms show a slight improvement, except for the SVM, where there is a negligible decrease in accuracy scores.

In Figure 34, the accuracy improvement we see is as low as 0.01% and 0.02%. The precision score in Figure 35 also shows a slight increase by 0.06% for DT and RF, KNN, SVM by 0.01%.



	RF	KNN	SVM	DT
🟦 Before	0.97968	0.96737	0.96793	0.9792
🟧 After	0.98206	0.9719	0.96697	0.97952

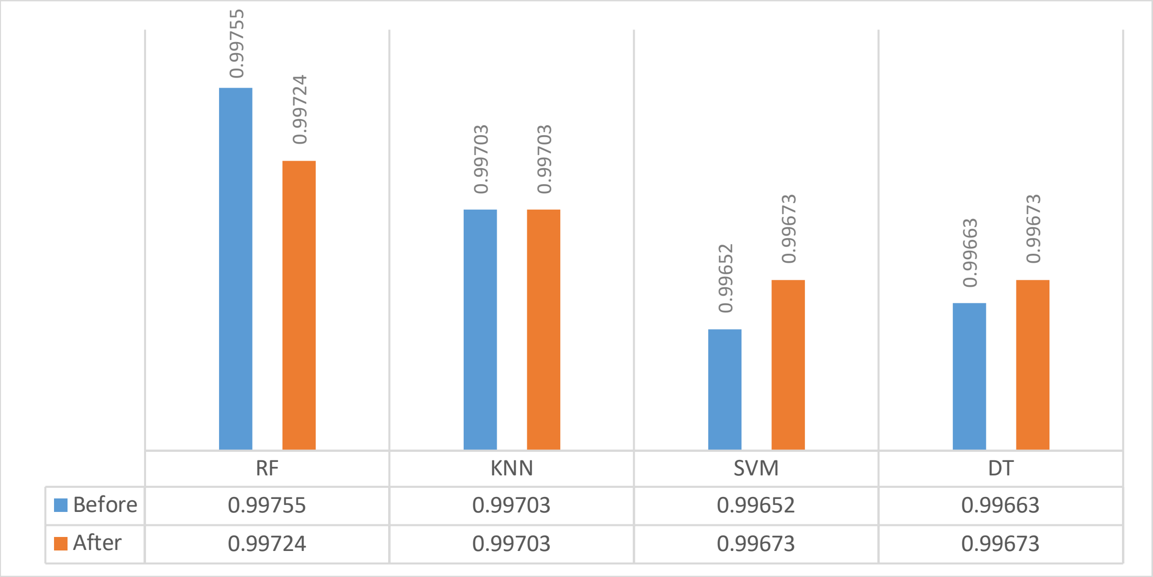
Figure 34. Accuracy Scores for R2L T-Closeness



	RF	KNN	SVM	DT
🟦 Before	0.97436	0.95311	0.94584	0.9175
🟧 After	0.97581	0.96075	0.94839	0.97279

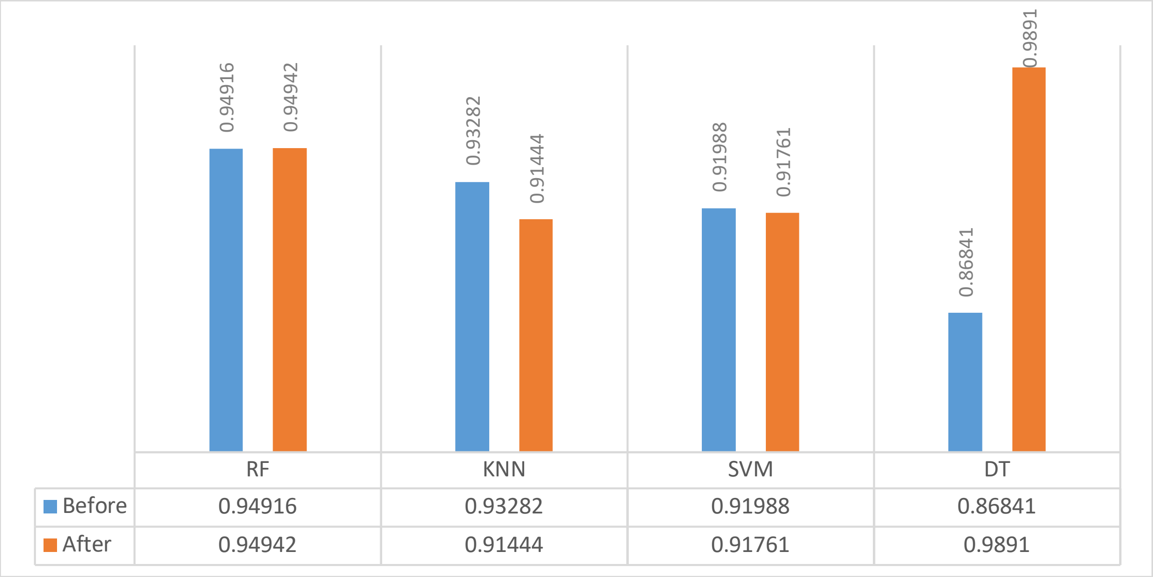
Figure 35. Precision Scores for R2L T-Closeness

As shown in Figures 36 and 37, the accuracy scores for the t-closeness data used in the U2R attack class, KNN, SVM, and DT algorithms show a slight improvement of 0.01%, and for RF it does not decrease significantly. The deterioration we see is as low as 0.01% and 0.02% respectively. The precision score in Figure 36 shows a slight improvement for RF by 0.01 per cent and DT by 0.12 per cent, while it decreases negligibly for SVM and KNN techniques.



	RF	KNN	SVM	DT
🟦 Before	0.99755	0.99703	0.99652	0.99663
🟧 After	0.99724	0.99703	0.99673	0.99673

Figure 36. Accuracy Scores for U2R T-Closeness



	RF	KNN	SVM	DT
🟦 Before	0.94916	0.93282	0.91988	0.86841
🟧 After	0.94942	0.91444	0.91761	0.9891

Figure 37. Precision Scores for U2R T-Closeness

Finally, we note that the accuracy and precision values achieved are the same for all four phases of our framework. We have a reasonable way of publishing to guarantee that the published data are anonymous and that the outsider/attacker's potential to re-identify is minimal.

Chapter 5 – Discussion

In this chapter, we first recall the research objectives that we mentioned at the beginning of the thesis and then illustrate to what extent these objectives were achieved.

5.1 Objectives Achievement

This thesis' main objective was to enable data sharing between members of the Intrusion Detection Systems (IDS) in a cooperative cloud network while preserving data privacy.

When coupled with machine learning techniques, our research and work helped detect cyber-attacks in the cloud in a dynamic and heterogeneous environment while preserving the privacy of the data shared and helped reduce the trade-off between privacy and accuracy. More specifically, we were able to achieve the two following objectives.

5.1.1 Building privacy preserving cloud-based intrusion detection system

For the different types of data collected, there are several privacy-preserving techniques available. Figure 2 showed the paradigm we have followed in such that each of the techniques is complemented by the one that follows it and helps resolve the limitations of the previous one.

Therefore, we prove that we have a steady-state framework, where each method is protected by the other, which complements each method in its functionality. Metrics of Machine learning, such as accuracy and precision scores, perform to check the reliability of each phase of the anonymization process. Like we have pursued, it remains essentially the same as presented in Chapter 4.

5.1.2 Maintaining efficiency of detection

As explained in Chapter 3, we posed the problem of whether the information is private or confidential in network traffic data. We also clarified and illustrated the ARX anonymization method and the principle of alpha-distinction and beta-separation between quasi-identifiers. In addition, we have expertly chosen the best combination of attributes with the best distinction and separation ratios from the results of the ARX anonymization tool.

Besides, the experiments were conducted on different machine learning techniques over different attack classes. And we verified that the accuracy of detection before and after the data modification has not degraded heavily, which allows us to share the semantically generated data.

5.2 Further Applications

The proposed structure is not limited in its approach, and it can be extended to other types of data, such as telemetry and analytics data. In the case of non-uniform data, we leave it to the network administrator to publish the collected data at his discretion, slightly or severely perturbing the data depending on its distribution. In the case of uniform data, the data perturbation parameters can be designed to be set by the algorithm, and the Laplacian noise generation is done spontaneously without the involvement of the cloud administrator.

Chapter 6 – Conclusion and Future Work

In this thesis, we proposed a new framework for achieving privacy in cooperative cloud-based IDS. We listed the challenges faced and laid out the specific problems we intended to address.

We undertook a literature review to ensure our framework's originality and to ensure to fill the void in the state-of-the-art research. We then analyzed the pursued solutions and found that they are used in case-specific and are not applicable at any point in a cloud-based IDS. To address this weakness and apply it in general, we designed a framework that allows us to integrate privacy-preserving methods into machine learning-based IDSs to achieve a privacy-aware cooperative IDS.

We have also devised a new algorithm that allows the IDS to decide which dataset is shared to improve detection accuracy while preserving users' privacy in the IDS region. The proposed framework allows the IDS to hide private and sensitive information in shared data while improving or maintaining its metrics detection.

We finally concluded the thesis by proving the designed framework on the NSL-KDD dataset. In the future, we plan to design an automated privacy-preserving cooperative IDS. More specifically, the IDS will automatically analyze and recognize the data and select the best privacy-preserving and machine-learning techniques to ensure optimal detection accuracy while preserving data privacy. We aim to make the system perform all the tasks spontaneously.

Bibliography

Abusitta, A., Bellaiche, M., Dagenais, M., & Halabi, T. (2019). A deep learning approach for proactive multi-cloud cooperative intrusion detection system. *Future Generation Computer Systems, 98*, 308-318.

Aggarwal, C. C., & Philip, S. Y. (2008). *Privacy-preserving data mining: models and algorithms*: Springer Science & Business Media.

Aggarwal, P., & Sharma, S. K. (2015). Analysis of KDD dataset attributes-class wise for intrusion detection. *Procedia Computer Science, 57*, 842-851.

Al-Rubaie, M., & Chang, J. M. (2019). Privacy-Preserving Machine Learning: Threats and Solutions. *IEEE Security & Privacy, 17*(2), 49-58.

Bakshi, A., & Dujodwala, Y. B. (2010). *Securing cloud from ddos attacks using intrusion detection system in virtual machine.* Paper presented at the 2010 Second International Conference on Communication Software and Networks.

Barbara, D., Couto, J., Jajodia, S., Popyack, L., & Wu, N. (2001). *ADAM: Detecting intrusions by data mining.* Paper presented at the In Proceedings of the IEEE Workshop on Information Assurance and Security.

Bayardo, R. J., & Agrawal, R. (2005). *Data privacy through optimal k-anonymization.* Paper presented at the 21st International conference on data engineering (ICDE'05).

Bhuyan, M. H., Kashyap, H. J., Bhattacharyya, D. K., & Kalita, J. K. (2014). Detecting distributed denial of service attacks: methods, tools and future directions. *The Computer Journal, 57*(4), 537-556.

Coull, S. E., Monrose, F., Reiter, M. K., & Bailey, M. (2009). *The challenges of effectively anonymizing network data.* Paper presented at the 2009 Cybersecurity Applications & Technology Conference for Homeland Security.

Dalenius, T., & Reiss, S. P. (1982). Data-swapping: A technique for disclosure control. *Journal of statistical planning and inference, 6*(1), 73-85.

Dara, S., & Muralidhara, V. (2016). Privacy preserving architectures for collaborative intrusion detection. *arXiv preprint arXiv:1602.02452*.

Dhanabal, L., & Shantharajah, S. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering, 4*(6), 446-452.

Domingo-Ferrer, J. (2018). k-Anonymity. In L. Liu & M. T. Özsu (Eds.), *Encyclopedia of Database Systems* (pp. 2053-2054). New York, NY: Springer New York.

Dwork, C. (2011). Differential privacy. *Encyclopedia of Cryptography and Security*, 338-340.

Erlingsson, Ú., Pihur, V., & Korolova, A. (2014). *Rappor: Randomized aggregatable privacy-preserving ordinal response.* Paper presented at the Proceedings of the 2014 ACM SIGSAC conference on computer and communications security.

Friedman, A., & Schuster, A. (2010). *Data mining with differential privacy.* Paper presented at the Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.

Fung, C. J., Lam, D. Y., & Boutaba, R. (2014). *Revmatch: An efficient and robust decision model for collaborative malware detection.* Paper presented at the 2014 IEEE Network Operations and Management Symposium (NOMS).

Fung, C. J., & Zhu, Q. (2016). FACID: A trust-based collaborative decision framework for intrusion detection networks. *Ad Hoc Networks, 53*, 17-31.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security, 28*(1-2), 18-28.

Gupta, S., & Kumar, P. (2013). *Vm profile based optimized network attack pattern detection scheme for ddos attacks in cloud.* Paper presented at the International Symposium on Security in Computing and Communication.

Herrington, L., & Aldrich, R. (2013). The future of cyber-resilience in an age of global complexity. *Politics, 33*(4), 299-310.

Hesamifard, E., Takabi, H., & Ghasemi, M. (2017). Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*.

Hromkovič, J., & Schnitger, G. (2003). *Nondeterminism versus determinism for two-way finite automata: generalizations of Sipser's separation.* Paper presented at the International Colloquium on Automata, Languages, and Programming.

Hudis, E., Helman, Y., Malka, J., & Barash, U. (2013). Adaptive data collection for root-cause analysis and intrusion detection. In: Google Patents.

Kalapanidas, E., Avouris, N., Craciun, M., & Neagu, D. *Machine learning algorithms: a study on noise sensitivity*.

Kargupta, H., Datta, S., Wang, Q., & Sivakumar, K. (2003). *On the Privacy Preserving Properties of Random Data Perturbation Techniques.* Paper presented at the ICDM.

Kayacik, H. G., Zincir-Heywood, A. N., & Heywood, M. I. (2005). *Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets.* Paper presented at the Proceedings of the third annual conference on privacy, security and trust.

Kodratoff, Y., Manago, M., & Blythe, J. (1987). Generalization and noise. *International Journal of Man-Machine Studies, 27*(2), 181-204.

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2005). *Incognito: Efficient full-domain k-anonymity.* Paper presented at the Proceedings of the 2005 ACM SIGMOD international conference on Management of data.

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2006). *Mondrian multidimensional k-anonymity.* Paper presented at the ICDE.

Li, N., Li, T., & Venkatasubramanian, S. (2007). *t-closeness: Privacy beyond k-anonymity and l-diversity.* Paper presented at the 2007 IEEE 23rd International Conference on Data Engineering.

Li, T., & Li, N. (2009). *On the tradeoff between privacy and utility in data publishing.* Paper presented at the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.

Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., & Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36*(1), 16-24.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news, 2*(3), 18-22.

Lin, Z., Shi, Y., & Xue, Z. (2018). Idsgan: Generative adversarial networks for attack generation against intrusion detection. *arXiv preprint arXiv:1809.02077*.

Lo, C.-C., Huang, C.-C., & Ku, J. (2010). *A cooperative intrusion detection system framework for cloud computing networks.* Paper presented at the 2010 39th International Conference on Parallel Processing Workshops.

Lonea, A. M., Popescu, D. E., & Tianfield, H. (2013). Detecting DDoS attacks in cloud computing environment. *International Journal of Computers Communications & Control, 8*(1), 70-78.

Low, C., Chen, Y., & Wu, M. (2011). Understanding the determinants of cloud computing adoption. *Industrial management & data systems*.

Machanavajjhala, A., Gehrke, J., Kifer, D., & Venkitasubramaniam, M. (2006). *l-diversity: Privacy beyond k-anonymity.* Paper presented at the 22nd International Conference on Data Engineering (ICDE'06).

Massey Jr, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association, 46*(253), 68-78.

Mendes, R., & Vilela, J. P. (2017). Privacy-preserving data mining: methods, metrics, and applications. *IEEE Access, 5*, 10562-10582.

Meng, Y., Li, W., Kwok, L.-F., & Xiang, Y. (2013). *Towards designing privacy-preserving signature-based IDS as a service: a study and practice.* Paper presented at the 2013 5th International Conference on Intelligent Networking and Collaborative Systems.

Meyerson, A., & Williams, R. (2004). *On the complexity of optimal k-anonymity.* Paper presented at the Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.

Michie, D. (1994). Spiegelhalter; DJ & Taylor, CC (1994). *Machine learning, neural and statistical classification*.

Motwani, R., & Xu, Y. (2007). *Efficient algorithms for masking and finding quasi-identifiers.* Paper presented at the Proceedings of the Conference on Very Large Data Bases (VLDB).

Niksefat, S., Sadeghiyan, B., Mohassel, P., & Sadeghian, S. (2014). ZIDS: a privacy-preserving intrusion detection system using secure two-party computation protocols. *The Computer Journal, 57*(4), 494-509.

Oliveira, T., Thomas, M., & Espadanal, M. (2014). Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors. *Information & Management, 51*(5), 497-510.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research, 12*(Oct), 2825-2830.

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia, 4*(2), 1883.

Prasser, F., & Kohlmayer, F. (2015). Putting statistical disclosure control into practice: The ARX data anonymization tool. In *Medical Data Privacy Handbook* (pp. 111-148): Springer.

Rastogi, V., Suciu, D., & Hong, S. (2007). *The boundary between privacy and utility in data publishing.* Paper presented at the Proceedings of the 33rd international conference on Very large data bases.

Ribeiro, M., Grolinger, K., & Capretz, M. A. (2015). *Mlaas: Machine learning as a service.* Paper presented at the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA).

Rowland, C. H. (2002). Intrusion detection system. In: Google Patents.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics, 21*(3), 660-674.

Sankar, L., Rajagopalan, S. R., & Poor, H. V. (2010). *A theory of utility and privacy of data sources.* Paper presented at the 2010 IEEE International Symposium on Information Theory.

Shokri, R., & Shmatikov, V. (2015). *Privacy-preserving deep learning.* Paper presented at the Proceedings of the 22nd ACM SIGSAC conference on computer and communications security.

Smith, A. (2012). *Pinning down "privacy" in statistical databases.* Paper presented at the Tutorial in Proceedings of the 32nd International Cryptology Conference (CRYPTO).

Song, D., Shi, E., Fischer, I., & Shankar, U. (2012). Cloud data protection for the masses. *Computer, 45*(1), 39-45.

Spathoulas, G. P., & Katsikas, S. K. (2010). Reducing false positives in intrusion detection systems. *computers & security, 29*(1), 35-44.

Su, M.-Y. (2011). Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification. *Journal of Network and Computer Applications, 34*(2), 722-730.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters, 9*(3), 293-300.

Sweeney, L. (2002a). Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10*(05), 571-588.

Sweeney, L. (2002b). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10*(05), 557-570.

Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., & Lin, W.-Y. (2009). Intrusion detection by machine learning: A review. *Expert systems with Applications, 36*(10), 11994-12000.

Wong, R. C.-W., Li, J., Fu, A. W.-C., & Wang, K. (2006). *(α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing.* Paper presented at the Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.

Zargar, S. T., Takabi, H., & Joshi, J. B. (2011). *DCDIDP: A distributed, collaborative, and data-driven intrusion detection and prevention framework for cloud computing environments.* Paper presented at the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom).

Zhang, H., Shu, Y., Cheng, P., & Chen, J. (2016). Privacy and performance trade-off in cyber-physical systems. *IEEE Network, 30*(2), 62-66.

Zhang, T., & Zhu, Q. (2018). Distributed privacy-preserving collaborative intrusion detection systems for VANETs. *IEEE Transactions on Signal and Information Processing over Networks, 4*(1), 148-161.

Zhang, X., Liu, C., Nepal, S., & Chen, J. (2013). An efficient quasi-identifier index based approach for privacy preservation over incremental data sets on cloud. *Journal of Computer and System Sciences, 79*(5), 542-555.

Zhou, C. V., Leckie, C., & Karunasekera, S. (2010). A survey of coordinated attacks and collaborative intrusion detection. *computers & security, 29*(1), 124-140.