Anomaly Detection in Participatory Sensing Networks

Sarah Monazam Erfani

Submitted in total fulfilment of the requirements of the degree of
Doctor of Philosophy

Department of Computing and Information Systems
THE UNIVERSITY OF MELBOURNE

August 2015
Abstract

Anomaly detection or outlier detection aims to identify unusual values in a given dataset. In particular, there is growing interest in collaborative anomaly detection, where multiple data sources submit their data to an online data mining service, in order to detect anomalies with respect to the wider population. By combining data from multiple sources, collaborative anomaly detection aims to improve detection accuracy through the construction of a more robust model of normal behaviour. Cloud-based collaborative architectures such as Participatory Sensing Networks (PSNs) provide an open distributed platform that enables participants to share and analyse their local data on a large scale. Two major issues with collaborative anomaly detection are how to ensure the privacy of participants’ data, and how to efficiently analyse the large-scale high-dimensional data collected in these networks.

The first problem we address is the issue of data privacy in PSNs, by introducing a framework for privacy-preserving collaborative anomaly detection with efficient local data perturbation at participating nodes, and global processing of the perturbed records at a data mining server. The data perturbation scheme that we propose enables the participants to perturb their data independently without requiring the cooperation of other parties. As a result our privacy-preservation approach is scalable to large numbers of participants and is computationally efficient.

By collecting the participants’ data, the PSN server can generate a global anomaly detection model from these locally perturbed records. The global model identifies interesting measurements or unusual patterns in participants’ data without revealing the true values of the measurements. In terms of privacy, the proposed scheme thwarts several major types of attacks, namely, the Independent Component Analysis (ICA), Distance-
inference, Maximum a Posteriori (MAP), and Collusion attacks.

We further improve the privacy of our data perturbation scheme by: (i) redesigning the nonlinear transformation to better defend against MAP estimation attacks for normal and anomalous records, and (ii) supporting individual random linear transformations for each participant in order to provide the system with greater resistance to malicious collusion. A notable advantage of our perturbation scheme is that it preserves participants’ privacy while achieving comparable accuracy to non-privacy preserving anomaly detection techniques.

The second problem we address in the thesis is how to model and interpret the large volumes of high-dimensional data that are generated in participatory domains by using One-class Support Vector Machines (1SVMs). While 1SVMs are effective at producing decision surfaces for anomaly detection from well-behaved feature vectors, they can be inefficient at modelling the variations in large, high-dimensional datasets. We overcome this challenge by taking two different approaches. The first approach is an unsupervised hybrid architecture, in which a Deep Belief Network (DBN) is used to extract generic underlying features, in combination with a 1SVM that uses the features learned by the DBN. DBNs have important advantages as feature detectors for anomaly detection, as DBNs use unlabelled data to capture higher-order correlations among features. Furthermore, using a DBN to reduce the number of irrelevant and redundant features improves the scalability of a 1SVM for use with large training datasets containing high-dimensional records. Our hybrid approach is able to generate an accurate anomaly detection model with lower computational and memory complexity compared to a 1SVM on its own.

Alternatively, to overcome the shortcomings of 1SVMs in processing high-dimensional datasets, in our second approach we calculate a lower rank approximation of the optimisation problem that underlies the 1SVM training task. Instead of performing the optimisation in a high-dimensional space, the optimisation is conducted in a space of reduced dimension but on a larger neighbourhood. We leverage the theory of nonlinear random projections and propose the Reduced 1SVM (R1SVM), which is an efficient and scalable anomaly detection technique that can be trained on large-scale datasets. The main objective of R1SVM is to replace a nonlinear machine with randomised features and a linear
machine.

In summary, we have proposed efficient privacy-preserving anomaly detection approaches for PSNs, and scalable data modelling approaches for high-dimensional datasets, which lower the computational and memory complexity compared to traditional anomaly detection techniques. We have shown that the proposed methods achieve higher or comparable accuracy in detecting anomalies compared to existing state-of-art techniques.
Declaration

This is to certify that

1. the thesis comprises only my original work towards the PhD,

2. due acknowledgement has been made in the text to all other material used,

3. the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Sarah Monazam Erfani, August 2015
I would like to express my sincere gratitude to my supervisors Associate Professor Shanika Karunasekera and Professor Christopher Leckie. I greatly appreciate their continuous support, patience, enthusiasm, and immense knowledge through my Ph.D study and research. They diligently guided me in all the aspects of my research and writing of this thesis. They have been great mentors and helped me grow in several ways.

My sincere thanks goes to Professor James Bezdek, for his insightful advice and encouragement. I would like to thank my Ph.D advisory Associate Professor Lars Kulik for his sound comments. I also would like to thank Dr. Sutharshan Rajasegarar, Dr. Yee Wei Law, and Dr. Mahsa Baktashmotlagh for their technical advice, and contribution in this thesis. I would like to thank Professor Marimuthu Palaniswami and Dr. Udaya Parampalli for sharing their insights with me. I also thank my fellow students at The University of Melbourne who have touched my life in many ways and helped me throughout my study.

I would like to thank my entire family, my parents Tala Tabein and Abbas Erfani for their selfless support and inspiration, and my brother, my father and mother in-law for their encouragement. Lastly, and most importantly, I wish to thank my loving husband, Iman Naghizade. He has been beside me at every step of my journey, believed in me, supported me and loved me. To him I dedicate this thesis.
## Contents

1 Introduction .......................... 1
   1.1 Focus of the Thesis ................. 2
       1.1.1 Background .................. 2
       1.1.2 Privacy-preserving Anomaly Detection ........ 5
       1.1.3 Large-scale High-dimensional Anomaly Detection .... 7
   1.2 Organisation and Contributions of the Thesis ........ 9

2 Background on Privacy-Preserving and Large-Scale Anomaly Detection in PSNs .... 17
   2.1 Basic definitions .................. 17
   2.2 Survey of Privacy-Preserving Data Mining ............ 23
       2.2.1 Adversarial Models ............ 23
       2.2.2 Taxonomy of PPDM Approaches ...... 24
       2.2.3 Summary .................... 29
   2.3 Survey of Large-Scale Anomaly Detection ............ 30
       2.3.1 Background on Anomaly Detection ........ 30
       2.3.2 Anomaly Detection Techniques ...... 31
       2.3.3 Anomaly Detection in High-dimensional Datasets .... 38
       2.3.4 Shallow and Deep Architectures .... 41
       2.3.5 Summary .................... 42

I Privacy-Preserving Anomaly Detection .................................... 43

3 KSSI: Privacy-Preserving Data Aggregation in Participatory Sensing Networks ........ 47
   3.1 Introduction ....................... 47
   3.2 Related Work ...................... 49
   3.3 Preliminaries and Problem Statement ............... 50
       3.3.1 Participatory Sensing Architecture .......... 50
       3.3.2 Aggregation Query Model .................. 52
       3.3.3 Security Model .................... 52
   3.4 Our Schemes for Privacy Preservation ............... 53
       3.4.1 Basic Secret Perturbation Scheme (BSP) ....... 53
       3.4.2 Universal Participation Scheme (UPS) ....... 55
       3.4.3 Key Splitting Scheme (KSS) ................ 56
       3.4.4 Key Splitting Scheme with Integrity (KSSI) .... 57
| 3.5 | Performance Analysis | 58 |
| 3.5.1 | Hidden Probability | 59 |
| 3.5.2 | Communication Overhead | 60 |
| 3.6 | Evaluation Results | 62 |
| 3.6.1 | Hidden Probability | 63 |
| 3.6.2 | Communication Overhead | 64 |
| 3.7 | Discussion and Conclusion | 65 |

| 4 | RMP: Privacy-Preserving Collaborative Anomaly Detection for Participatory Sensing | 67 |
| 4.1 | Introduction | 67 |
| 4.2 | RMP Privacy-Preserving Anomaly Detection Scheme | 71 |
| 4.2.1 | Problem Statement and Design Principles | 71 |
| 4.2.2 | Basic RMP | 73 |
| 4.2.3 | Mediated RMP | 76 |
| 4.3 | Experimental Analysis | 77 |
| 4.4 | Conclusion | 80 |

| 5 | ERMP: Enhanced Random Multiparty Perturbation | 81 |
| 5.1 | Introduction | 81 |
| 5.2 | Enhanced RMP (ERMP) | 83 |
| 5.3 | Privacy Analysis | 83 |
| 5.3.1 | Attack Based on Bayesian Estimation | 86 |
| 5.4 | Experimental Results on Accuracy | 91 |
| 5.4.1 | Impact of Two-Stage Transformations | 92 |
| 5.4.2 | Impact of Imposed Noise | 95 |
| 5.5 | Experimental Results on Privacy | 95 |
| 5.6 | Conclusion | 99 |

| 6 | IRMP: Individually Random Multiparty Perturbation | 101 |
| 6.1 | Introduction | 101 |
| 6.2 | IRMP Privacy-Preserving Anomaly Detection Scheme | 104 |
| 6.2.1 | Privacy-Preserving Random Perturbation | 104 |
| 6.2.2 | Privacy-preserving Anomaly Detection with CAE | 105 |
| 6.3 | Simulations and Comparison | 106 |
| 6.3.1 | Accuracy Evaluation | 108 |
| 6.3.2 | Scalability Evaluation | 112 |
| 6.3.3 | Efficiency Evaluation | 114 |
| 6.4 | Discussion and Conclusions | 115 |

| II | High-Dimensional and Large-Scale Anomaly Detection | 117 |

| 7 | High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning | 121 |
| 7.1 | Introduction | 121 |
7.2 Background DBN and Hybrid DBN-SVM ........................................ 125
7.3 DBN-SVM Hybrid Model ............................................................. 126
  7.3.1 Deep Belief Nets (DBNs) ......................................................... 127
  7.3.2 One-class SVM ................................................................. 129
7.4 Evaluation and Discussion ............................................................ 132
  7.4.1 Experimental Methodology ................................................... 132
  7.4.2 Choice of Kernel ............................................................... 134
  7.4.3 Influence of Dimensionality Reduction .................................... 136
  7.4.4 Influence of Number of Hidden Layers ................................. 139
  7.4.5 Influence of Number of Training Records .............................. 142
  7.4.6 Complexity ................................................................. 143
7.5 Conclusion ................................................................................. 144

8 RISVM: a Randomised Nonlinear Approach to Large-Scale Anomaly Detection 147
  8.1 Introduction ................................................................................ 147
  8.2 Related Work and Background .................................................. 149
  8.3 Proposed Approach — RISVM .................................................... 151
    8.3.1 One-class SVM (1SVM) ....................................................... 151
    8.3.2 Randomised 1SVM ............................................................. 154
  8.4 Evaluation and Discussion .......................................................... 155
    8.4.1 Experiment 1: Interpretation of Basis for Anomaly Detection with RISVM ......................................................... 156
    8.4.2 Experiment 2: Empirical Performance of RISVM ................. 158
  8.5 Conclusion and Research Directions .......................................... 161

9 Conclusion and Future Work .............................................................. 163
  9.1 Summary of Contributions ......................................................... 163
  9.2 Future Research ........................................................................ 167
List of Figures

1.1 An architecture of participatory sensing networks. ................................. 4
1.2 Organisation of thesis. .............................................................................. 9

2.1 Data sharing scenarios necessitating privacy preservation. ...................... 19
2.2 Comparing various I SVM methods. ........................................................... 32
2.3 Stacked autoencoder. ................................................................................ 36

3.1 Architectures of PSN and aggregation schemes. ....................................... 51
3.2 Hidden probability for a given number of cover nodes $c$. ......................... 62
3.3 Communication cost under the default settings. ....................................... 63
3.4 Communication cost with $c = 4$. ............................................................. 64
3.5 Comparison of communication cost for KSSI. ......................................... 64

4.1 The RMP architecture. ............................................................................. 71
4.2 Impact of noise on various datasets. ......................................................... 79

5.1 Plot of the tanh, the double logistic and the poly7 functions. .................. 88
5.2 Distributions of $y = \mathcal{N}(x)$ (note here that in this illustration $x$ is Gaussian distributed with zero mean and covariance matrix diag[1, 1]): (a) $\mathcal{N}$ is the identity function; (b) $\mathcal{N}$ is tanh. (c) $\mathcal{N}$ is the double logistic function; (d) $\mathcal{N}$ is poly7—note the PDF is uniformly close to zero but not zero. ....... 89
5.3 Distributions of $y = \mathcal{N}(x)$ (note here that in this illustration $x$ is Laplace distributed with zero mean and covariance matrix diag[1, 1]): (a) $\mathcal{N}$ is the identity function; (b) $\mathcal{N}$ is tanh. (c) $\mathcal{N}$ is the double logistic function; (d) $\mathcal{N}$ is poly7—note the PDF is uniformly close to zero but not zero. ....... 89
5.4 Demonstration of the effect of random projection on normal (star) and anomalous (circle) records. ................................................................. 93
5.5 Recovery rates of MAP estimation attacks against the evaluated schemes, on $w \times 1000$ data projected from $15 \times 1000$ normalised Gaussian-distributed data (zero mean, identity covariance matrix). .............................................. 97
5.6 Recovery rates of MAP estimation attacks against the evaluated schemes, on $w \times 1000$ data projected from $15 \times 1000$ normalised Laplace-distributed data (zero mean, unity scale). ....................................................... 97
5.7 0.1-recovery rates of the MAP estimation attack against the evaluated schemes, on various datasets. The rank of the perturbation matrix, \( w \), is set as \( \lfloor (n + 1)/2 \rfloor \), where \( n \) is the number of attributes. Note that the recovery rates are zero in many cases.

6.1 The IRMP architecture.

6.2 Comparing the impact of different transformations on normal and anomalous records.

6.3 Impact of dimensionality reduction in terms of size of \( w \) on AUC.

6.4 Influence of number of participants on AUC.

6.5 Influence of data dimensionality on training and testing time.

6.6 Influence of number of records on training and testing time.

7.1 Model architecture of AE, DBN and the proposed hybrid DBN-ISVM.

7.2 Comparison of rankings of anomaly detection methods for 3 metrics.

7.3 Demonstrating the impact of number of hidden layers on the separability of normal and anomalous records.

7.4 Comparing the accuracy of anomaly detection methods as the number of training records is varied.

7.5 Comparing the performance results of the proposed and studied anomaly detection techniques.

8.1 Demonstration of the effect of nonlinear projection on normal and anomalous records.

8.2 Comparison of accuracy of anomaly detection methods as the number of training records is varied.

8.3 Comparison of the training and testing time of AE and R1SVM on large datasets.
List of Tables

3.1 Default Evaluation Parameters ............................................. 63

4.1 Comparing AUC values of RMP against non-privacy anomaly detection. 78

5.1 Suitable Values for the Coefficients of Poly7 ................................. 88
5.2 Comparing the impact of different nonlinear transformations on the AUC of anomaly detection techniques .................................................. 94
5.3 Comparing the impact of the added noise level $\alpha$ on the AUC of anomaly detection techniques .................................................. 94
5.4 Evaluated schemes ................................................................. 96

6.1 Comparing AUC Values of IRMP Against Non Privacy-Preserving Anomaly Detection .......................................................... 109

7.1 Performance results ............................................................... 135
7.2 Wilcoxon test to compare the performance of the PSVM and SVDD based methods with respect to the choice of kernel. ................................. 136
7.3 Shaffer test to compare AUC values. ............................................. 139
7.4 Shaffer test to compare training times .......................................... 139
7.5 Shaffer test to compare test times ................................................. 140
7.6 Wilcoxon test to compare the performance of DBN-based methods regarding $p$-values of Wilcoxon test .................................................. 140
7.7 Performance results of DBN-based methods with one hidden layer. 141
7.8 Wilcoxon tests to determine the influence of the number of hidden layers on the performance of DBN-based methods ................................................. 141
7.9 Comparing time- and memory-complexities .................................... 143

8.1 Comparison of AUC, train and test times of RISVM with SVDD and autoencoder (AE) ........................................................ 158
8.2 Comparing computational and memory complexity of SVDD, AE and RISVM ........................................................ 161
List of Publications

The following is the list of publications that have arisen from this thesis. The abstract of each chapter indicates the corresponding publications arising from that chapter.

Published Papers

P 1. Sarah M. Erfani, Shanika Karunasekera, Christopher Leckie, Udaya Parampalli, “Privacy-Preserving Data Aggregation in Participatory Sensing Networks”, in Proceedings of the IEEE Eighth International Conference on Intelligent Sensor, Sensor Networks and Information (ISSNIP), 2013 (Chapter 3 contains material from this publication).

P 2. Sarah M. Erfani, Yee Wei Law, Shanika Karunasekera, Christopher Leckie, “Privacy-Preserving Collaborative Anomaly Detection for Participatory Sensing”, in Proceedings of Advances in Knowledge Discovery and Data Mining (PAKDD), 2014 (Chapter 4 contains material from this publication).


Papers under Review

P 4. Sarah M. Erfani, Yee Wei Law, Sutharshan Rajasegarar, Shanika Karunasekera, Christopher Leckie, “Survey of Privacy-Preserving Collaborative Data Mining”, submitted to ACM Computing Surveys (CSUR) (Chapter 2 contains material from this publication).

P 5. Sarah M. Erfani, Yee Wei Law, Shanika Karunasekera, Christopher Leckie, “Efficient and Scalable Privacy-Preserving Collaborative Anomaly Detection for Participatory Sensing”, submitted to IEEE Transactions on Knowledge and Data Engineering (TKDE) (Chapter 5 contains material from this publication).
P 6. Sarah M. Erfani, Yee Wei Law, Shanika Karunasekera, Christopher Leckie, “Privacy-Preserving Collaborative Anomaly Detection using Individually Random Perturbation”, under revision in Machine Learning (Chapter 6 contains material from this publication).

P 7. Sarah M. Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, Christopher Leckie, “One-class Support Vector Machine Based Deep Learning Anomaly Detection”, under revision in Pattern Recognition (Chapter 7 contains material from this publication).
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSN</td>
<td>Participatory Sensing Network</td>
<td>3</td>
</tr>
<tr>
<td>SMC</td>
<td>Secure Multiparty Computation</td>
<td>6</td>
</tr>
<tr>
<td>TTP</td>
<td>Trusted-Third Party</td>
<td>5</td>
</tr>
<tr>
<td>1SVM</td>
<td>One-class Support Vector Machine</td>
<td>8</td>
</tr>
<tr>
<td>HMAC</td>
<td>Hash Message Authentication Code</td>
<td>11</td>
</tr>
<tr>
<td>RMP</td>
<td>Random Multiparty Computation</td>
<td>12</td>
</tr>
<tr>
<td>ERMP</td>
<td>Enhanced Random Multiparty Computation</td>
<td>13</td>
</tr>
<tr>
<td>ICA</td>
<td>Independent Component Analysis</td>
<td>13</td>
</tr>
<tr>
<td>DBNs</td>
<td>Deep Belief Networks</td>
<td>15</td>
</tr>
<tr>
<td>R1SVM</td>
<td>Randomised One-class Support Vector Machine</td>
<td>16</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
<td>17</td>
</tr>
<tr>
<td>PPDM</td>
<td>Privacy-Preserving Data Mining</td>
<td>21</td>
</tr>
<tr>
<td>RP</td>
<td>Random Projection</td>
<td>27</td>
</tr>
<tr>
<td>RT</td>
<td>Uniform Random Transformation</td>
<td>28</td>
</tr>
<tr>
<td>RSVM</td>
<td>Reduced Support Vector Machine</td>
<td>28</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a Posteriori</td>
<td>29</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
<td>32</td>
</tr>
<tr>
<td>PSVM</td>
<td>Hyperplane-Based One-class Support Vector Machine</td>
<td>32</td>
</tr>
<tr>
<td>CESVM</td>
<td>Centered Hyperellipsoidal Support Vector Machine</td>
<td>33</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>QSSVM</td>
<td>Quarter-Sphere Support Vector Machine.</td>
<td>33</td>
</tr>
<tr>
<td>SVDD</td>
<td>Support Vector Data Description</td>
<td>33</td>
</tr>
<tr>
<td>AE</td>
<td>Autoencoder.</td>
<td>36</td>
</tr>
<tr>
<td>CAE</td>
<td>Contractive Autoencoder.</td>
<td>37</td>
</tr>
<tr>
<td>DAE</td>
<td>Denosing Autoencoder.</td>
<td>37</td>
</tr>
<tr>
<td>AODV</td>
<td>Ad hoc On-demand Distance Vector</td>
<td>50</td>
</tr>
<tr>
<td>BSP</td>
<td>Basic Secret Perturbation Scheme</td>
<td>53</td>
</tr>
<tr>
<td>UPS</td>
<td>Universal Perturbation Scheme</td>
<td>53</td>
</tr>
<tr>
<td>KSS</td>
<td>Key Splitting Scheme</td>
<td>58</td>
</tr>
<tr>
<td>KSSI</td>
<td>Key Splitting Scheme with Integrity</td>
<td>58</td>
</tr>
<tr>
<td>CN</td>
<td>Cover Node</td>
<td>54</td>
</tr>
<tr>
<td>MAC</td>
<td>Message Authentication Code.</td>
<td>58</td>
</tr>
<tr>
<td>pdf</td>
<td>Probability Density Function.</td>
<td>70</td>
</tr>
<tr>
<td>DL</td>
<td>Double Logistic</td>
<td>75</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
<td>78</td>
</tr>
<tr>
<td>AUC</td>
<td>Area Under the ROC Curve</td>
<td>78</td>
</tr>
<tr>
<td>RPCA</td>
<td>Randomised Principal Component Analysis</td>
<td>148</td>
</tr>
<tr>
<td>RCCA</td>
<td>Randomised Canonical Correlation Analysis</td>
<td>148</td>
</tr>
<tr>
<td>QP</td>
<td>Quadratic Programming</td>
<td>150</td>
</tr>
<tr>
<td>KKT</td>
<td>Karush-Kuhn-Tucker</td>
<td>158</td>
</tr>
<tr>
<td>STM</td>
<td>Support Tensor Machine</td>
<td>168</td>
</tr>
</tbody>
</table>
Symbols and Notation

For the convenience of the reader, this section describes the symbols and notation used throughout the thesis.

**Matrices and vectors:** Matrices are denoted by bold, upright capital letters, e.g., \( \mathbf{X} \), and vectors by bold, italic lower-case letters, e.g., \( \mathbf{x} \). Matrix elements are denoted by non-bold, italic lower-case letters, e.g., \( x_{ij} \) denotes the element of \( \mathbf{X} \) indexed by \( i \) and \( j \). \( \mathbf{X} \in \mathbb{R}^{n \times m} \) means \( \mathbf{X} \) is a real \( n \times m \) matrix. The augmented matrix \([\mathbf{X}_1|\mathbf{X}_2]\) is formed by vertically concatenating \( \mathbf{X}_1 \) and \( \mathbf{X}_2 \). \( \|\mathbf{x}\|_p \) (\( p \in \mathbb{N} \)) denotes the \( p \)-norm of vector \( \mathbf{x} \), whereas \( \|\mathbf{X}\|_F \) denotes the Frobenius norm of matrix \( \mathbf{X} \). The transpose of a vector \( \mathbf{x} \) or a matrix \( \mathbf{X} \) is denoted as \( \mathbf{x}^T \) or \( \mathbf{X}^T \). \( J_f(\mathbf{x}) \) denotes the Jacobian matrix of (mapping) function \( f \) calculated at \( \mathbf{x} \).

**Probability and statistics:** \( p_X(x) \) denotes the probability density function (pdf) of \( x \), where \( x \sim X \), but when the context is clear, we write \( p_X(x) \) as just \( p(x) \) for brevity. Given two random variables \( x \) and \( y \), their joint probability is denoted as \( p(x,y) \), and the conditional probability of \( x \) given \( y \) is \( p(x|y) \). To denote a scalar-valued random variable, we use a capital letter in the normal font, e.g., \( X \). To denote a vector-valued or matrix-valued random variable, we use a capital letter in the euscript font, e.g., \( \mathcal{X} \). \( U(a,b) \) denotes a scalar random variable that is uniformly distributed between \( a \) and \( b \), whereas \( U_{w \times n}(a,b) \) denotes a \( w \)-by-\( n \) matrix-valued random variable whose elements are uniformly distributed between \( a \) and \( b \).
Chapter 1
Introduction

Modern sensor network technology makes it possible to collect, store and analyse measurements in a wide variety of monitoring applications. Traditionally, this has required dedicated wireless sensor nodes to be deployed in the environment to be monitored. For example, by deploying motion and noise sensors in an urban environment, we can monitor pedestrian movements and their impact on the amenity of the area in terms of noise levels [121][128][142]. Deployments of sensors in this way have motivated substantial investments in “smart city” applications, in which the interactions between the activities of citizens and urban infrastructure can be further understood [15][93][126].

As sensor technology has further advanced and become relatively inexpensive, it has become possible to integrate a diverse range of sensors into modern smartphones. For example, Apple’s Health application allows individuals to track their training and see their overall health status [10]. Apple’s HealthKit service allows third-party health-tracking devices and applications to share their data with Apple’s Health application and access the collected data [11]. This means it is possible to monitor the health and activities of individual users, as well as make observations of their immediate environment. Rather than relying on fixed sensors to provide observations of groups of people within range of the sensor, we can collect much finer grained information about individual users.

The introduction of sensors on smartphones has been a major factor in the emergence of participatory sensing, in which users contribute their own personal sensor data to an online service [30][62]. The online service can then apply various types of analysis to the contributed sensor data in order to support the users of the online service. For example, health specialists can then use the results of the online service to better understand the
development and effective treatment of disease. Some real-world examples of participatory health applications can be found in [64, 67].

In this thesis, we focus on two major theoretical challenges that arise in this context. First is the issue of **privacy**. It is likely that users will be reluctant to contribute their personal data to a participatory sensing service if there is a risk that their data can be misused by untrusted third-parties. Second is the issue of **data analytics on high-dimensional data**. As the variety and sophistication of sensors increases, users will be able to contribute much higher dimensional datasets for analysis. This raises challenges in terms of the accuracy and computational efficiency of data analytics services that may be used to perform more complex types of analysis on high-dimensional sensor observations.

In this thesis, we address both of these challenges for data analysis on sensor data, and consider the effect of solutions for privacy on the accuracy and efficiency of certain classes of data analytics. We begin in Section 1.1 by providing a background on participatory sensing, its motivation and its challenges. Next, we define the scope of the problem we are addressing in this thesis and elaborate on the shortcomings of existing approaches. Finally, in Section 1.2 we outline the structure of our proposed solution to the raised problems.

### 1.1 Focus of the Thesis

#### 1.1.1 Background

As a motivating example, consider the application of participatory sensing to public health monitoring. We can make use of a variety different sensors to monitor the signs or symptoms of health issues for individual subjects, as well as relevant health hazards in the environment. In this section, we first describe the type of sensor measurements that can be used in this context, and we then describe a possible architecture for participatory sensing in this application. Based on the architecture, we then highlight the types of analysis tasks that are useful in this health monitoring context, and some of the privacy issues that are raised by these analysis tasks.

The sensors on a participant’s mobile phone or other wearable devices can provide
1.1 Focus of the Thesis

(a) activity measurements — including movement observations of the person’s limbs, body and gait; as well as (b) biometric measurements — including respiration, heart rate, oxygen levels and other metabolic indicators\cite{44,102}. Environmental sensors, which can be either phone-based or fixed sensor networks, can also provide measurements of potential hazards — including temperature, light, noise and dust.

Participants may be willing to contribute these sensor measurements to an analytics service in order survey their “normal” state, as well as detecting unusual or anomalous episodes that could affect their health. However, there is a fine line between providing useful analysis, and invading the participant’s privacy. For example:

- users might want the service to identify episodes of unusual activity or inactivity, but not identify what they were doing;
- users might want the service to identify unusual physical or metabolic responses, but not what these raw body measurements were;
- users might want the service to detect extreme environmental conditions, but not exactly when they have been exposed to these conditions.

Consequently, we require an analytics service that can support sophisticated queries on data from a potentially large number of participants, while preserving the privacy of the participants’ raw data.

We consider a Participatory Sensing Network (PSN) architecture as shown in Figure 1.1. A potentially large number of participants contribute their sensing data from their mobile device to an application server via a wireless network. The server then computes certain queries on these incoming data streams. In this thesis, we aim to identify anomalous behaviour in such collaborative networks, which could be identified with respect to the aggregated data, i.e., considering the mean and standard deviation of sampled data, or the normal distribution. Hence we focus on two main classes of queries:

i) Aggregation based queries — such as SUM or MEAN, which compute the sum of all values or the mean in a given set of data streams.

ii) Distribution based queries — detecting episodes when observations deviate significantly from the normal distribution of values in a given set of data streams.
The results of this query processing can then be made available to the clients of the service.

More formally, our aim is to detect anomalous records in a general participatory sensing architecture comprising three types of parties: a set of users (participants) $\mathcal{U} = \{u_i | i = 1, \ldots, q\}$, a mining server $\mathcal{S}$, and an arbitrary number of clients $\mathcal{C}$. Each user $u_i$ is an individual who captures data records for the same set of attributes (i.e., a horizontal partition), and contributes the sampled records to $\mathcal{S}$ for training and testing purposes. The server $\mathcal{S}$ is a third-party providing a data mining service to the participants. After receiving the contributed data records, the server trains an anomaly detector that generates a global classification model $\mathcal{M}$ from the locally collected records. The clients $\mathcal{C}$ could be the participants themselves or third-parties such as analysts trying to learn about the monitored phenomena. We share a similar underlying assumption with [30], in which the computation and communication demands on the participants should be minimised, and data processing/analysis should be shifted to the server. The anomaly detection model $\mathcal{M}$ should be in a form that can be disseminated and used by an arbitrary number of clients, and not limited to the original participants or customised for a small number of clients.

Let us now examine this problem with respect to the two main foci of this thesis, namely privacy-preserving anomaly detection, and efficient, high-dimensional anomaly detection.
1.1 Focus of the Thesis

1.1.2 Privacy-preserving Anomaly Detection

A major challenge for collaborative networks such as participatory sensing is how to maintain the trust of participants. Contributed sensor data may contain information concerning the private life of the participants, including, for example, records of intimate discussions, photos, traces of visited locations, and other confidential sensor measurements. Possible intrusion attempts can reveal the participants’ identity or the value of collected data. Preserving the privacy of anomalous records is of importance, since they often contain uniquely identifiable information that enables a data point to be linked to a data repository. Moreover, an anomalous point indicates that the observed subject has deviated from the normal pattern/behaviour. This could be a major concern in various application scenarios. For example, where data from multiple air carriers are collaboratively analysed [23], but the information must be kept private to each airline to protect proprietary information. Once participants are aware of possible threats, they may lose interest in donating their data or providing faithful information. Consequently, lack of participant interest would diminish the impact of sensing campaigns deployed on a large scale. To encourage participation, privacy mechanisms are required to encounter and mitigate intrusion risks. In this thesis we are not focussed on hiding participants’ identity, rather we aim to preserve the privacy of participants’ data.

Intrusion attempts that reveal participants’ data can be caused by external and internal parties. Secure communication channels [24] enable participants to transfer their data without worry of external attackers, e.g., eavesdroppers. However, avoiding intrusion attempts by internal parties, i.e., server, participants and clients, is a major concern. A common solution is to rely on a Trusted-Third Party (TTP) and ground privacy policy. A TTP is a trusted entity that facilitates interactions among all the internal parties in the network. For example, in the PSN architecture a trusted server collects participants raw data, analyses them, and replies to clients’ queries in a way that does not reveal any information concerning individual participants. The problem here is that policies can be over looked and trusted parties themselves can be subject to attack, therefore a more reliable mechanism is required to preserve the participants’ privacy.

A more practical approach is that participants mask sensitive attributes of their records,
before sending them to the server. Cryptographic techniques such as Secure Multiparty Computation (SMC) are a widely adopted solution that enables individual privacy, but they are computationally demanding and suffer from a lack of scalability. SMC approaches are designed for a small number of participants, while in participatory applications thousands or millions of participants may join. A more efficient and scalable solution is to use data perturbation techniques. The design of a data perturbation technique depends on the type of underlying query. As mentioned earlier, we study two common types of queries, aggregation and anomaly detection queries. Aggregation queries are used to investigate summary statistics such as mean and standard deviation, while anomaly detection queries aim to generate a global model and identify universal observations in the data. We aim to design a masking technique for these types of queries by imposing additive or multiplicative noise on the raw data. In comparison to SMC, perturbation approaches are scalable to large networks of participants and have marginal computational cost. Therefore, perturbation schemes offer a more practical solution for addressing privacy in large collaborative networks. In particular, our perturbation approaches fulfil the following requirements.

- **Privacy** — It maintains the privacy of sensitive attributes, so that given a masked record the server cannot infer the original value.

- **Accuracy** — The accuracy of the query results from masked data should be close to the results from raw (unmasked) data.

- **Efficiency** — Given that users may depend on resource constrained devices, the masking should not exhaust the available computational resources of those devices.

- **Scalability** — The system should scale to any number of participants.

- **Usability/Accessibility** — The generated results should be accessible and usable by all the participants and other end-users.

The interaction between these challenges of privacy, accuracy, efficiency, scalability and usability provide the focus for our research in Part I. This part explores the privacy concerns that are in the focus of this thesis, highlights their challenges, and describes our
proposed solution. In summary we aim to address the following open questions in this context:

Q 1. How can we provide data privacy and data integrity in aggregation queries?

Q 2. How can we design a collusion resistant privacy-preserving scheme for mining anomaly queries in collaborative networks?

Q 3. How can we improve participants’ privacy so that they can achieve better concealment of both normal and anomalous records?

Q 4. How can we give more control to participants over masking their data, while being able to generate and use a global anomaly detection model that can be applied to individually, randomly masked measurements?

1.1.3 Large-scale High-dimensional Anomaly Detection

The embedded sensors in smart phones empower ordinary individuals to collect and share sensed data from their daily life and surrounding environments. The variety of embedded sensors such as cameras, GPS, microphones, gyroscopes, accelerometers, in addition to sophisticated external sensors connected via Blacktooth or WiFi, e.g., biometric and air pollution sensors, enables enrich data collection. Given the ubiquity of smart phones and the high density of users in metropolitan areas, participatory applications can provide an unprecedented coverage for observing phenomena of interest, in both space and time. The fine-detailed data collected from participants can be aggregated to generate a comprehensive view of the studied events and to extract important statistics.

Anomaly detection (also known as outlier detection) plays a key role in data mining for detecting unusual patterns or events in an unsupervised manner. In particular, there is growing interest in collaborative anomaly detection \cite{23,55,168}, where multiple data sources submit their data to an on-line service, in order to detect anomalies with respect to the wider population. For example, in PSN’s public health monitoring applications \cite{30}, participants collect and upload their data to a central service to detect unusual events, such as the emergence of a source of pollution in environmental sensing,
or disease outbreaks in public health monitoring. When considering only one person
with the following details, e.g., height= 185 cm, weight= 50 kg, age=25, and daily calorie
intake=2500 kJ over time, the details may stay within the normal range for the individ-
ual. But if we compare this record with a wide range samples from other people we may
identify it as an anomalous record. Then a core challenge in the context of collaborative
anomaly detection is how to cope with noisy, large-scale datasets [100].

Compared with other classification problems, anomaly detection poses a special re-
quirement, in the sense that anomaly detection algorithms should be trained with unla-
belled records, i.e., trained in an unsupervised manner. Obtaining a large training set of
clean and labelled data is often a labour and time intensive task. Moreover, anomaly de-
tection becomes more challenging when applied to high-dimensional datasets that con-
tain a large number of records. Many of the available methods for identifying anomalies
assume small datasets with low numbers of features. Processing a large dataset collected
from a heterogenous environment requires an anomaly detection technique that meets
the following criteria.

• **Accuracy** — Its accuracy should be maintained as the volume or dimensionality of
  the data to be analysed grows.

• **Scalability** — Its computational- and memory-complexity should allow it to be scal-
able to large numbers of records and dimensions in the contributed data.

In particular, we aim to overcome the limitations of One-class Support Vector Ma-
chine (1SVM) [149][165], a widely used anomaly detection technique, to process large
and complex datasets. In Part II we explore how the performance of 1SVMs are nega-
tively influenced as the data volume and dimensionality grows. We investigate different
approaches to avoid computationally expensive non-linear kernels in 1SVMs, without
sacrificing the accuracy of the resulting anomaly detection model. In summary, we aim
to address the following open questions:

Q5. How can we achieve large-scale high-dimensional anomaly detection with 1SVMs?
1.2 Organisation and Contributions of the Thesis

Q 6. How can we minimise the memory- and computational-complexity of 1SVMs while achieving acceptable accuracy?

1.2 Organisation and Contributions of the Thesis

Figure 1.2 provides an overview of the organisation of this thesis. In this section we provide an overview of the focus and contributions of each chapter in the thesis.

Figure 1.2: Organisation of thesis.

Chapter 2 — Background and Survey of Privacy-Preserving and Large-Scale Anomaly Detection in Collaborative Networks

In this chapter, we give a detailed review of existing privacy-preserving data mining approaches and large-scale high dimensional anomaly detection. We also raise open questions which we believe are worthy of further research.

- We give an introduction to architectures for distributed data mining and various data sharing scenarios.

- We review existing privacy-preserving techniques and propose a taxonomy for privacy-preserving data mining.
• We highlight the characteristics of high-dimensional and large-scale datasets and the challenges they impose for anomaly detection.

• We give a survey of existing techniques to address the challenges of high-dimensional anomaly detection and highlight their limitations.

The work arising from this chapter is under review as paper P4.

Part I — Privacy-Preserving Anomaly Detection

Participants’ privacy is a central concern for participatory sensing applications. In these networks no entity, including the server, is assumed to be trusted, i.e., any party may attempt to reveal others’ data. Moreover, presence of a TTP is not guaranteed. A more practical option may be for the participants themselves to be in charge of maintaining the privacy of their own sampled measurements, i.e., masking sensitive attributes in their data. A major challenge is how to achieve accurate anomaly detection without disclosing the raw values of the participants’ data. One approach to this challenge is to randomly perturb the participants’ raw data, e.g., by adding noise to the data. However, this randomisation may result in a degradation of the accuracy of data analytics. Privacy-preserving techniques are required that can achieve a balance between privacy and accuracy, without making a major sacrifice of one for the other. Furthermore, delegating privacy maintenance to participants necessitates that the computational complexity be kept as low as possible, in order to permit the use of resource constrained devices that may be available to the participants.

To overcome these concerns, in Part I of the thesis we propose different privacy-preserving techniques to support different families of queries. The first chapter of Part I looks at aggregation queries, while the three remaining chapters look at anomaly detection queries, specifically anomaly detection queries.
Chapter 3—KSSI: Privacy-Preserving Data Aggregation in Participatory Sensing Networks

Aggregation queries allow a system to aggregate participants’ sampled measurements to provide exact summary results. In this chapter, we address, research question Q1, the problem of privacy-preserving data aggregation. Our proposed solution is based on a mutual protection approach called data slicing, where sampled measurements are split into random “slices” and transmitted to the server via the neighbouring participants in the network. Since the data slices can also be revealing of the original data, the measurements are masked (perturbed) using homomorphic encryption [76]. While concealing the real value of data slices, homomorphic encryption enables the neighbouring participants and the server to simply aggregate all the received data slices, without decrypting them.

Although this scheme ensures that data slices are likely to remain private, there is a potential risk of contaminating the aggregation result. Due to network failure or misbehaviour of a neighbour, the server may not receive all the slices in their true form, hence, it is essential that the server be able to verify the validity and legitimacy of the aggregated results. In our proposed scheme the participants provide the server with a Hash Message Authentication Code (HMAC) of their record, i.e., a one-way hash that facilitates integrity checking of the aggregated data.

The objective of this chapter is to design a data-aggregation scheme for participatory sensing systems that addresses user privacy and data integrity while keeping communication overhead as low as possible. We introduce four techniques to address these challenges and validate them through analytical models and simulations. The initial technique is based on an existing work [37] that we gradually enhance over three stages. The main characteristics of our final approach are as follows.

- It maintains the privacy of individual participants by relying on peer-collaboration.
- It ensures the integrity of the contributed data.
- It imposes low computational complexity on the participatory network.

The publication arising from the work in this chapter is paper P1.
Chapter 4 — RMP: Privacy-Preserving Collaborative Anomaly Detection for Participatory Sensing

In this chapter, we study research question Q2, the problem of privacy-preserving collaborative anomaly detection, where two or more sources contribute their local datasets to an on-line service (a data mining server), to generate a global model from the union of their records. A major challenge is how to achieve reasonable detection accuracy without disclosing the actual values of the participants’ raw data. We propose a basic approach called Random Multiparty Computation (RMP), which is a form of random perturbation to be used by each participant. Previous approaches in this context require all participants to share the same perturbation. This assumption is essential to maintain the level of accuracy, but it potentially exposes the scheme to collusion attacks (among rogue participants and the server) as well as breaches of privacy. In contrast, our proposed scheme RMP enables private perturbation, using a combination of nonlinear and participant-specific linear perturbations. The linear transformation is personalised by imposing random noise to a publicly available perturbation matrix. Empirical analysis on several benchmark datasets using a form of anomaly detector called an autoencoder [19] show that RMP yields comparable results to non-privacy preserving anomaly detection.

The main characteristics of the proposed technique are as follows:

- It is resistant to collusion, since participants independently generate their own random perturbations.

- The randomisation method has low computational complexity.

- The scheme is scalable to large numbers of participants on existing hardware platforms.

- It imposes no restrictions on the kind or the number of clients accessing the generated anomaly detection for testing.

The publication arising from the work in this chapter is paper P2.
Chapter 5 — ERMP: Enhanced Random Multiparty Perturbation

In Chapter 4, we introduced RMP to address privacy challenges for anomaly detection in PSNs. RMP maintains participants’ privacy while avoiding computationally intensive cryptographic techniques or relying on a trusted third party. It is mainly characterised by allocating a private random transformation to each participant. In this chapter, we address research question Q3 and enhance the performance of RMP by proposing a new nonlinear transformation. The new scheme, Enhanced Random Multiparty Computation (ERMP) improves the privacy of normal and anomalous records, while maintaining the accuracy of anomaly detection. Formally we elaborate the logic behind the two-stage perturbations and the role of each in thwarting major attacks, such as Bayesian estimation and Independent Component Analysis (ICA) attacks. Further, we provide a statistical privacy measure, called recovery resistance, to estimate the fraction of perturbed data that can be recovered. We conduct an extensive theoretical and experimental analysis, and compare the effectiveness of the new perturbation with the earlier method from Chapter 4.

The main characteristics of ERMP are as follows:

- It maintains the privacy of both normal and anomalous records, in contrast to existing works that can only preserve the privacy of either normal or anomalous data points.
- It achieves comparable accuracy to non privacy-preserving anomaly detection schemes on a variety of benchmark datasets.

The work arising from this chapter is under review as paper P5.

Chapter 6 — IRMP: Individually Random Multiparty Computation

Over the previous two chapters we introduced RMP, a framework for maintaining the privacy of participants in collaborative anomaly detection, and an enhanced variation of it in ERMP. These two schemes offer various desirable properties as detailed above,
a distinctive feature of which is that they enable participant-specific perturbation. Although the perturbation matrices vary from one participant to another, they are derived from a shared matrix. In this chapter, we up the ante and remove the constraint of using a shared matrix and address research question Q\text{4}. We propose IRMP, a similar perturbation scheme consisting of a nonlinear stage and a linear stage, but which allows participants to independently perturb their sampled records. Applying unique individual transformations provide the system with stronger security against malicious collusion. To maintain the level of accuracy and compensate for the individually random transformation, we make use of a deep learning architecture for our anomaly detector based on a contractive autoencoder [145]. Moreover, we show that our proposed transformation results in a clear separation of normal and anomalous records, thus facilitating anomaly detection. The results of our empirical analysis on synthetic and real-life datasets show that IRMP achieves higher overall recovery resistance than existing schemes, and comparable accuracy to non-privacy-preserving anomaly detection methods. To the best of our knowledge IRMP is the first scheme that preserves the privacy of both normal and anomalous points for anomaly detection in large collaborative networks.

The main characteristics of the enhanced IRMP are as follows:

- It enables individual random perturbation by participants.

- It delivers comparable accuracy to non privacy-preserving anomaly detection schemes (given that sampled records are perturbed independently).

The work arising from this chapter is under review as paper P\text{6}.

Part II — Large-Scale High-Dimensional Anomaly Detection

The second issue that we have studied in participatory networks is the problem known as the “curse of dimensionality”. The growing range of available sensors gives the opportunity for participants to capture a wide variety of features. Having a greater number of features is interesting for anomaly detection, as it has the potential to help identify more complex unusual behaviour [190]. The challenge is actually drawn from traditional
anomaly detection techniques, such as 1SVMs. They face two major shortcomings when it comes to high-dimensional data analysis: their computational complexity grows exponentially, and they can fail to generate a robust model, which can result from the introduction of noise in the form of irrelevant features to the data. In Part II, we investigate how to overcome these challenges and build a robust anomaly detection model from high-dimensional data using 1SVMs.

Chapter 7—DBN1SVM: High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning

One way to building a robust anomaly detection model for use in high-dimensional spaces is to combine an unsupervised feature extractor and an anomaly detector. While one-class support vector machines are effective at producing decision surfaces from well-behaved feature vectors, they can be inefficient at modelling the variations in large, high-dimensional datasets. In contrast, architectures such as Deep Belief Networks (DBNs) are a promising technique for learning robust features. In this chapter we address the research question Q5, and present a hybrid model where an unsupervised DBN is trained to extract generic underlying features, and a one-class SVM is trained from the features learned by the DBN. The main characteristics of the proposed technique are as follows:

- It boosts the scalability of a one-class SVM for use with large-scale and high-dimensional datasets.

- Its deep architecture provides better generalisation, and hence enables the replacement of nonlinear kernels with linear ones, which are computationally much simpler.

- It delivers comparable accuracy to deep autoencoders, and significant improvement in accuracy compared to traditional one-class SVMs.

- It achieves a three-fold reduction in training time, and a 1000-fold reduction in testing time compared to deep autoencoders, which are an established method for deep learning.
The work arising from this chapter is under review as the paper P\(^7\).

**Chapter 8 — R1SVM: a Randomised Nonlinear Approach to Large-Scale Anomaly Detection**

To address research question Q\(^6\) in this chapter we present a Randomised One-class Support Vector Machine (R1SVM) model for anomaly detection. We begin by recalling a few key aspects of one-class SVMs, and then introduce the use of nonlinear random projections for detecting anomalies in large-scale data. Random projections have been utilised mainly in distance-based classification or data reconstruction schemes to speedup search, as it approximately preserves $L_2$ distances among a set of points. Thus instead of performing the search in a high-dimensional space, the search is conducted in a space of reduced dimension but on a larger neighbourhood.

The main characteristics of R1SVM are as follows:

- It enables nonlinear kernels to be replaced with linear kernels, which are computationally much simpler.
- It delivers comparable accuracy to deep autoencoders, and a significant improvement in accuracy compared to traditional one-class SVMs.
- It achieves 1000-fold speed up in training and testing time compared to traditional one-class SVMs, and 100-fold speed up compared to deep autoencoders.

The publication arising from the work in this chapter is paper P\(^3\).
Chapter 2

Background on Privacy-Preserving and Large-Scale Anomaly Detection in PSNs

The detailed data collected from participatory sensing applications offers a rich resource for in-depth study and discovery of phenomena of interest. Anomaly detection is a particularly interesting application in this context as it enables the detection of significant events or unusual behaviours with respect to a wider population of participants. Two major challenges arising in the context of participatory learning are privacy and data analysis on high-dimensional datasets. This chapter reviews existing works on these two challenges and raises a variety of open questions. It starts with an introduction to the architectures of distributed data mining and data sharing scenarios. To propose an effective solution to the above challenges it is necessary to have a clear understanding of the participatory architecture and its characteristics. Then this chapter surveys existing privacy-preserving techniques and proposes a taxonomy for privacy-preserving data mining. Finally, it highlights the characteristics of high-dimensional datasets, the challenges they create for anomaly detection, and common techniques to overcome these challenges.

The work arising from this chapter is under review as paper P4.

2.1 Basic definitions

Central to the Internet of Things [O1, S8, 93] is the ability to capture data, either through sensors or active contribution by people — the latter data collection paradigm is known as participatory sensing, where individuals and communities use evermore capable mobile phones and cloud services to systematically collect and analyse data about the participants [30, 62]. Participatory sensing is also referred to as opportunistic sensing, people-
centric sensing and many other names [44]. To emphasise the characteristics of PSNs, in this section we describe various data sharing scenarios. We discuss typical types of data, databases and data mining operations, since they determine the type of applicable privacy and anomaly detection approaches.

Data sharing scenarios

Suppose the City of Melbourne is monitoring urban noise [54], by collecting noise samples from noise sensors and through participatory sensing. The data, collected and deposited in one or more databases belonging to the City of Melbourne, can contain sensitive attributes / features / variables, such as the participants’ location and speech. The participants are data owners [72] or data donors [26], whereas the City of Melbourne plays the role of a data curator [58] or data custodian [69], who is ethically (if not legally) bound to protect the participants’ sensitive information. A data curator may also provide mining services, or may just collect the data from participants and out-source the task to a third-party data miner. Another stakeholder that appears in some data sharing scenarios is a client, who is a third-party who can access the data mining results (in addition to the participants). Regarding the existence of a third-party client, we can categorise PSN architectures as open or closed. The former determines the cases where a third-party can learn of the data mining results as well as the curator (data owner), while the latter determines the case where no third-party can learn of the results, and the results are only accessible by the curator (data owner).

We often speak of data in the relational sense, where an individual’s data is divided into attributes that form a record (hence the reference to data owners as record owners [69]), and multiple records are grouped into a table, and multiple tables are stored in a relational database. This makes sense for slow-changing data. In some participatory applications, however, where data is collected over a period of time, time series data and time series databases are the norm. In this thesis we only consider relational databases, and real time data analysis is postponed for future work. Regardless of the data type, data sharing scenarios determine the requirements for privacy preservation solutions, and correspondingly the required data mining techniques. Privacy issues in some sce-
narios necessitate special consideration, which may disturb the utility of some conventional data mining techniques (see Section 2.2). The following are the most common data sharing scenarios, as depicted in Figure 2.1:

• **Data publishing**: In this scenario, a data curator publishes data records (not aggregate statistics or any data mining results) about individuals. The privacy requirements for this publication are that an attacker cannot (i) identify the presence or ab-
Background on Privacy-Preserving and Large-Scale Anomaly Detection in PSNs

The presence of an individual’s record in the database, (ii) identify the record belonging to an individual, and (iii) infer the values of an individual’s sensitive attributes. The problem of preserving privacy in this scenario is referred to as privacy-preserving data publishing [39, 69].

- **Statistical disclosure**: In this scenario, a data curator releases aggregated statistics (e.g., sample mean and count) about a group of individuals represented in the database to a third-party. The privacy requirement for this release is that an attacker cannot learn any of the sensitive attributes of any individual represented in the database [4]. The problem of preserving privacy in this scenario is referred to as statistical disclosure control [48] or statistical disclosure limitation, and research in this area dates back to the 1970s.

- **Computation outsourcing**: In this scenario, a data curator, for lack of local resources, outsources its data mining operations to a third-party data miner with more resources. The privacy requirement is that the data miner cannot learn any part of the curator’s data or the generated result, implying that the data curator must send its data to the data miner in a masked form. In the literature this problem is usually referred to as verifiable computing [75], in the case that the data curator must be able to verify the result, or secure outsourced computation [116] in the case that the result does not need to be verified. For more relaxed forms of these problems, there are no specific names.

- **Distributed data mining**: In this scenario, multiple data curators jointly analyse the union of their data. In some cases a third-party data miner may exist to assist the curators with data mining operations. The privacy requirement is that no data curator (or data miner, if one exists) can learn any part of the data belonging to any other data curator. The problem of preserving privacy in this scenario is referred to as “Privacy-Preserving Distributed Data Mining” (PPDDM) [45]. Historically, Lindell and Pinkas [110] were the first to study an instance of this problem involving two data curators and no third-party, under the less precise title “privacy-preserving data mining”.
2.1 Basic definitions

- **Collaborative learning**: In this scenario, multiple data owners enlist the help of a third-party data miner to analyse the union of their data. Traditionally, data owners trust the data curator to protect their privacy, but rising privacy awareness creates the expectation that data owners cannot only send their data to the data curator securely, but also send their sensitive attributes in a *masked* form (e.g., masked value = original value $\times$ artificial noise). The privacy requirement is that a third-party cannot infer the original values from the masked values. The above description is also valid if we replace “data owners” with “data curators”. The problem of preserving privacy in this scenario is referred to as “privacy-preserving collaborative learning” [111].

Remarks on terminology:

- Liu et al. [111] define “collaborative learning” as a process where “multiple users contribute individually collected training samples (usually extracted from raw sensor data) so as to collaboratively construct statistical models for tasks in pattern recognition”. This is classified under “distributed data mining” in the data mining literature [186, Ch. 13], but privacy considerations necessitate the detailed differentiation proposed above. In fact, the term “collaborative learning” is widely used in the participatory sensing literature (e.g., [5, 46, 61, 86, 92, 117, 121, 128]). Succinctly, the major difference between collaborative learning and distributed data mining lies in the role of data owners, i.e., in the collaborative learning scenario data owners only provide the data, while in the distributed data mining scenario data owners (curators) also contribute to the data mining process.

- We use Privacy-Preserving Data Mining (PPDM) as an umbrella term to refer to the preservation of privacy in any of the above scenarios except data publishing [57]. The focus of our work is PPDM for the collaborative learning scenario.

**Types of data and databases**

Different PPDM approaches are applicable to different types of data and different types of databases. We speak of data as a collection of attributes. An attribute is either *continuous*
/ numerical / quantitative, if it is numerical and arithmetic operations can be defined on
the attribute; or categorical, if it assumes values from a finite set and arithmetic operations
on it do not make sense. A categorical attribute is further ordinal if it takes values from
an ordered set, but nominal otherwise.

Data mining operations are performed over a collection of attributes (attributes associated
with an individual, and attributes from multiple individuals) organised in the form
of a database. Today still, relational databases represent the most common type of data
storage. A relational database contains tables, and a table contains records/tuples/rows,
and a record contain attributes/columns. Relational datasets belonging to different data
owners (or data curators), such as in distributed data mining or collaborative learning
scenarios (see Figure 2.1), are said to be:

- horizontally partitioned if different parties hold different records containing the same
  attributes;

- vertically partitioned if different parties hold different attributes of the same records;
  and

- arbitrarily partitioned if for each record, the attributes are arbitrarily distributed across
  the parties [94].

The manner of partitioning affects the design of PPDM solutions for relational data. Section
2.2 is devoted to PPDM schemes for relational data.

Data mining operations

A discussion of PPDM for the IoT would not be complete without a discussion of the
common data mining operations on privacy-sensitive data in the IoT. “Data mining” is
popularly used to refer to “knowledge discovery in databases” when in fact it refers to a
step in the latter. In short, data mining is the process of discovering interesting knowl-
edge from data stored in databases, data warehouses, or other information repositories.
In the IoT, most data is unlabelled due to the impractical cost of manual labelling, and
hence unsupervised learning is required. Among the myriad of unsupervised learning
2.2 Survey of Privacy-Preserving Data Mining

Data generates value when it is shared, but data sharing raises privacy issues. The privacy literature has no shortage of examples of how shared data can be misused to violate individual privacy (see [44] for a survey). Similarly, many PPDM schemes have been proposed to date (see [6, 39, 50, 66, 69] for surveys), but there are still many open problems, both practical and theoretical. Before reviewing existing PPDM approaches, we determine the adversarial models and privacy requirements of PPDM in the following.

2.2.1 Adversarial Models

Attacker models describe an attacker’s abilities and the constraints under which it operates. Generally, PPDM deals with two main types of attackers:

- **Semi-honest** (also called honest-but-curious) attackers are passive attackers that abide by the protocol but try to learn the private states of other parties from the information they receive during the protocol. By definition, they do not collude with each other. This is a weak attacker model that is commonly used to model inadvertent leakage of information by honest parties, for example, the case where an honest party becomes compromised after following a protocol to completion, thereby allowing the attacker to obtain a transcript of the protocol execution.

- **Malicious** attackers are active attackers that deviate arbitrarily from the protocol, in addition to trying to learn the private states of other parties.

The ultimate goal of PPDM is of course to be secure against malicious attackers. A common approach to designing a PPDM scheme is to first frame it in the semi-honest model, and then enhance it for the malicious model. This enhancement can be achieved through the use of zero-knowledge proofs to ensure that each protocol step follows the
protocol specification [134], but this generic method is often not practical. Luckily, the semi-honest adversarial model is often realistic, because it captures the all-too-common partial trust relationships between users and service providers in the real world who are usually not malicious. For example, email users trust their service providers to relay their emails but do not trust them to not snoop on their correspondence. Most PPDM solutions to date target the semi-honest model.

Against either semi-honest or malicious attackers, it should be clear at the outset that plain encryption is not a solution for PPDM. Legal contracts such as nondisclosure agreements are necessary, but even if data disclosure is confined to contractees, the possibility of data theft despite the contractees’ greatest effort motivates the goal of PPDM: released data should be sanitised in such a way that even if the data is accessible to unauthorised parties, the privacy of the data owners is preserved.

2.2.2 Taxonomy of PPDM Approaches

Generally, we can categorise PPDM approaches as either syntactic or semantic, regarding the type of attacks they deal with.

**Syntactic approaches** aim at minimising the correlation among attributes of different data sources to prevent syntactic attacks [69], which include:

- **Table linkage** — In this attack the attacker has access to an anonymous table and a nonanonymouse table, with the anonymous table being a subset of the nonanonymouse table. The attacker can infer the presence of its target’s record in the anonymouse table from the target’s record in the nonanonymouse table.

- **Record linkage** — In this attack the attacker has access to an anonymous table and a nonanonymouse table, and the knowledge that its target is represented in both tables. The attacker can uniquely identify the target’s record in the anonymouse table from the target’s record in the nonanonymouse table.

- **Attribute linkage** — In this attack the attacker may not be able to identify the record of the target victim, but it can infer the sensitive value(s) of the target from the group to which the target belongs (e.g., 30-40 year-old females).
A widely used example of a syntactic approach to PPDM is $k$-anonymity \cite{147,162}. A table is said to provide $k$-anonymity for $k > 1$ if for each combination of quasi-identifier values (e.g., age=“30-40 years”, gender=“female”), at least $k$ records in the table share that combination. A quasi-identifier is a set of attributes that can potentially identify the record owners. From the discussion so far, it should be clear that syntactic approaches are inherently oriented towards relational data.

**Semantic approaches** aim to satisfy semantic privacy criteria, which are concerned with minimising the difference between adversarial prior knowledge and adversarial posterior knowledge about individuals represented in the database. Prior (posterior) knowledge is commonly expressed in prior (posterior) probabilities. Informally, semantic security says that the masked data should reveal no information about the raw data (the original message). The following is an example of common attacks to semantic approaches.

*Known input-output attack* — In this attack the attacker has access to some masked records and their corresponding original text. These can help to reveal the secret information used to mask the records, and eventually reveal other masked records.

Both syntactic and semantic PPDM approaches involve masking data to hide the real values from the data miner. Common data modification functions include:

*Generalisation* — the generalisation of an attribute value to a range in order to reduce data granularity. For example, age=“35 years” can be generalised to age=“30-40 years”.

*Suppression* — the removal of an attribute value or entire tuples. The original $k$-anonymity scheme \cite{147} uses just generalisation and suppression. For example, achieving acceptable anonymisation in a dataset that includes anomalies may require significant generalisation (which results in significant information loss). Therefore, anomalous records can be suppressed by removing them.

*Randomisation* — the randomised distortion or perturbation of data, mostly through a linear or nonlinear transformation. For example, randomising the data by adding random noise.
More data modification functions such as condensation, data swapping, rank swapping and synthetic data generation are discussed in \cite{6,69}; most of these functions are used for syntactic privacy. Since in participatory sensing the participants are responsible for masking their own data before they send it to the data miner (i.e., no entity has access to the entire database), semantic approaches are the only options here. In this context randomisation or SMC\footnote{Some authors write “secure MPC” instead of “SMC”. Secure multiparty computation is also referred to as secure function evaluation.} are more prevalent approaches. The next section surveys privacy criteria/definitions with a focus on semantic approaches.

Semantic approaches to privacy

**Secure Multiparty Computation (SMC):** SMC refers to the distributed computation of a publicly known function \( f(x_1, \ldots, x_n) \) by \( n \) parties with respective inputs \( x_1, \ldots, x_n \), such that at the end of the computation, each party only learns its own input and the function’s output. Despite advances in homomorphic encryption (a building block of SMC) \cite{76}, SMC-based schemes still require significant time and computing resources from the participants, conflicting with the vision of participatory sensing. For example, Chen et al.’s scheme \cite{43} and Bansal et al.’s scheme \cite{13} have a computational/communication complexity that is quadratic in the number of participants. Yuan et al.’s scheme \cite{187} requires the participants to “stay online with broadband access to the cloud,” and be “equipped with one or several contemporary computers.” SMC-based privacy-preserving anomaly detection schemes like \cite{47,55,168} face similar challenges.

Some efficient homomorphic cryptosystems such as CMT \cite{37} propose an efficient and provably secure additive homomorphic stream cipher, which can deliver an effective approach to encrypted data aggregation. In particular, CMT replaces Exclusive-OR (\( \oplus \)), an operation typically found in stream ciphers, with modular addition (\( + \)). Since the message size in this scheme can be quite small (e.g., few hundreds of bits) and it applies only modular addition to the cipher streams, it is quite efficient and can be implemented on many resource-constrained devices. As further discussed in Chapter \cite{3}, schemes such as CMT are only suitable for aggregation queries. To fulfil the privacy requirements of more
sophisticated mining queries, randomisation techniques are more appropriate.

**Randomisation approaches:** A more apposite approach to semantic PPDM is data *randomisation*, which refers to the randomised distortion or perturbation of data prior to analysis. Perturbing data elements, for example, by introducing additive or multiplicative noise, allows the server to extract the statistical properties of records without requiring the participants to allocate substantial resources or coordinate with the server during the training process. We now outline the main types of data perturbation methods: *additive, multiplicative, geometric, and nonlinear* transformations.

- **Additive perturbation** perturbs data matrix $X$ by adding i.i.d. (independent and identically distributed) noise $R$ from a zero mean, unit variance Gaussian distribution, as $R + X$ [7,8]. Since independent random noise can be filtered out [91,98], the general principle is that the additive noise should be correlated with the data [72,131].

- **Multiplicative perturbation** multiplies data $X$ with random noise matrix $R$, as $RX$. In general, multiplicative perturbation is the most common privacy-preserving approach in large collaborative architectures, although the strength of the perturbation is reliant on the design of the perturbation matrix $R$. So far three types of perturbation matrices have been used in the context of PPDM — rotation perturbation, random projection and uniform random transformation.

  - *Rotation Perturbation* — the noise matrix $R$ is an orthogonal matrix with orthonormal rows and columns [40]. This scheme is vulnerable to “known input” attacks [77], where a few leaked inputs are sufficient to recover the original data from its perturbed version.

  - *Random Projection (RP)* — the noise matrix is a rectangular zero-mean Gaussian random matrix $R \sim \mathcal{N}(0,\sigma)$, effectively projecting the data matrix to a lower dimension [114]. Random projection is in fact a Johnson-Lindenstrauss transform [96], a general tool for dimensionality reduction. Note that while rotation perturbation is absolutely distance-preserving, random projection is only statistically distance-preserving. However, if the original data follows a
multivariate Gaussian distribution, a large portion of the data can be reconstructed to within 20% error from its randomly projected version via maximum a posteriori estimation [148].

Uniform Random Transformation (RT) — the noise matrix $R \sim U(0, 1)$ is a random matrix with elements uniformly distributed between 0 and 1 [109, 123, 124]. This scheme is inspired by the Reduced Support Vector Machine (RSVM) [107], which replaces the conventional kernel matrix $K(X, X)$ with $K(X, \tilde{X})$, where $\tilde{X}$ is a random subset of the original data matrix $X$. Mangasarian et al. [123, 124] further replace $\tilde{X}$ with a random matrix (of a reduced dimension from that of $X$). Unlike rotation perturbation and random projection, this perturbation scheme does not aim for orthogonality or distance preservation.

- **Geometric perturbation** offers an alternative to purely additive or purely multiplicative perturbations [42]. Under this perturbation, the data matrix $X$ is mapped to $RX + \Psi + \Delta$, where $R$ is a rotation perturbation matrix, $\Psi$ is a random translation matrix that translates the same attributes of different records by the same amount, and $\Delta$ is an i.i.d. Gaussian noise matrix. Without $\Delta$, geometric perturbation is vulnerable to known input attacks [77]. To the best of our knowledge there are no general results on how the $\Delta$ term influences the effectiveness of known input attacks.

- **Nonlinear transformations** can be used to break the distance-preserving effect of a distance-preserving transformation on some data points. Nonlinear transformations are not a common type of data perturbation. An example of this is the work of Bhaduri et al. [23], which defines a general nonlinear perturbation as $B + QN(A + RX)$, where $B$, $Q$, $A$, $R$ are random matrices, and $N$ is a bounded, continuous nonlinear function that operates on individual matrix elements, such as tanh. This function approximately preserves the distance between normal data points, but collapses the distance between outliers [23] — hence its suitability for anomaly detection. However, this approach only preserves the privacy of anomalous records,
while the normal records are vulnerable to Maximum a Posteriori (MAP) estimation.

Note that the aforementioned works satisfy privacy requirements in the case where all participants are assumed to be semi-honest (i.e., participants do not collude with other parties) and agree on using the same perturbation matrix. As discussed in Chapter 4, this assumption is restrictive in applications such as participatory sensing. A few works have attempted to extend current randomisation approaches to a collaborative learning architecture [41, 111]. For example, the framework of Chen and Liu [41] is a multiparty collaborative mining scheme using a combination of additive and multiplicative perturbations. This scheme requires the participants to stay in touch for an extended period to generate the perturbation matrix, which is impractical for large-scale participatory sensing. Liu et al. [111] build a framework on top of Liu et al. [14], and allow every participant to perturb the training data with a distinct perturbation matrix. This scheme then regresses these mathematical relationships between training samples in a unique way, thereby preserving classification accuracy. Although participants use private matrices, the underlying transformation scheme is still vulnerable to distance inference attacks [113, 148].

2.2.3 Summary

Data randomisation is the most promising approach for privacy-preserving data mining in large collaborative networks such as participatory sensing. Unlike distance-preserving transformations, random transformation does not preserve Euclidean distances and inner products between data instances, hence it thwarts distance inference attacks. However, if this method is applied to collaborative learning, all the participants must agree on the same perturbation matrix, and then collusion attacks may succeed. Moreover, the mining models generated from the perturbed records are specific to the participants. Hence, the open question is how to designing a secure participant-specific randomisation approach that can be applied to open scenarios where the clients can be untrusted third-parties without risking participants’ privacy. Further study of the challenges associated with privacy issues in collaborative learning is conducted in Part I and several solutions are provided
to overcome these challenges.

This section studied the state-of-the-art privacy-preserving data mining approaches for large collaborative networks. The next section briefly reviews anomaly detection techniques for use in this context.

2.3 Survey of Large-Scale Anomaly Detection

2.3.1 Background on Anomaly Detection

An anomaly (or outlier) is an observation that appears to be inconsistent with the remainder (majority) of the dataset, and hence arouses the suspicion that it can be generated by a different mechanism. The objective of anomaly detection is to mine unusual and interesting information from a large amount of data. Anomaly detection has been extensively studied in various disciplines such as statistics, data mining, machine learning and information theory, and its applications has been widely expanded to numerous domains such as fraud detection, network intrusion, health monitoring, environmental monitoring and performance analysis.

A straightforward solution for anomaly detection is to construct a pattern of normal observations, and then one can use the pattern to identify anomalies. When applying the pattern on test data, the observations whose characteristics follow the normal pattern are declared as normal, and those that deviate significantly from the normal pattern are labelled as anomalies. Based on the availability of labeled training data, anomaly detection techniques can be classified into three main categories, namely, supervised, semi-supervised and unsupervised approaches.

In supervised learning all the training samples are required to be paired with a label or desired output, i.e., normal or abnormal observations, to characterise all anomalies or non-anomalies. Semi-supervised learning techniques make use of unlabelled records as well as labelled ones for training. Typically semi-supervised techniques are trained with a large amount of unlabelled records and a small amount of labeled records. It should be noted that pre-labeled data is not always available nor easy to obtain in many real-life applications such as PSNs, and also new types of observations (normal or abnormal) may
occur that are not included in the labeled training data. Unsupervised approaches are often more appealing for anomaly detection, since they require no labelled data, rather they apply certain criteria to identify anomalous observations. An example type of such techniques are distance-based approaches, e.g., classifying records according to the average distance between every data record to its corresponding the nearest neighbour observations. If the measured distance for a given record significantly exceeds nearest neighbour distance of all objects then the data record is considered as an anomaly, otherwise it is considered to be normal. In the next section, we survey some of the existing unsupervised anomaly detection techniques, which form the basis for this thesis. After that we study the challenges that traditional anomaly detection techniques face in the domain of PSNs.

2.3.2 Anomaly Detection Techniques

We begin this section by describing two types of anomaly detection techniques: one-class SVMs, and deep autoencoders.

i. One-Class SVMs (1SVMs)

One-class SVMs are widely used for anomaly detection. Their general approach is to implicitly map the data vectors (measurements) from the input space to the feature space by means of a non-linear kernel function. The mapped vectors in the feature space are image vectors. Then a smooth surface or boundary is found in the feature space that separates the image vectors into normal and anomalous measurements. By using a Mercer kernel function [169], the data vectors are implicitly mapped to a higher dimensional inner product space without any knowledge of the mapping function. The boundaries found in the implicit feature space usually yield a non-linear boundary in the input space [150].

Supervised SVMs (or lSVM, short for labelled SVM) are not always practical for anomaly detection, due to their need for labelled training data that identifies both normal and anomalous records in the dataset, or relying on “pure” normal data that are free of any anomalies [149]. Such training datasets are expensive to compile in practice. Unsuper-
The figure on the top shows a simple dataset. Normal records are represented with solid dots and anomalies with crosses. The figures on the bottom, show the data projected to a higher dimensional space using four different 1SVM approaches.

**Figure 2.2: Comparing various 1SVM methods.**

Unsupervised anomaly detection approaches such as 1SVMs overcome this challenge by constructing a nonlinear model of normal behaviour, where data points that deviate from the normal model are identified as anomalies [149, 165]. Recently, several 1SVM methods have been proposed for anomaly detection and some of the state-of-the-art 1SVM formulations are [104, 139–141, 149, 165, 173]. In the following, we briefly described these techniques and Figure 2.2 illustrates the difference among them.

Schölkopf et al. [149] proposed a Hyperplane-Based One-class Support Vector Machine (PSVM), where image vectors in the feature space are separated from the origin by a hyperplane with the largest possible margin. The vectors in the half space containing the origin are identified as anomalous. This scheme uses quadratic optimisation to fit the hyperplane. Campbell and Bennett [33] formulated a linear programming approach for the 1SVM when used with a Radial Basis Function (RBF) kernel. This formulation is
2.3 Survey of Large-Scale Anomaly Detection

based on attracting the hyperplane towards the average of the image data distribution, rather than minimising the maximum norm distance from the bounding hyperplane to the origin as in [149].

Tax and Duin [165] formulated the 1SVM using a hypersphere, in which a minimum radius hypersphere is fixed around the majority of the image vectors in the feature space. The data vectors that fall outside the hypersphere are identified as anomalies. The optimisation of this hypersphere formulation uses quadratic programming. Further, Tax and Duin [165] have shown that the hyperplane-based 1SVM becomes a special case of the (equivalent) hypersphere-based scheme when used with a radial basis kernel. Wang et al. [173] formulated the 1SVM using hyperellipsoids with minimum effective radii around a majority of the image vectors in the feature space. This hyperellipsoidal formulation involves two phases. First, the image vectors are partitioned into a number of distinct clusters using Ward’s linkage algorithm [176]. Second, the image vectors in each cluster are fixed with a hyperellipsoid that encapsulates a majority of the image vectors in that cluster. The image vectors that do not fall within any of the hyperellipsoids are identified as anomalous. This problem is formulated as a second order cone programming optimisation problem, imposing more computation than quadratic programming.

Laskov et al. [104] have observed the one-sidedness of the data distribution in many practical applications, and exploited this property to develop a special type of SVM called a one-class Quarter-Sphere Support Vector Machine (QSSVM). This is extended from the hypersphere-based 1SVM approach proposed by Tax and Duin [165]. The QSSVM finds a minimal radius hypersphere centred at the origin that encapsulates a majority of the image vectors in the feature space [104]. Data vectors that fall outside the quarter sphere are classified as anomalies. This problem is formulated as a linear programming problem. In [141], a distributed approach is presented using the QSSVM. In [139, 140] a centered-hyperellipsoidal 1SVM, called a Centered Hyperellipsoidal Support Vector Machine (CESVM), is presented based on a linear programming approach. Among all the available 1SVMs, we make use of one of the widely adopted techniques by Tax and Duin [165], which is known as Support Vector Data Description (SVDD).

Support Vector Data Description (SVDD): 1SVMs aim to model the underlying dis-
Background on Privacy-Preserving and Large-Scale Anomaly Detection in PSNs

Distribution of normal data while being insensitive to noise or anomalies in the training records. A kernel function implicitly maps the input space to a higher dimensional feature space to make a clearer separation between normal and anomalous data.

SVDD \cite{165} essentially finds the smallest possible hypersphere around the majority of the training records, while leaving out some points to be excluded as anomalies. Denoting the centre of the hypersphere by \( a \) and its radius by \( R \), this hypersphere formulation involves solving the following quadratic programming optimisation problem

\[
\begin{align*}
\min_{a, R, \xi} & \quad R^2 + \frac{1}{mv} \sum_{i=1}^{m} \xi_i \\
\text{s.t.} & \quad \| \phi(x_i) - a \| \leq R^2 + \xi_i, \\
& \quad \forall i = 1, \ldots, m, \xi_i \geq 0.
\end{align*}
\]

where \( \{x_i | i = 1, \ldots, m\} \) denotes the training data, \( \xi_i \) are the slack variables that allow some of the data vectors to lie outside of the hypersphere, \( \nu \in (0, 1] \) is a regularisation parameter that controls the fraction of anomalies and the fraction of support vectors, and \( \phi \) is a feature mapping to a higher space \( \mathcal{H} \).

The dual problem for the above primary problem is as follows

\[
\begin{align*}
\max_{\alpha} & \quad \sum_{i=1}^{m} \sum_{t=1}^{m} \alpha_i \alpha_t (x_i, x_t), \\
& \quad \forall i, t \quad 1 \leq i, t \leq m, \\
& \quad 0 \leq \alpha_i \leq \frac{1}{mv}.
\end{align*}
\]

Maximising this optimisation gives a set of \( \alpha_i \) that identify whether \( x_i \) is normal (\( \alpha = 0 \)), a support vector (\( 0 < \alpha \leq \frac{1}{mv} \)), or an outlier (\( \alpha = \frac{1}{mv} \)).

ii. Deep Autoencoders

Autoencoders were first introduced in the early 1980’s \cite{146} as a form of multilayer perceptron that has the same number of inputs and outputs. In practice, training autoencoders was limited to networks with a small number of hidden layers, due to the effects of local minima and random initialisation on the gradient-based optimisation of the non-
convex objective function used in deep multilayer networks [16]. This problem persisted until 2006 when a deep network was trained by using unsupervised pre-training followed by back-propagation fine-tuning [89]. Since then, many approaches have been introduced for deep learning, which follows the two-stage training method of pre-training and fine-tuning (a comprehensive review of these approaches can be found in [16]).

In general, an autoencoder comprises two parts, an encoder and a decoder. The encoder normally forms a bottleneck, and maps the input data to a hidden layer representation in a lower dimensional space, while the decoder maps back from the hidden representation into a reconstruction of the input at the output. The goal here is to minimise the reconstruction error between the input and output representations. By using only unlabelled data in a deep architecture (e.g., three or five hidden layers) with non-linear activation functions, autoencoders have been shown to learn useful features from complex and high dimensional datasets [21].

A major advantage of using an autoencoder as a building block is that it yields a non-linear representation that can be stacked, to capture a deeper network architecture, which can capture more abstract features and provide better generalisation. By minimising the reconstruction error, an autoencoder captures as much of the information contained in each training record as possible. In addition to the basic autoencoder, four main regularised autoencoders have been proposed, which we summarise here. Regularised autoencoders are interesting since they have to balance two factors: the reconstruction error, which forces autoencoders to maintain enough information to distinguish different training samples from each other, as well as a regulariser that prefers a representation insensitive to variation in the input data. Now we formally define the main types of autoencoders. Figure 2.3 illustrates how to stack autoencoders and train them in a greedy manner. The greedy layer-wise approach for pretraining a deep network (i.e., initialising the parameters) works by training one layer at a time, and uses the trained layer as an input to the subsequent layer. As can be seen in Figure 2.3a, the first layer is trained on the raw dataset to obtain the parameters, by transforming the input $x$ into the hidden units $h_1$. The procedure is repeated for the subsequent layers, using the hidden layer of the previous layer $h_{i-1}$ as the input for the current layer $i$. 
The greedy layer-wise method trains the parameters of each layer, while freezing parameters of the other layers. Once this stage of training is complete, to obtain better results, the parameters of all layers can be tuned at the same time by using backpropagation. We now summarise four common forms of autoencoder that are used in this thesis: (i) the basic autoencoder, (ii) the autoencoder with weight-decay, (iii) the denoising autoencoder, and (iv) the contractive autoencoder.

**Basic Autoencoder (AE):** In the basic autoencoder \cite{19}, the encoder is a function $f$ that maps an input vector $x \in \mathbb{R}^n$ to a hidden representation $h \in \mathbb{R}^d$, where $d < n$, e.g.,

$$h = f(x) = s_f(W_f x + b_f).$$

In Eq. (2.3.2), $s_f$ denotes the encoder’s activation function. Typically a nonlinear function like $Sigm(\chi) = \frac{1}{1+e^{-\chi}}$ is used for the activation function, and $W$ and $b$ are the encoder parameters, where $W_f \in \mathbb{R}^{n \times d}$ is a weight matrix, and $b_f \in \mathbb{R}^d$ is a bias vector.

The decoder function $g$ maps the hidden representation $h$ back into a reconstruction $x'$, as

$$x' = g(h) = s_g(W_g h + b_g).$$

Figure 2.3: Stacked autoencoder.
The decoder’s activation function $s_g$, similar to $s_f$, could be a nonlinear function or a linear function. The decoder is parameterised by a bias $b_g \in \mathbb{R}^n$ and a weight matrix $W_g$, where $W_g$ and $W_f$ are tied (i.e., $W_g = W_f^T$). Training an autoencoder implies finding a set of parameters $\theta = \{W_f, b_f, b_g\}$ that minimise the reconstruction error between the inputs and outputs $l(x, x') = \| x - x' \|_2^2$, by minimising the training objective function

$$L_{AE}(\theta) = \sum_{x \in X} l(x, x').$$

**Autoencoder + Weight-decay:** The most common regulariser is ($L_2$) weight-decay, in which the objective function favours small weights by simply adding a $\frac{\lambda}{2} \| W \|_2^2$ penalty term to (2.1), where $\lambda$ is a hyper-parameter that controls the strength of the regularisation. This regulariser is also known as shrinkage since the squared $L_2$ norm pushes the weights towards zero, in proportion to their magnitude.

**Denosing Autoencoder (DAE):** The Denoising Auto-Encoder (DAE) \cite{71} uses the idea of forcing an autoencoder to learn the data distribution without a constraint on the dimensionality or sparsity of the captured representation. The principle of the DAE slightly deviates from the basic autoencoder. It corrupts the input, using additive isotropic Gaussian noise: $\tilde{x} = x + \epsilon, \epsilon \in \mathcal{N}(0, \sigma^2 I)$, before sending it through the autoencoder, and then trains it to reconstruct the clean version $x$. The level of corruption $\sigma$ controls the level of regularisation.

**Contractive Autoencoder (CAE):** The CAE \cite{145} encourages mappings that are more strongly contracting at the training samples, i.e., it has small derivatives in all directions. It includes a regulariser that encourages the derivatives of $f$, the squared Frobenious norm of the Jacobian matrix $\| \frac{\partial f(x)}{\partial x} \|_F^2$, to be as small as possible. The compromise between the reconstruction error and the introduced penalty yields an autoencoder with tiny derivatives in most directions, except the ones required to reconstruct the training samples, i.e., the directions that are tangential to the manifold near which the data concentrates. Comparing the DAE with the CAE, the former learns to contract the reconstruction function, whereas the latter learns to contract the encoder.
2.3.3 Anomaly Detection in High-dimensional Datasets

As mentioned in Section 1.1.3 in PSNs each node may be equipped with multiple sensors (e.g., mobile sensors and wearables), and the data make sense when the correlations among attributes of sensor data are considered. The trend today is towards more observations, collecting fine details about each observed instance, where a single observation might comprise tens or thousands of dimensions. Capturing numerous features is desirable since certain random fluctuations, i.e., anomalous behaviour, can only be observed if studying high dimensions [53]. Looking at attributes individually, sometimes none of them may reveal any anomalous behaviour. For example, considering a patient who is suffering from high blood pressure and is under medication. Just by looking at blood pressure measurements one may not be able to quantify the effectiveness of the medication. Blood pressure measurements may show steady patterns during the medication, while it was expected that the patient’s blood pressure drops. Combining with other observations, such as the calorie and alcohol intake during this period, may lead to more accurate conclusions.

Anomaly detection in high-dimensional datasets is a double-edged sword, revealing more accurate behaviour of the data as well as posing various challenges. The following factors summarise the underlying challenges [190]:

Concentration of distances — The attribute-wise distances of i.i.d. distributed samples, due to the central limit theorem, approximately converge to a normal distribution. Hence, this results in numerical and parametrisation issues.

Exponential search space — The number of potential feature subspaces grows exponentially with increasing input dimensionality, resulting in an exponential search space.

Data-snooping bias — Every point in a high-dimensional space appears as an anomaly. Given enough alternative subspaces, at least one feature subspace can be found for each point such that it appears as an anomaly.

Irrelevant features — A high proportion of irrelevant features effectively creates noise
in the input data, which masks the true anomalies. The challenge is to choose a subspace of the data that highlights the relevant attributes.

The properties of high-dimensional data can negatively influence the ability of statistical models to extract meaningful information. More specifically, the performance of a statistical model is tied to the interrelationship among the number of samples (records), number of features, complexity of the classifier and the variability of outcome measures. Owing to the “curse of dimensionality” phenomenon, statistical learning techniques break down in high dimensions. For example, some naive learning techniques require the number of training records to grow exponentially with the number of dimensions \cite{21}. This rapidly degrades the ability of an algorithm to converge to a true model as data dimensionality increases. Model over-fitting, estimation instability, and local convergence are some of the common problems that arise due to the lack of a sufficient number of training records.

The most significant impact, however, is reflected in traditional statistical methods \cite{190}, as well as 1SVMs. They are simply not designed to cope with the growth in variables associated with each observation. Hence, they experience a considerable drop in the accuracy of anomaly detection, and a rise in memory- and computational- complexity, when trained on large numbers of features. Some machine learning techniques avoid the curse of dimensionality in high-dimensional data analysis through feature selection or feature extraction. Feature selection searches for the most relevant (best fitted) features by evaluating various subsets of the original features, and is applicable to domains where there are many redundant features \cite{127}. Feature extraction is related to dimensionality reduction, and creates new features through a linear or non-linear combination of all the original features \cite{65}. Feature selection approaches can have drawbacks because the search for an optimal subset imposes an additional layer of complexity in the modelling task. The objective of our work is to propose highly efficient and scalable approaches, hence we focus on dimensionality reduction rather than feature selection methods.

The purpose of dimensionality reduction may vary from one application domain to another, and hence different techniques have been designed subject to the domain requirements. In anomaly detection applications, unsupervised techniques are often con-
Background on Privacy-Preserving and Large-Scale Anomaly Detection in PSNs


cidered due to the lack of labelled records. Principal Component Analysis (PCA) is one of the techniques extensively used in the context of anomaly detection \[101,157\]. PCA is a general multivariate statistical projection technique for dimensionality reduction. Under the constraint of orthogonality, PCA identifies principal components as the directions that maximise the variance along each component. In principal component space, anomalies will appear more visible, and they are those records that increase the variance along their respective directions. However, it is widely known that standard PCA is extremely vulnerable to the presence of anomalous records, and even a single noisy sample can deliver arbitrarily skewed results \[106\]. Since obtaining clean data may not always be attainable, much work has been dedicated to make PCA more robust to contaminated samples, e.g., \[34,63,183\]. Another serious issue for PCA is that it fails when facing nonlinear dependencies between sampled variables, i.e., projecting a more complex dataset using linear techniques often yields poor results. Hence, nonlinear techniques such as Kernel PCA (KPCA) \[36,180,184\], and Isomap \[182\] have attracted significant attention.

After close examination of various nonlinear dimension reduction techniques, Lee et al. \[106\] concluded that nonlinear techniques are often more powerful, though in some cases the use of PCA as a preprocessing stage is justifiable. They find that most nonlinear techniques remain more sensitive to the curse of dimensions than linear ones. They may outperform linear techniques, such as PCA, when there are a few tens of dimensions, but not when the data dimensionality is very high. In this case, linear dimensionality reduction may be very useful for “hard” dimensionality reduction, i.e., eliminating the largest number of variables while maintaining the reconstruction error very close to zero. The use of linear methods as a preprocessing step is justified by the fact that most nonlinear methods remain more sensitive to the curse of dimensionality than PCA, due to their more complex model, which involves many parameters to adjust. Subsequently, nonlinear dimensionality reduction can be used as the last step in the data flow.

An alternative approach to nonlinear techniques are deep neural networks, with an architecture composed of multiple layers of parameterised nonlinear modules. There are a range of advantageous properties that have been identified for such deep networks \[21\]: they can learn higher-level features that yield good classification accuracy; and they are
parametric models, whose training time scales linearly with the number of records \[99\]; and they can use unlabelled data to learn from complex and high dimensional datasets. A major limitation of deep networks is that their loss function is non-convex, therefore the model often converges on local minima and there is no guarantee that the global minimum will be found. In addition, the need to specify a large number of hyperparameter settings can make training deep networks challenging.

In summary, a core challenge of anomaly detection that distinguishes it from other classification problems is that anomaly detection algorithms are usually trained without labelled records, i.e., trained in an unsupervised manner. Some unsupervised anomaly detection methods — such as, \(k\)-means clustering, \(k\)-nearest neighbour, 1SVMs — have obtained promising results in various application domains. However, when it comes to anomaly detection in complex and high-dimensional datasets, these methods reveal certain limitations \[17,190\]. In contrast, deep neural networks, have been demonstrated to be advantageous for such datasets. They are capable of learning a wide variety of underlying distributions of training data with a training time that scales linearly in the number of records \[99\]. Moreover, they can be resilient to small perturbations of the underlying normal data distribution \[21\], which could be advantageous for anomaly detection as we discuss in Part I. These capabilities are due to their architecture and the representation that deep models generate. The next section gives an overview of deep networks and compares them with shallow approaches.

### 2.3.4 Shallow and Deep Architectures

Classification techniques with shallow architecture typically comprise an input layer together with a single layer of processing. Kernel machines such as SVMs, for example, are a layer of kernel functions that are applied to the input, followed by a linear combination of the kernel outputs \[16\]. In contrast, deep architectures are composed of several layers of nonlinear processing nodes. A widely used form of the latter architectures are multi-layer neural networks with multiple hidden layers \[19\].

While shallow architectures offer important advantages when optimising the parameters of the model, such as using convex loss functions, they suffer from limitations in
terms of providing an efficient representation for certain types function families [16]. In particular, shallow models have difficulties in representing functions that capture dependencies in joint distributions, i.e., dependencies between multiple variables can be difficult to capture using a shallow architecture. They can also be inefficient in terms of the required number of computational elements and training examples. Consequently, a shallow architecture can result in a model that does not generalise well, unless trained with a very large number of examples and implemented with a sufficiently large number of nodes, thus increasing the computational resources required [21].

Deep architectures can include multiple layers of representation and abstraction to help model and generalise from complex datasets. This enables deep architectures to provide a compact representation for a much wider range of functions than is possible using shallow ones [19]. In general, functions with a $k$-layer architecture can provide a compact representation using a number of hidden units that is polynomial in the number of input features, whereas a $(k - 1)$-layer architecture requires an exponentially large number of hidden units. Furthermore, these units are generally organised in multiple layers so that many levels of computation can be composed. A thorough description of shallow and deep architectures is presented in [21], and the interested reader is referred to this paper for further information.

### 2.3.5 Summary

The large volume of high-dimensional records that can be contributed in participatory networks question the utility of traditional anomaly detection techniques in such domains. High-dimensional datasets impose extra challenges for traditional anomaly detection. To make use of these approaches in the PSN domain, they should be combined with dimensionality reduction techniques. Hence, the open question is how to overcome the limitations of traditional anomaly detection techniques and scale them for analysing large-scale, high-dimensional datasets. In Part II, we further study the challenges associated with high-dimensional anomaly detection. More specifically, with a focus on 1SVMs, we explore which approaches to dimensionality reduction are more effective and what are other approaches can be used to enhance their performance.
Part I

Privacy-Preserving Anomaly Detection
Introduction to Part I

A central goal in distributed systems involving multiple parties, such as PSNs, is information sharing. The motivation behind data sharing is clear, but an immediate concern is the privacy of participating parties, given the personal nature of sampled data in PSNs. How to address the privacy issues that are raised in this context depends on the type of query being queried. In this part of the thesis we consider two classes of queries — aggregation and data mining queries, and over the next four chapters we present our solutions to privacy-preserving data mining in this context.

Chapter 3 investigates privacy issues concerning aggregation queries, and presents a privacy-preserving approach over four stages. Chapters 4 to 6 investigate privacy issues concerning data mining queries. Chapter 4 presents a generic privacy approach for anomaly detection in a collaborative learning architecture. Chapter 5 enhances the approach from Chapter 4 and provides a detailed analysis to support its effectiveness. Chapter 6 presents a privacy-preserving approach that enables the use of individualised data randomisation methods without impacting the accuracy of the final anomaly detection model.
Chapter 3
KSSI: Privacy-Preserving Data Aggregation in Participatory Sensing Networks

Participatory sensing using mobile devices is emerging as a promising method for large-scale data sampling. A critical challenge for participatory sensing is how to preserve the privacy of individual contributors’ data. In addition, the integrity of the data aggregation is vital to ensure the acceptance of the participatory sensing model by the participants. Existing approaches to these issues suffer from excessive communication cost, long delays or rely on a trusted third-party. The objective of this chapter is to design a data aggregation scheme for participatory sensing systems that addresses user privacy and data integrity while keeping communication overhead as low as possible. We propose four approaches based on data slicing and participants’ mutual coordination, where each approach adds additional protection to help satisfy the data privacy and integrity requirements. In particular, our last approach is able to minimise collusion attacks and verify whether the aggregated data is contaminated (e.g., by malicious participants).

The publication arising from the work in this chapter is paper P1.

3.1 Introduction

In many participatory applications, aggregation queries allow the system to aggregate individuals’ measurements in a way that provides exact summary results while hiding personal information. An open challenge in this context is how to aggregate raw data from users when the aggregator is untrusted. A practical solution for this issue has been proposed by Shi et al. [156], in which users’ privacy is preserved based on a mutual protection approach called data slicing. In this approach, sensed values are split into “slices”
and distributed among neighbours before being transmitted to the server. Acting as intermediate aggregators, neighbours partially aggregate the received data slices and then forward the result to the server. While this scheme ensures that captured sensor measurements are likely to remain private, there is potential for the intermediate aggregators to make inferences about their neighbours or the whole network. For example, consider the case of a group of participants who are on a low-cal diet. Even receiving a slice of measured calories with a greater value than the defined limit reveals that the corresponding neighbour did not follow the diet plan.

A further challenge that arises from sharing data slices according to this scheme is that there is no way to guarantee that the collected sensor readings are trustworthy. If malicious users in the network modify other participants’ data, then the integrity of the system can be compromised. In addition, factors like delay or collisions over wireless network channels may cause messages to be lost or corrupted, thus degrading the integrity of the aggregation results from the PSN.

To the best of our knowledge, the latter issue, data integrity, has not been addressed in the existing research on aggregation in PSNs. The lack of a comprehensive method for data aggregation in PSNs that simultaneously ensures user privacy and integrity motivates our work.

To address the above-mentioned issues, we adapt a secret perturbation scheme [37] for use in the PSN architecture, shown in Figure 3.1, to achieve the following objectives.

- **User Privacy**: We require that the sample measurements of an active user are not revealed to any other entity in the network. A similar requirement applies to fully or partially aggregated data, that they should only be accessible to the aggregation server.

- **Data Integrity**: We require that the aggregated value at the aggregation server should be equal to the sum of the original data sensed by the participating active users.

- **Efficiency**: While privacy and integrity are requirements of data aggregation, we also require that the system complexity should be kept as low as possible.
This chapter is organised as follows. Section 3.2 summarises related work. Section 3.4 presents our proposed approaches, starting with a basic approach and gradually optimising it to meet our requirements. Section 3.5 introduces two measures to evaluate the effectiveness of our approaches in terms of privacy and communication cost. Section 3.6 demonstrates the empirical results of the proposed approaches, based on the introduced measures. Section 3.7 concludes the chapter and presents some ideas for future research.

3.2 Related Work

Since the idea of participatory sensing was first introduced in [30], it has recently become an active topic of research. A broad overview on the importance of PSNs, their challenges and opportunities are illustrated in [32, 44, 159]. Two major issues for data aggregation in participatory systems are the privacy of personal information and data accuracy. Although some similar issues have been addressed in WSNs [37, 85, 136], participatory sensing systems inherit a complex communication environment. Accordingly, with dynamic architectures, untrusted aggregators and the personal content of queries, PSNs cannot directly employ such solutions.

Privacy - In contrast to the large number of studies on privacy-preserving data aggregation conducted in WSNs, it is largely an open problem in PSNs.

The closest works to our focus on privacy-preserving data aggregation are [156] in participatory networks, and [37, 85] in WSNs. In SMART [85], user privacy is addressed by slicing each data measurement randomly and relying on neighbours to transmit data slices. Recently, PriSense [156] applied SMART to PSNs, and studied the effectiveness of neighbour selection for data sharing.

A promising approach for secure data aggregation is CMT [37], which is an efficient and provably secure additive homomorphic stream cipher that provides effective encrypted data aggregation. However, there are some disadvantages associated with CMT. First, active users need to send their non-aggregatable ID to the server, thus increasing the communication cost. Second, the additive homomorphic property of CMT means
that a malicious participant can simply add fraudulent data to the content of a message. Finally, it does not prevent a malicious server seeking to disclose participants’ records.

A number of studies have attempted to address some of these deficiencies of the CMT secret perturbation scheme. For example, in [132] to avoid sending user IDs, it is assumed that all the users are participating. Li and Cao [108] applied CMT to mobile sensing networks with a dynamic architecture and a malicious server. However, the authors rely on a TTP for assigning secrets to users, which may be a restrictive assumption in real world scenarios.

**Privacy and Integrity** - Privacy and integrity are still open challenges for PSNs. The authors in [132] adopt a secret sharing scheme for integrity assurance, in which the system is required to have background knowledge about the users that are willing to contribute. This might be a feasible assumption in WSNs with a more static environment, and where it can be assumed that all users are participating. However, these assumptions are too restrictive for PSNs.

### 3.3 Preliminaries and Problem Statement

In this section, we outline the network architecture, query model and security assumptions for our problem statement.

#### 3.3.1 Participatory Sensing Architecture

We consider the case of a participatory sensing network that comprises an aggregation server $S$ and a set of participating mobile user $U = \{u_i | i = 1, \ldots, U\}$, as can be seen in Figure 3.1a. Similar to the architecture in [189], we assume that the server $S$ transmits packets via one hop to the $U$ users, and the users (e.g., using WiFi or cellular) can communicate directly with $S$. Users within the network can also form a wireless ad hoc network with each other, so that they can communicate with their neighbours, for example, using the Ad hoc On-demand Distance Vector (AODV) protocol [133] for route discovery. For ease of understanding, in this chapter we assume the network has a flat architecture. However, our schemes can be applied to hierarchical architectures as well.
3.3 Preliminaries and Problem Statement

(a) PSN architecture.

(b) BSP architecture.

(c) UPS architecture.

(d) KSS architecture.

Figure 3.1: Architectures of PSN and aggregation schemes.
Without loss of generality, we describe the case of a single measurement variable at each user. The server $S$ issues a query $Q$ to the users $U$, and computes the aggregation value of the measurement values returned by those users who choose to respond to the query or have a valid value satisfying $Q$. Let $AU \in U$ denote the set of users that report to the query from $S$, such that $AU = \{ au_l | l = 1, \ldots, AU \}$, and let $X_l$ denote the measurement value returned by $au_l$, where $X_l$ is an integer.

### 3.3.2 Aggregation Query Model

In this chapter, we consider the case of the widely used exact summation query, where $S$ is to compute $X_{SUM} = \sum_{l=1}^{AU} X_l$. We also expect that $S$ is able to determine the number of active users $AU$ who have provided their measurement value. In this way, the participatory network is able to compute a range of queries, such as $MEAN$, $AVERAGE$ or $STANDARD DEVIATION$.

### 3.3.3 Security Model

Our aim is to ensure the integrity of the data aggregation process in response to a query, while ensuring the privacy of the active users. In particular, our aim to minimise the possibility that an active user $au_l$ can be associated with his measurement data $X_l$ by any third-party eavesdropper, $S$ or any other user $u_i \in U \setminus \{ au_l \}$. In this section we summarise the main security assumptions we have made and the type of threat model that we address.

**Security Assumptions**

Each user $u_i$ is assigned an identifier $ID_i$ and a private key $K_i$ that is shared with $S$. Each pair of neighbouring users can derive their own symmetric key, so that any communication between that pair of users can be encrypted to prevent eavesdropping attacks. The allocated key is temporal and is valid as long as the user is in the network (or active). Each user is also given a list of all available neighbouring nodes in the network, with which it can communicate, following the approach proposed in [189].
3.4 Our Schemes for Privacy Preservation

Threat Model

We consider that no entity can be guaranteed to be trustworthy, including both $S$ and the users. In the case of $S$, we consider it to be an “honest-but-curious” adversary, i.e., it is interested in the correct aggregation results, but may want to violate the privacy of an active user $au_l$ by being able to associate a user with his measurement data $X_l$. In the case of a user, we assume he can exhibit two types of malicious behaviour:

1. The user can try to violate the privacy of a neighbour by attempting to obtain the measurement data $X_l$ collected from a neighbour $au_l$.

2. If used to aggregate values from other users, a malicious user may want to manipulate the result of aggregation to produce an incorrect result.

3.4 Our Schemes for Privacy Preservation

To achieve privacy for exact sum queries in participatory networks, we adopt the basic ideas of secret perturbation [37] and data splitting [85]. In the following four subsections, we build different schemes based on these basic ideas to meet our desired objectives for PSNs. We refer to these schemes as: (i) Basic Secret Perturbation Scheme (BSP), (ii) Universal Perturbation Scheme (UPS), (iii) Key Splitting Scheme (KSS), and (iv) Key Splitting Scheme with Integrity (KSSI).

3.4.1 Basic Secret Perturbation Scheme (BSP)

The idea of secret perturbation [37] is based on additive homomorphic encryption. Let $X \in [0, I - 1]$ denote a measurement value, where $I$ is a large integer, and let $K$ denote a key. The encryption of $X$ using $K$ gives a ciphertext

$$\mathcal{Y} = Enc(X, K, I) = (X + K) \mod I,$$

(3.1)
where the decryption gives

\[ X = Dec(Y, K) = (Y - K) \mod I. \]  \hspace{1cm} (3.2)

Using this scheme, it can be shown that if \( Y_1 = Enc(X_1, K_1, I) \) and \( Y_2 = (X_2, K_2, I) \) then \( Dec(Y_1 + Y_2, K_1 + K_2, I) = X_1 + X_2 \), i.e., the encryption is homomorphic with respect to addition.

We can use secret perturbation in participatory sensing as follows. Each active user \( au_l \) has a measurement value \( X_l \) and a secret key \( K_l \) that is shared with the aggregation server \( S \). The user \( au_l \) also has an identifier \( ID_l \) that is assigned by \( S \). If a user wants to participate in a query it takes the following steps:

\begin{itemize}
  \item[i.] Encrypt the observation \( Y_l = (X_l + K_l) \mod I \).
  \item[ii.] Notify the server \( S \) of the participation of \( au_l \) by sending \( ID_l \) directly from \( au_l \) to \( S \).
  \item[iii.] Send \( Y_l \) to a neighbour node of \( au_l \) for aggregation.
\end{itemize}

An active user selects a neighbour randomly from the list of available neighbours and forwards its data via the neighbour to \( S \). We refer to these randomly chosen neighbours as a Cover Node (CN) \cite{156} of \( au_l \). Although all users can communicate directly to \( S \), a cover node helps to protect the user’s privacy. As shown in Figure 3.1b, when a user \( u_3 \) receives perturbed observations (say, \( Y_1 \) and \( Y_2 \)) from its neighbours (\( au_1 \) and \( au_2 \)), it transmits the summation \( Y_1 + Y_2 \) to the aggregation server \( S \). Thus, \( S \) receives \( Y_1 + Y_2 \) and separately \( ID_1 \) and \( ID_2 \), from which it infers that it must use \( K_1 \) and \( K_2 \) in decryption. The final aggregation \( X_1 + X_2 \) can then be determined as \( X_1 + X_2 = Dec(Y_1 + Y_2, K_1 + K_2, I) \).

Note that by sending the \( Y_l \) values in aggregated form to the server, we ensure that it does not have the ability to decrypt individual data observations. In addition, the cover nodes are unable to decrypt the encrypted values \( Y_l \).

**Potential drawbacks:** Applying secret perturbation to PSNs enhances data secrecy by concealing measurement values from cover nodes. However, it suffers from the following issues:
3.4 Our Schemes for Privacy Preservation

Privacy — Note that $S$ could be malicious in PSNs. Although the cover nodes receive encrypted values, the scheme is vulnerable to a violation of privacy if there is collusion between an aggregating cover node and $S$.

Efficiency — BSP requires each active user to inform $S$ about their participation in order to calculate the decryption key. This is accomplished by forwarding the active user’s ID. However, it imposes extra communication overhead, especially when the number of participants is relatively large.

Integrity — Data may become corrupted either due to the weakness of homomorphic aggregation, where one can add any artificial message, or accidental causes such as a user leaving before conveying received messages to $S$.

In the following, first we seek to address the issues of privacy and efficiency, then we address the problem of integrity.

3.4.2 Universal Participation Scheme (UPS)

In order to overcome the issue of privacy in BSP, we use the approach of data splitting in combination with secret perturbation to prevent any single user having all the data from an active user. Moreover, our UPS scheme avoids the need to transmit the IDs of the participating active users by having all active users contribute a value, which can be zero if the active user has no valid measurement value to contribute.

Inspired by the fact that relying on a single cover node is not a significant challenge for malicious entities to expose measurements, as suggested in [85], we apply a splitting technique to augment users’ privacy. Without loss of generality, we express measurements and keys as a function of $X_l = f(x_{i,j})$ and $K_l = f(k_{l,j})$, for $j = 1, \ldots, c$, where $c < U$. $f$ is an additive function under which $X_l \in \mathbb{N}$ is sliced randomly into $c$ number of pieces subject to $X_l = \sum_{j=1}^{c} x_{i,j}$, $x_{i,j} \in \mathbb{N}$.

Let $CN_l$ be a set of cover nodes selected at random by $au_l$. Each active user $au_l$ takes the following steps:

i. Slice the data and key as $X_l = \sum_{j=1}^{c} x_{i,j}$ and $K_l = \sum_{j=1}^{c} k_{l,j}$, for $j = 1, \ldots, c$, and generate $Y_{l,j} = (x_{i,j} + k_{l,j}) \mod I$. 

ii. Select \( c \) number of cover nodes and send each a \( Y_{i,j} \).

The selected cover nodes, similar to the previous scheme, submit the aggregated ciphertext, e.g., \( Y_{1,2} + Y_{2,2} \) to \( S \). With the assumption that all users have contributed their data, on receiving partiality aggregated values from the cover nodes, the aggregation server \( S \) performs the following steps:

\( i. \) Calculate the sum of the ciphertexts

\[
\sum_{l=1}^{AU} \sum_{j=1}^{c} Y_{l,j} = Y_{1,1} + \ldots + Y_{AU,c}
\]

\( ii. \) Decrypt the aggregated value \( X_{sum} \)

\[
\sum_{l=1}^{AU} X_l = \left( \sum_{l=1}^{AU} \sum_{j=1}^{c} Y_{l,j} - \sum_{l=1}^{AU} K_l \right) \mod T.
\]

Due to slicing, the absolute data \( X_l \) remains secret unless \( S \) as well as all of the cover nodes selected by \( au_l \) collude.

**Potential drawbacks:** While this scheme is less vulnerable to collusion due to the use of data splitting, it has two potential drawbacks of its own.

*Erroneous aggregation* — This scheme has more risk of producing an erroneous result, with all the \( U \) users each submitting \( c \) slices, if any of the active users fails to contribute a value, or if a value is lost in the transmission.

*Communication overhead* — If the number of active users with a valid measurement is small compared to the total number of users, then there is a high communication overhead. Though having all users contribute data is a common solution [108], such an assumption may not be realistic for PSNs with highly dynamic membership.

### 3.4.3 Key Splitting Scheme (KSS)

In order to address the potential drawbacks of the previous two schemes, we propose a variation on data splitting with secret perturbation. This new scheme uses a random key from each active user, rather than the shared key of the server \( S \).
Users with an observation value to contribute generate a random integer $\tilde{K}$ and use this value to perturb their data. The $\tilde{K}$ can be generated by using a pseudo-random function, like HMAC, taking the $K$ and an arbitrary string. The advantage of perturbing sensor measurements with $\tilde{K}$, instead of the shared key $K$, is that these values can be aggregated and active users are not required to inform the server $S$ by sending their IDs.

On receiving a query, $au_l$ generates $Y_l = (X_l + \tilde{K}_l) \mod I$ and forwards it directly to $S$, while the implemented $\tilde{K}$ can be transmitted via a set of $c$ randomly chosen cover nodes as $\tilde{K}_l = \sum_{j=1}^{c} \tilde{k}_{l,j}$, using the random slicing technique. In this way, a malicious $\tilde{S}$ cannot disclose the perturbed data $Y_l$ unless all the cover nodes $CN_l$ are malicious.

Receiving the ciphertexts and the partially aggregated keys, $S$ takes the following steps:

i. Calculate the sum of the perturbed values and the keys

\[
\sum_{l=1}^{AU} Y_l = (X_1 + \ldots + X_{AU} + \tilde{K}_1 + \ldots + \tilde{K}_{AU}) \mod I,
\]

\[
\sum_{l=1}^{AU} \sum_{j=1}^{c} \tilde{k}_{l,j} = (\tilde{k}_{l,1} + \ldots + \tilde{k}_{AU,c}) \mod I.
\]

ii. Decrypt the aggregated data $X_{sum}$

\[
\sum_{l=1}^{AU} X_l = (\sum_{l=1}^{AU} Y_l - (\sum_{l=1}^{AU} \sum_{j=1}^{c} \tilde{k}_{l,j})) \mod I.
\]

Potential drawbacks: While our KSS scheme keeps perturbed values out of the reach of malicious neighbours, still they can falsify the key slices $\tilde{k}$. Consequently, we propose the following approach to address this issue.

3.4.4 Key Splitting Scheme with Integrity (KSSI)

Having devised an efficient privacy-preserving data aggregation scheme, we now address the last issue, data integrity. As discussed in Section 3.4.1, transmitted data, whether it is a data slice of an observation value or a key, can be tampered with by a malicious
user or corrupted unintentionally. For either reason, falsified data degrades system accuracy and reliability. Consequently, we equip $S$ with an integrity check to detect distorted results. We propose a secure homomorphic Message Authentication Code (MAC), and use KSS as our underlying scheme, although our integrity check can be applied to the other approaches as well.

The proposed MAC is based on the discrete logarithm, and its homomorphic property allows $S$ to verify the integrity of aggregated keys. Let $g$ be a generator of a multiplicative cyclic group $G_q$ of prime order $q$ and $z$ be a large prime number. The server $S$ circulates $g$ and $z$ to all users joining the network as public values. An active user, $au_l$, generates the MAC of its randomly generated key as $MAC(\tilde{K}_l, g) = g^{\tilde{K}_l} \mod z$ and sends it to $S$. This MAC has the homomorphic property \cite{76} since $MAC(\tilde{K}_1, g) \times MAC(\tilde{K}_2, g) = g^{\tilde{K}_1 + \tilde{K}_2} \mod z$.

Following KSS, $S$ receives partially aggregated key slices and calculates $\sum_{l=1}^{AU} \sum_{j=1}^{c} \tilde{k}_{l,j}$. The integrity of aggregated keys can be checked against the $MAC(\tilde{K}, g)$ values sent by the active users. To do this, $S$ takes the following steps:

\begin{enumerate}
  \item Aggregate the MAC values
  \[ MAC(\sum_{l=1}^{AU} \tilde{K}_l, g) = MAC(\tilde{K}_1, g) \times \ldots \times MAC(\tilde{K}_{AU}, g). \]

  \item Generate the MAC of the aggregated keys
  \[ MAC'(\sum_{l=1}^{AU} \sum_{j=1}^{c} \tilde{k}_{l,j}, g) = g^{k_{1,1} + \ldots + k_{AU,c}} \mod z. \]
\end{enumerate}

Any inconsistency between MAC and $MAC'$ implies that the keys have been corrupted.

### 3.5 Performance Analysis

In the following, we analyse the performance of the aforementioned schemes using the measures introduced in \cite{156}.
- **Hidden Probability** ($H_p$): the probability that a sensor measurement, $X_l$, remains hidden from malicious entities.

- **Communication Cost** ($T$): the total communication overhead associated with aggregated sensor measurements in response to a query.

### 3.5.1 Hidden Probability

Hidden probability measures the likelihood that a data observation cannot be associated with its corresponding active user. Since both data and keys are shared between cover nodes and the aggregation server $S$, this metric can be defined in terms of the number of malicious users in the network. In this chapter we use $\hat{\sim}$ as a symbol for malicious entities and we assume an average of $\hat{A}$ out of $A$ servers and $\hat{U}$ out of $U$ users are malicious.

To overcome the privacy issues in PSNs we have used a splitting technique. Here we study the impact of data splitting (as in UPS and KSS) on hidden probability compared to the non-splitting approach (BSP).

**Non-Splitting:** Since $S$ can also be malicious in PSNs, schemes like BSP that forward their measurements via other users fall short of providing robust privacy. If the selected neighbour is malicious it can collude with $\hat{S}$ and disclose the data. Recall our assumption that on average, $\hat{U}$ out of $U$ users are malicious, then the probability that $au_i$ selects a malicious cover node is equal to $\frac{\hat{A}}{A}$. Moreover, if all the active users except for $au_i$ are malicious, a malicious $\hat{S}$ can collude with them, to breach the users’s privacy by revealing their records. Suppose $\Gamma(\Psi)$ is the event that $\Psi$ is a malicious entity, then the hidden probability $H_p$ for BSP is estimated as\(^1\)

\[
H_p = 1 - p(\Gamma(S) \cap (\Gamma(CN_l) \cup \Gamma(AU\backslash\{i\})))
= 1 - \left(\frac{\hat{A}}{A} p(\Gamma(cn)) + \frac{\hat{A}}{A} \prod_{u \in AU \backslash \{i\}} p(\Gamma(au))\right)
= 1 - \left(\frac{\hat{A}}{A} \left(\frac{\hat{U}}{U}\right) + \frac{\hat{A}}{A} \left(\frac{\hat{U}}{U}\right)^{AU-1}\right)
\]

\(^1\)For ease of estimating $H_p$, we ignore the impact of those active users selected by $au_i$ as a cover node, which also overlap with $CN_l$.\]
Splitting: Under UPS and KSS, active users share their perturbed data and key with their neighbours, respectively. Sharing makes data disclosure more difficult for $\hat{S}$, as it requires collaboration of all cover nodes $\mathcal{CN}_i$ or $\mathcal{AU}\{i\}$ to obtain $X_i$, otherwise the data remains secret. Therefore, the hidden probability for UPS can be estimated as:

$$H_p = 1 - p(\Gamma(A) \cap (\Gamma(\mathcal{CN}_i) \cup \Gamma(\mathcal{AU}\{i\})))$$

$$= 1 - \left( \frac{\hat{A}}{A} \left( \frac{\hat{U}}{U} \right)^c + \frac{\hat{A}}{A} \left( \frac{\hat{U}}{U} \right)^{AU-1} \right)$$

(3.4)

3.5.2 Communication Overhead

For each of the aforementioned schemes, we measure the communication overhead associated with each user’s interaction separately. Let $T_{AU-CN}$ and $T_{AU-S}$ denote the communication cost incurred due to sending data from active users to their cover nodes and $S$, respectively, and $T_{CN-S}$ denote the cost incurred due to transmitting data from selected cover nodes to $S$. Finally, the total communication cost in each scheme is obtained as $T = T_{AU-CN} + T_{AU-S} + T_{CN-S}$. The communication cost $T$ for each proposed approach is calculated as follows.

• BSP

Usually in PSNs not all users respond to each query. Using BSP, $S$ needs to be informed about the involved active users to derive the decryption key, and it is accomplished by forwarding active users’ identities.

We first estimate $T_{AU-CN}$, corresponding to the cost incurred by route discovery to a cover node, in addition to the cost of unicasting messages. We assume the average cost associated with route discovery and the route reply is $\varphi$ bits. Let $\ell_X$ be the average length of a message in bits, and let $AU$ denote number of active users, then $T_{AU-CN} = AU \times (\ell_X + \varphi)$.

Assuming that devices are capable of direct communication, and $\ell_{ID}$ is the length of an ID in bits, then $T_{CN-S}$ is obtained as $T_{AU-S} = AU \times \ell_{ID}$. However, our scheme is flexible enough to be extended to hierarchical architectures as well.
Next we model the overhead $T_{CN-S}$ that results from a set of cover nodes transmitting the aggregation of data and IDs. Let $N_{cn}$ be the expected number of users selected as cover nodes. A user $u_i \in U$ is selected as a cover node by an active user with a probability of $1/(U - 1)$. Therefore, the probability of the user $u_i$ being chosen by at least one active user is $1 - (1 - \frac{U - 2}{U - 1})^{AU}$, and $N_{cn}$ is calculated as $N_{cn} = U \times (1 - (1 - \frac{U - 2}{U - 1})^{AU})$. Hence, the cost incurred from cover nodes communicating to $S$ can be calculated as $T_{CN-S} = N_{cn} \times \ell_X$.

Using the BSP scheme, the risk of privacy violation potentially increases, as well as the overhead incurred due to the transmission of all user IDs in large communities.

**UPS**

In order to avoid the communication overhead of active users sending their IDs in BSP, the UPS scheme requires every user $u_i$, whether or not $u_i \in AU$, to reply to the disseminated queries. Accordingly, $N_{cn}$ can be estimated as $N_{cn} = U \times (1 - (1 - \frac{U - c - 1}{U - 1})^{AU})$.

Since users are not required to pass their IDs to $S$ the total communication cost includes $T_{AU-CN} = U \times c \times (\ell_X + \varphi)$ and $T_{CN-S} = N_{cn} \times \ell_X$. Although all the users need to send some form of message, this scheme is beneficial when the majority of users have data to submit.

**KSS**

To estimate the communication cost of the KSS scheme where active users directly send their perturbed values $Y$ to $S$, the incurred $T_{AU-S} = AU \times \ell_X$.

Let $\ell_{\tilde{K}}$ be the size of generated key $\tilde{K}$, which should be at least the same size as $X$ to ensure adequate secrecy [125]. So the $T_{AU-CN}$ is calculated as $T_{AU-CN} = AU \times c \times (\ell_{\tilde{K}} + \varphi)$.

The sliced keys are integer values and since data overflow may occur due to data aggregation, we include a carry-bit header and estimate its length as $H_{cb} = \log N_{au}$. Let $N_{au}$ be the expected number of active users choosing a user as a cover node, then $N_{au} = AU \left(\frac{c}{U-1}\right)$ and $N_{cn} = N \times (1 - (1 - \frac{U - c - 1}{U - 1})^{AU})$. Therefore, $T_{CN-S}$ is obtained as $T_{CN-S} = N_{cn} \times (\ell_{\tilde{K}} + H_{cb})$. 
The additional option of an integrity check comes with a substantial communication overhead, which can be considered as the trade-off for improved accuracy. Let $\ell_{\tilde{KMAC}}$ be the size of $\tilde{K}$ chosen for the KSSI scheme and $\ell_{MAC}$ be the size of the MAC, then $T_{AU-S} = AU \times (\ell_{MAC} + \ell_X)$. $T_{AU-CN}$ and $T_{CN-S}$ are essentially the same as in KSS, except with a key size of $\ell_{\tilde{KMAC}}$ bits (n.b., $|\ell_{\tilde{KMAC}}| \geq 180$ bits).

### 3.6 Evaluation Results

In this section, we evaluate the performance of the proposed schemes in terms of their hidden probability and communication overhead based on our analytical model and empirical results. For this purpose, we analysed the number of malicious users $\hat{U}$, active users $AU$ and data slices $c$ on the performance of a network with 100 participants and an aggregation server. Table 3.1 shows the other default parameter values. The main parameters here are the number of cover nodes $c$ and the number (or percentage) of anomalous nodes, and we explore their impacts on our algorithm in the following empirical analysis.
### 3.6 Evaluation Results

#### Table 3.1: Default Evaluation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>AU</th>
<th>$\hat{U}$</th>
<th>$c$</th>
<th>$\ell_X$</th>
<th>$\ell_{MAC}$</th>
<th>$\ell_{\hat{K}}$</th>
<th>$\ell_{\hat{K}MAC}$</th>
<th>$\ell_{ID}$</th>
<th>$\varphi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>50</td>
<td>50</td>
<td>2</td>
<td>12</td>
<td>180</td>
<td>7</td>
<td>180</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

![Graphs showing communication cost](image)

(a) Total communication cost  
(b) Cover nodes communication cost

**Figure 3.3:** Communication cost under the default settings.

### 3.6.1 Hidden Probability

Figure 3.2 illustrates the effect of varying the number of cover nodes $c$ and malicious users $\hat{U}$ on hidden probability, when $S$ is malicious. It confirms that selecting a greater number of cover nodes and sharing information enhances the hidden probability, regardless of the number of malicious users. As can be seen in Figure 3.2, when $c = 1$, i.e., non-splitting, the hidden probability is much lower, and thus is more vulnerable. As the number of malicious users $\hat{U}$ increases, so does the probability of data exposure. This figure shows how sensitive user privacy is to the number of malicious participants. Although the increase in $\hat{U}$ gradually impacts the resistance of splitting techniques, it is more significant in the case of non-splitting. For example, when about half the users are malicious, the risk of violating the privacy of an active user exceeds 50%. However, with two or more cover nodes each user can have a confidence of privacy of at least 75%.
3.6.2 Communication Overhead

We use the analysis of Section 3.5.2 to calculate the total communication overhead for each scheme as well as the cost imposed on cover nodes as a result of forwarding their neighbours’ data. In terms of communication overhead, the following three factors play the main role in defining the network communication cost: $U$, $AU$ and $c$. Accordingly, we examine our scheme efficiency in terms of these factors. Figure 3.3 shows the communication overhead for the BSP, UPS and KSS schemes under the default settings. As expected, the cost of UPS is independent of the number of active users, while the other
two schemes experience a linear increase with the growth of \( AU \). The higher overhead of the \( UPS \) scheme implies that it is not an efficient solution for those networks in which only a small proportion of users are active users.

Figure 3.4 illustrates the impact of the number of cover nodes on communication overhead. Doubling the number of cover nodes (from \( c = 2 \) in Figure 3.3 to \( c = 4 \) in Figure 3.4), nearly doubles the transmission cost for \( UPS \) and \( KSS \), which can be considered as the trade-off for privacy. Figure 3.5 shows the overhead incurred by the integrity check. Though including the option of integrity imposes a greater burden on the system (especially on the cover nodes), this is the cost of accuracy.

### 3.7 Discussion and Conclusion

In this chapter we presented four novel schemes to address the problems of data privacy and integrity for aggregation queries in participatory sensing. The proposed approaches are designed to fulfil different levels of privacy regarding the proportion of malicious entities in the network. Obliviously, higher levels of privacy require the cooperation of more users and incur a higher communication cost. The simulation results and analytical models show that our approaches ensure user privacy with high probability, while detecting a loss of integrity. However, the privacy is achieved by relying on the cooperation of other participating nodes (cover nodes) in the network.

More specifically, from the empirical analysis we can conclude that the level of privacy and efficiency of our proposed methods are tied to the number of selected cover nodes. Higher levels of privacy are achieved by involving more cover nodes, however, this imposes higher communication cost as well. The results show that with four cover nodes the privacy of participants is preserved with high probability, even if 50% of users are malicious. This also indicates that the communication cost of the participants and cover nodes roughly increases by the factor of four.

The proposed data aggregation architecture can be applied to online applications. Then following the algorithm introduced in Section 3.4.3 (or Section 3.4.4) the participants can send the captured measurements one at a time, or send the sum of (a certain number
of) measurements to the server.

In the next chapter we study the privacy-issues in anomaly detection queries, and propose various solutions over the next three chapters.
Chapter 4

RMP: Privacy-Preserving Collaborative Anomaly Detection for Participatory Sensing

In this chapter we address the problem of collaborative anomaly detection, where multiple data sources submit their data to an online service in order to detect anomalies with respect to the wider population. A major challenge is how to achieve reasonable detection accuracy without disclosing the actual values of the participants’ data. We propose a lightweight and scalable privacy-preserving collaborative anomaly detection scheme called Random Multiparty Perturbation (RMP), which uses a combination of nonlinear and participant-specific linear perturbations. Each participant uses a randomly perturbed uniformly distributed random matrix, in contrast to existing approaches that use a common random matrix for all participants. Each participant’s transformation matrix is derived from a public matrix and additive noise. The experimental results suggest that by imposing a reasonably small level of noise RMP delivers acceptable privacy and accuracy. RMP is designed so that it does not rely on the presence of a TTP, so that it can be used in participatory sensing architectures that lack a TTP.

This chapter provides the foundations for Chapters 5 and 6. We introduce the basic problem statement, motivation, challenges and give an elementary solution. In the later chapters, we improve our solution and provide rigorous theoretical and experimental analysis of its privacy and accuracy.

The publication arising from the work in this chapter is paper P2.

4.1 Introduction

Anomaly detection or outlier detection aims to identify unusual values in a given dataset. In particular, there is growing interest in collaborative anomaly detection, where multiple
data sources submit their data to an online data mining service, in order to detect anomalies with respect to the wider population. By combining data from multiple sources, collaborative anomaly detection aims to improve detection accuracy through the construction of a more robust model of normal behaviour [27][51] (note that by “normal” we mean non-anomalous, unless otherwise stated). Cloud-based collaborative architectures such as participatory sensing networks [30][44] provide an open distributed platform that enables participants to share and analyse their local data on a large-scale. One major issue with collaborative anomaly detection is how to ensure the privacy of participants’ data while maintaining the accuracy of anomaly detection [23][55][168].

For example, in public health applications based on participatory sensing using smartphones, anomaly detection can be used to identify emerging symptoms of a disease outbreak. However, users may be reluctant to contribute accurate data if there is a risk that their personal data may be disclosed to other parties, thus breaching their privacy. In this context, a method for preserving the privacy of participants’ data while maintaining both accuracy and scalability is essential for such participatory systems.

In participatory networks, part of the solution to privacy is to provide a secure communication channel between participants and the data mining server in order to prevent eavesdropping attacks. However, this does not prevent data disclosure by the server. Consequently, one also requires a privacy-preserving scheme that masks the raw values of participants’ sensitive attributes. Such a masking scheme should be designed so that the data miner can construct a global model from the participants’ perturbed values. To address this privacy issue most existing works in the literature, e.g., [55][168] are based on Secure Multiparty Computation (SMC) to generate a global model from encrypted local data. While SMC can achieve high levels of privacy and accuracy, SMC based methods impose a high communication and computational overhead. Moreover, they require the simultaneous coordination of all participants during the entire training process, which limits the number of participants in practice.

As discussed in Chapter 2, a promising alternative to SMC is data randomisation, where participants perturb (mask) their records by imposing additive or multiplicative

1Although we focus on cloud-based architectures in this thesis, our approach to privacy-preserving anomaly detection can be applied to a wide range of distributed computation models.
noise on the raw data. In comparison to SMC, randomisation approaches are scalable to large networks of participants and have marginal computational cost. Therefore, randomisation schemes offer a more effective solution for addressing privacy in large collaborative networks. Existing privacy-preserving randomisation approaches in the literature, e.g., \cite{23, 41, 111, 114}, are based on the assumption that all participants share the same perturbation matrix. This assumption is required because the accuracy of traditional data mining algorithms deteriorates if there are even slight variations in the perturbation matrix between participants. The prerequisite of consistency among perturbation matrices may not be feasible in practice, since it makes the schemes potentially vulnerable to breaches of privacy if collusion occurs between rogue participants and the data mining server. Moreover, forcing the system to use a single perturbation matrix restricts the general model to only the primary participants or other parties that have access to the original perturbation matrix. Thus, an open research challenge is how to relax the requirement for a single perturbation matrix throughout the system, while still achieving a high level of accuracy. In particular, we focus on this problem in the context of anomaly detection in participatory sensing applications.

To address this challenge, we propose a privacy-preserving scheme for anomaly detection called Random Multiparty Perturbation. RMP supports the scenario where participants contribute their local data to a public service that trains an anomaly detection model from the combined data. This model can then be distributed to clients who want to test for anomalies in their local data. In this chapter, we make use of an autoencoder as our anomaly detection model, although our scheme is also applicable to other types of anomaly detectors.

In order for the participants of RMP to maintain the privacy of their data, we propose a form of random perturbation to be used by each participant. Previous approaches to random perturbation in this context \cite{23, 41, 114, 123} require all participants to perturb their data in the same way, which makes this scheme potentially vulnerable to breaches of privacy if collusion occurs between rogue participants and the server. In contrast, RMP proposes a scheme in which each participant first perturbs their contributed data using a unique, private random perturbation matrix. In addition, any client can apply the
resulting anomaly detection model to their own local data by using a public perturbation matrix. This provides a scalable collaborative approach to anomaly detection, which ensures a high level of privacy while still achieving a high level of accuracy to the clients.

RMP also offers the following desirable properties:

(i) It maintains participants’ privacy, through a two-stage perturbation, which is a combination of a participant’s individual nonlinear and linear transformations. Although random transformations have been used in conjunction with support vector machines to speed up the computation of kernel functions, its application to privacy-preserving anomaly detection is new and has not been analysed before. The nonlinear function is used to condition the Probability Density Function (pdf) of the data in order to thwart maximum a posteriori (MAP) attacks. The random linear transformation maps the data to a lower dimension space in order to thwart Independent Component Analysis (ICA) and distance inference attacks.

(ii) It is resistant to collusion, since participants individually generate their own random perturbations.

(iii) The randomisation method has low computational complexity. Unlike cryptographic techniques, such as SMC, RMP has very competitive computational complexity.

(iv) The scheme is scalable to large numbers of participants on existing hardware platforms.

(v) It imposes no restrictions on the kind or the number of clients accessing the anomaly detection function generated by RMP, i.e., the model can be used by any party who accesses the transformation matrix.

This chapter is organised as follows. Section 4.2 first elaborates the problem statement and design principles, then presents our proposed scheme, based on two architectures. Section 4.3 evaluates the impact of our two-stage transformation on accuracy using various benchmark datasets. Section 4.4 summarises this chapter and offers some ideas for future work.

2For the sake of brevity, we only introduce existing attacks to data randomisation in this chapter, and the detailed analysis of the attacks is postponed to the next chapter.
4.2 RMP Privacy-Preserving Anomaly Detection Scheme

In this section we present our Random Multiparty Perturbation (RMP) scheme, based on two architectures. Basic RMP considers a general collaborative learning architecture comprising three types of parties, namely, users, the data miner server and clients, as discussed in Section 2.1 and mediated RMP is designed for architectures including a TTP. Note that in this thesis we are not relying on a TTP, so the mediated scheme is just presented for the sake of comparison and to show how RMP can be adapted to such architectures.

4.2.1 Problem Statement and Design Principles

In the classical participatory sensing scenario [30], an initiator creates a data crowdsourcing campaign; a group of users $\mathcal{U}$ contribute opportunistically sensed data; and a server $S$ analyses the data for valuable information. An initiator could be the mining server, or an external entity like the application that raises the query. Since the design of our privacy-preserving scheme is independent of the role of the initiator, we do not include it into the general PSN architecture, shown in Figure 4.1. More details on the initiator are presented in Sections 4.2.2 and 4.2.3.

A set of system requirements are established here as a basis for the design of RMP:

- **Online data submission, off-line data processing**: After submitting data online, participants should not be expected to stay online or possess the computational resources for collaborative anomaly detection.
Continuous data: An attribute is either continuous if it is numerical and arithmetic operations on it make sense; or categorical if it assumes values from a finite set and arithmetic operations on it do not make sense. RMP is designed to handle continuous attributes.

Horizontally partitioned data: Each participant contributes one or more data records with the same set of attributes. Mathematically, we represent a data record with a column vector in this chapter.

Targeted threat model: Like most schemes in the literature, RMP is based on the semi-honest adversarial model [134]. In this model, the participants and the data miner are “honest-but-curious”, in the sense that they might attempt to learn other parties’ private states from the information they receive during the protocol, although they do not deviate from the protocol.

Semi-honesty captures the all-too-common partial trust relationships between users and service providers in the real world who are usually not malicious. For example, email users trust their service providers to relay their emails but do not trust them to not snoop on their correspondence. Semi-honesty is the reason why data owners cannot send their privacy-sensitive data to the data miner by relying solely on encryption using the data miner’s public key; the data also has to be perturbed. Semi-honest participants are by definition passive attackers who do not collude with each other. However, RMP is designed to resist collusion between a subset of the participants and the data miner. Designing a collaborative data mining scheme that fulfils these requirements with low communication and computation costs while preserving the participants’ privacy is an open problem.

For privacy, we need to ensure that the attribute values in the participants’ contributed records are properly masked: given the masked values, the server cannot infer the original values. However, this must be done without over-sacrificing accuracy, i.e., the results of anomaly detection based on the masked data should be close to the corresponding result using the original data.

Multiplicative perturbation projects data to a lower dimensional space. The perturbed
4.2 RMP Privacy-Preserving Anomaly Detection Scheme

data matrix has a lower rank than the original data matrix, thereby forcing the attacker to solve an underdetermined system of linear equations. However, this is not enough, and the following design principles are pertinent.

**Resilience to distance inference attacks**: The review of existing multiplicative perturbation schemes in Section 2.2.2 reveals that distance-preserving transformations are susceptible to distance inference attacks [77, 113]. The challenge is to find a non-distance preserving transform that is suitable for certain data mining tasks. Random transformation [123] qualifies as such a transform in that it does not preserve the dot product or Euclidean distance among transformed data points, yet it is suitable for anomaly detection.

**Resilience to Bayesian Estimation attacks**: Bayesian Estimation is a general class of attack that exploits the pdf of the original data. Gaussian data is particularly exploitable because it reduces a maximum a posteriori estimation problem into a simple convex optimisation problem [148]. A suitable defense is to prevent this reduction by conditioning the pdf through a nonlinear transformation.

**Resilience to collusion**: Let \( X_i \in \mathbb{R}^{n \times m_i} \) be participant \( u_i \)'s dataset, and \( T \in \mathbb{R}^{w \times n} \) be a random matrix shared by all participants, where \( w < n \). The participant \( u_i \) perturbs its records as \( Z_i = TX_i \). Since the constructed anomaly detector is perturbed, the perturbation matrix needs to be shared with all clients. This approach of using a common \( T \) poses a serious privacy risk. If a rogue participant or client colludes with the server, the server can recover any participant’s original data using the breached perturbation matrix. The ultimate solution of generating an arbitrarily different perturbation matrix \( T_i \) for each participant \( u_i \) does not work, because building an accurate mining model requires consistency among the perturbation matrices. To overcome this challenge, RMP generates participant-specific perturbation matrices by perturbing \( T \).

4.2.2 Basic RMP

Now that the system requirements of PSNs are clearly formulated, we formally specify RMP as described in Algorithm 4.1. At the start of a participatory sensing campaign, the initiator generates a random matrix \( T \sim U_{w \times n}(0, 1) \) \((w < n)\) and perturbation parameter \( \alpha \). The initiator then publicises (i) \( T \), (ii) \( \alpha \), and (iii) the minimum and maximum values
Algorithm 4.1. RMP pseudocode

Role: Initiator
1: \( T \leftarrow \text{generate a random matrix from } \mathbb{U}_{w \times n}(0, 1) \)
2: \( \alpha \leftarrow \text{chose from } (0, 1] \)
3: \( x_{\text{min}} \leftarrow \text{Minimum value for each attribute} \)
4: \( x_{\text{max}} \leftarrow \text{Maximum value for each attribute} \)

Role: Participant
Require: \( N \) — nonlinear transformation function
5: if TTP does not exist then
6: \( T_i \leftarrow \text{generate a random matrix } T + \delta T_i, \text{ where } \delta T_i \sim \mathbb{U}_{w \times n}(-\alpha, \alpha) \)
7: else
8: TTP provides \( T_i \)
9: end if
10: \( Z_i \leftarrow T_i(N(X_i)) \)
11: Send \( Z_i \) to the server

Role: Mining server
12: \( Z_{\text{all}} \leftarrow \text{concatenate received } Z_i, \text{ for } i = 1, \ldots, q \)
13: \( M \leftarrow \text{train an anomaly detection algorithm and generate a model from } Z_{\text{all}} \)
14: Send \( M \) to the clients

Role: Client
Require: \( T, N, M \)
15: Perturb local test data \( Z_j \leftarrow T(N(X_j)) \)
16: Apply \( M \) to \( Z_j \) to test for anomalous records

of \( x \), for each attribute \( x \), denoted \( x_{\text{min}} \) and \( x_{\text{max}} \) respectively.

Suppose the participant \( u_i \) intends to contribute dataset \( X_i \in \mathbb{R}^{n \times m_i} \), where \( n \) is the number of attributes and \( m_i \) is the number of records. The participant first normalises his data. Normalising an attribute \( x \) means replacing \( x \) with \( (x - x_{\text{min}}) / (x_{\text{max}} - x_{\text{min}}) \), where \( x_{\text{min}} \) and \( x_{\text{max}} \) are the minimum and maximum values of \( x \) respectively. The participant then transforms \( X_i \) to \( Z_i \) in two stages.

Stage I:

The participant transforms \( X_i \) to \( Y_i \), by applying a nonlinear function \( N(x) \), to \( X_i \) element-wise:

\[
Y_i \overset{\text{def}}{=} N(X_i). \tag{4.1}
\]

The role of this nonlinear transformation is to condition the pdf of \( Y_i \) to thwart Bayesian Estimation attacks (as detailed later in Section 5.3). We design the function as
4.2 RMP Privacy-Preserving Anomaly Detection Scheme

\[ N'(x) = \text{sgn}(x)[1 - \exp(\beta_{dl} x^2)], \quad (4.2) \]

where \( \text{sgn} \) is the signum function. For suitable values of \( \beta_{dl} \) and normalised values of \( x \) (i.e., \(|x| \leq 1\)), the Double Logistic (DL) function approximates the identity function \( N'(x) = x \) well. To maximise this approximation, we set the optimal value of \( \beta_{dl} \) to the value that minimises the integral of the squared difference between the two functions:

\[ \beta_{dl} = \arg \min_\beta \int_0^1 [1 - \exp(-\beta x^2) - x]^2 dx \approx 2.81. \quad (4.3) \]

**Stage II:**

The participant generates

\[ T_i \overset{\text{def}}{=} T + \delta T_i, \quad (4.4) \]

where the elements of \( \delta T_i \) are drawn from \( U(-\alpha, \alpha), \ 0 < \alpha < 1 \). With \( T_i \), the participant then transforms \( Y_i \) to \( Z_i \):

\[ Z_i \overset{\text{def}}{=} T_i Y_i. \quad (4.5) \]

Note that while \( T \) represents a common uniformly distributed random transformation matrix (as introduced in Section 2.2.2), \( T_i \) is a participant-specific uniquely perturbed random transformation matrix. The participant then sends \( Z_i \) to the data miner.

**Stage III:**

Once the server \( S \) has received perturbed datasets \( Z_i \), for \( i = 1, \ldots, q \), from the participants, it concatenates them as:

\[ Z_{\text{all}} \overset{\text{def}}{=} \left[ Z_1 | Z_2 | \cdots | Z_q \right]. \quad (4.6) \]

The role of the data miner is to learn an anomaly detection function encoding a model of the underlying distribution of \( Z_{\text{all}} \). Given access to the transformation function \( T \),
clients can detect anomalies in their data with respect to the model. The overall information flow is depicted in Figure 4.1.

4.2.3 Mediated RMP

In the basic scheme just described, with knowledge of the matrix $T$ and the parameter $\alpha$, an attacker may try to estimate the $T_i$ of its target $i$, and by colluding with the data miner, try to recover $X_i$ from $Z_i$. RMP can be reinforced with the addition of a TTP to conceal $T$ and $\alpha$. In this TTP-mediated scheme, the initiator of the participatory sensing campaign no longer publicises $T$ and $\alpha$, which are held by the TTP. The participant $u_i$ still applies the stage-I nonlinear perturbation to $X_i$, transforming it to $Y_i$. However, the stage-II linear perturbation is modified as follows.

Stage II of (TTP-mediated)

The participant requests a perturbation matrix from the TTP, who then sends $T_i$, generated according to Eq. (4.4), to the participant. The participant transforms $Y_i$ to $Z_i$ as per Eq. (4.5), and sends $Z_i$ to the data miner.

From the privacy viewpoint, the TTP-mediated scheme would require collusion between the rogue participants $U_{\text{rog}} \subset U$ and the data miner in order to recover the original data of an honest participant. A sample attack starts with the rogue participants estimating $T$ and $\alpha$:

- $T$ is estimated as $\hat{T} = \sum_{v \in U_{\text{rog}}} T_v / |U_{\text{rog}}|$.  

- $\delta T_v$ can then be estimated as $\delta \hat{T}_v = T_v - \hat{T}, \forall v \in U_{\text{rog}}$. Each element of $\delta T_v$ is identically distributed according to $U(\alpha, \alpha)$, which has a variance of $\alpha^2 / 3$. Therefore, $\alpha$ can be estimated as $\hat{\alpha} = \sqrt{3S}$, where $S$ is the bias-corrected sample variance of the elements of $\delta T_v$ (there are $wn |U_{\text{rog}}|$ such elements).

The rogue participants then collude with the data miner, and send $\hat{T}$ and $\hat{\alpha}$ to the data miner. From here onwards, any attempt to recover the original data $X_i$ from $Z_i$ using $\hat{T}$ and $\hat{\alpha}$ is the same as in the basic scheme using publicly known $T$ and $\alpha$. Therefore, from
hereafter, by RMP we refer to the basic RMP method, and the detailed privacy analysis provided in Section 5.3 only concerns the basic RMP method.

4.3 Experimental Analysis

In this section we evaluate the quality of our proposed privacy-preserving anomaly detection scheme RMP when used with a basic AE. The main objective of our experiment is to measure the trade-off in accuracy of our AE anomaly detection algorithm as a result of maintaining the participants’ privacy. Note that the level of privacy in RMP is bounded to the level of the added noise $\alpha$, thus in our empirical evaluation, we consider the effect of different levels of privacy in terms of $\alpha$ on the overall accuracy of anomaly detection.

We now describe in detail our experimental methodology in terms of: (i) the data mining algorithm used for anomaly detection, (ii) the datasets used, and (iii) the accuracy metric.

Data mining algorithm: The experiments are conducted using an autoencoder, since (i) it can be trained in an unsupervised manner on either normal data alone, or a mixture of normal data with a small but unspecified proportion of anomalous data; (ii) it is capable of learning a wide variety of underlying distributions of training data; and (iii) the resulting anomaly detection function is compact and computationally efficient, making it practical for dissemination to clients. We deploy a three-layer AE with the same number of input and output units, i.e., the number of units is set corresponding to the dataset attributes $n$. The number of hidden units for each dataset is set to approximately half of the number of input units (empirically we found that increasing the number of hidden units causes overfitting). All weights and biases in the neural network are initialised randomly in the range of $[-0.1, 0.1]$. The learning rate and momentum are set to 0.25 and 0.85, respectively, and the number of training epochs range from 300 to 1000.

Anomalies can be identified by the autoencoder based on the integrated squared error between the inputs and outputs of the training records. Let $e$ be the reconstruction error of $z_j \in Z_{all}$, where $j = 1, \ldots, |Z_{all}|$. If the reconstruction error for a test sample is larger than the threshold $\tau = \mu(e) + 3\sigma(e)$, the record is identified as an anomaly, otherwise it
Table 4.1: Comparing AUC values of RMP against non-privacy anomaly detection.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$n$</th>
<th>Raw</th>
<th>$\alpha$</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
<td>0.1</td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>Abalone</td>
<td>8</td>
<td>1</td>
<td>0.95</td>
<td>0.92</td>
<td>0.87</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Banana</td>
<td>8</td>
<td>1</td>
<td>0.98</td>
<td>0.92</td>
<td>0.81</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Gas</td>
<td>168</td>
<td></td>
<td>0.99</td>
<td>0.98</td>
<td>0.98</td>
<td>0.95</td>
<td>0.92</td>
</tr>
<tr>
<td>HAR</td>
<td>561</td>
<td>0.99</td>
<td>0.98</td>
<td>0.98</td>
<td>0.97</td>
<td>0.96</td>
<td>0.95</td>
</tr>
<tr>
<td>OAR</td>
<td>242</td>
<td>0.99</td>
<td>0.99</td>
<td>0.98</td>
<td>0.94</td>
<td>0.90</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Note: “Raw” indicates the non-privacy preserving anomaly detection on the non-perturbed data, and $\alpha$ values indicate the level of imposed noise in RMP anomaly detection.

is identified as normal.

Datasets: Experiments are conducted on four real datasets from the UCI Machine Learning Repository (all collected from sensor networks except the fourth): (i) Human Activity Recognition using Smartphones (HARS), (ii) Opportunity Activity Recognition (OAR), (iii) Gas Sensor Array Drift (Gas), and (iv) Abalone. We also use the Banana synthetic dataset, generated from a mixture of two banana-shaped distributions. We ran the experiment on the first 1000 records of each dataset. Feature values in each dataset are normalised between $[0,1]$ and merged with 5% anomalous records, which are randomly drawn from $\mathbb{U}(0,1)$. In each experiment a random subset of the dataset is partitioned horizontally among the participants in batches of 30 records and submitted to the server for training. From hereafter the above settings have been used for all the experiments, unless otherwise stated.

Accuracy metric: We use the Receiver Operating Characteristic (ROC) curve and the corresponding Area Under the ROC Curve (AUC) to compare the anomaly detection accuracy of an autoencoder with and without data perturbation by RMP. Without data perturbation, the AUC should be close to 1 (where 0% false positive rate corresponds to 100% true positive rate). With data perturbation, the AUC is expected to decrease, and the results below show that RMP causes only small decrements.
4.3 Experimental Analysis

Empirical results

Table 4.1 compares the results using an autoencoder anomaly detector on the unper- turbed data records (“Raw”) along with the corresponding results using the privacy pre- serving scheme of RMP. The accuracy of RMP is affected by its two stages of transforma- tion, i.e., applying the double logistic function and the random transformation, and the level of added noise $\alpha$. As can be seen from the table, when data are perturbed with a marginal level of noise $\alpha = 0.01$, the accuracy decreases slightly (about 1%). Hence, it shows that the transformations do not have a significant impact on the accuracy of anomaly detection. In the reported results in Table 4.1, the dimensionality of the datasets
is reduced by $r = 1$, where $r = n - w$. Our empirical experiments show that the accuracy on datasets with a larger number of attributes $n$ are less affected by an increase of $r$, e.g., reducing $n$ by 40% only decreases accuracy by about 1%. However, that is not the case for datasets with small $n$, e.g., the Abalone and Banana datasets, where reducing the data dimensionality by half might result in a 10% reduction in accuracy, see Figure 4.2.

The level of added noise to the perturbation matrices has a major influence on the accuracy of RMP. As $\alpha$ increases, so does the loss in accuracy, especially in datasets with smaller numbers of attributes. Since the accuracy loss is generally small, RMP is a highly effective approach for privacy-preserving anomaly detection.

4.4 Conclusion

In a typical participatory sensing scenario, participants send data to a data mining server; the server builds a model of the data; and clients download the model for their own analyses. Collaborative anomaly detection refers to the case where the learned model is designed for anomaly detection. RMP is a privacy-preserving collaborative learning scheme that masks the participants’ data using a combination of nonlinear and linear perturbations, while maintaining detection accuracy. RMP protects the private data of participants using private perturbation matrices, imposes minimal communication and computation overhead on the participants, and scales for an arbitrary number of participants or clients. Our experiments show that RMP yields comparable results to non-privacy preserving anomaly detection using an autoencoder on a variety of real and synthetic benchmark datasets.

However, in this chapter we did not consider the privacy analysis of RMP. In the next chapter, we analytically investigate how RMP is resilient to two common types of attacks, namely, Bayesian Estimation and ICA, and further we improve the privacy of the nonlinear perturbation.
Chapter 5
ERMP: Enhanced Random Multiparty Perturbation

In the previous chapter, we introduced RMP to address privacy challenges for anomaly detection in PSNs. RMP maintains participants’ privacy while avoiding computationally intensive cryptographic techniques or the use of a TTP. It consists of a two-staged perturbation that can be executed on resource constrained devices and scales to any number of participants and clients. More specifically, RMP is able to allocate a customised random transformation to each participant.

In this chapter, we enhance RMP and call the new scheme as ERMP. ERMP includes a new non-linear transformation that improves the privacy of normal and anomalous records. Additionally, we provide extensible theoretical and experimental analysis in support of privacy effectiveness of the two-staged transformation. A statistical privacy measure called recovery resistance is proposed to estimate the fraction of perturbed data that can be recovered. We also show that ERMP is resilient to major attacks, ICA and MAP estimation.

The work arising from this chapter is under review as paper P5.

5.1 Introduction

In Chapter 4 we studied the problem of privacy-preserving anomaly detection in large collaborative networks. More specifically, we consider the case of a large number (thousands or more) of participants with limited computing resources and communication bandwidth, and who cannot stay online throughout the data mining session. We investigated how participants can share data with a data miner to train an accurate anomaly detection model without disclosing the true values of their sensitive attributes.

The privacy requirement in PSNs that we consider is that the data miner is not able to infer the original values from the masked values. Additionally, the accuracy requirement
indicates that the result of mining masked data should be close to the result of mining the raw (original) data. Randomisation approaches are an attractive approach to achieving privacy and accuracy. Exiting works on randomisation mostly focus on perturbing a single participant, or consider the case where all participants trust each other and agree on using the same perturbation matrix. While these approaches offer great efficiency and scalability, such approaches still have limitations for a real-life collaborative learning scenarios like PSNs. When multiple participants perturb their data by applying the same random matrix (or matrices), knowing the random matrix can help a rogue data miner reconstruct all the participants’ data. Some recent works, e.g., [41, 111, 120], attempted to overcome this limitation, but they are still susceptible to collusion attacks.

To encourage private random perturbation, in the previous chapter we have proposed Random Multiparty Perturbation, a privacy-preserving collaborative anomaly detection scheme. RMP is designed based on clearly defined PSNs requirements, including the ability to handle tabular structured continuous data. It distinguishes itself from existing schemes, reviewed in Section 2.2.2, by using a combination of nonlinear and participant-specific linear perturbations. The linear perturbation is based on what is called a random transformation in the literature, with the difference being that in RMP, each participant uses a private random matrix, instead of a common random matrix.

In this chapter, we propose introduce an enhanced RMP algorithm by using a new nonlinear perturbation, called poly7, to improve the privacy properties of the scheme. Previously we took it for granted that the two-staged transformation thwarts ICA and MAP estimation attacks. Now we conduct an extensive theoretical and experimental analysis of the effectiveness of our two-staged approach. Our empirical analysis shows that the poly7 nonlinear perturbation is overall more recovery-resistant than prior schemes. Furthermore, our experiments show that ERMP boosts the accuracy of anomaly detection by improving the separation of the transformed normal and anomalous records, as well as alleviating the computational complexity by significantly reducing the data dimensionality. Hence, it enables the use of more computationally demanding techniques such as one-class SVMs for large datasets.

ERMP has the following properties:
(i) It maintains the privacy of both normal and anomalous records. Unlike existing works that can only preserve the privacy of either normal or anomalous data points, e.g., [23], the nonlinear perturbation of ERMP is designed to provide privacy for both.

(ii) It is resistant to collusion, since participants individually generate their own random perturbations (linear and nonlinear).

(iii) It obtains comparable accuracy to non privacy-preserving anomaly detection schemes on a variety of benchmark datasets.

This chapter is organised as follows. Section 5.2 elaborates our proposed approach. Section 5.3 present a theoretical analysis of the privacy of ERMP. Sections 5.4 and 5.5 present our empirical evaluations of ERMP in terms of privacy and accuracy. Finally, Section 5.6 concludes the chapter and offers some directions for future research.

### 5.2 Enhanced RMP (ERMP)

ERMP is proposed to enhance the privacy of RMP. We redesign the nonlinear function $N$, by substituting the double logistic function with a $7^{th}$-order polynomial. The new algorithm follows Algorithm 4.1 except in Line 10, stage II, $N = \text{poly}7$. As detailed in the next section, poly7 is used to better condition the pdf of the data in order to thwart MAP attacks, thus achieving a lower recovery rate for both normal and anomalous records, and hence enhancing participants’ privacy. A detailed presentation of the derivation of poly7 is given in Section 5.3.1

### 5.3 Privacy Analysis

The very question of what “privacy” means needs to be answered for any privacy analysis to be meaningful. In the statistical disclosure control and privacy-preserving data publishing literature, the most popular semantic privacy criterion/definition is differential privacy. Differential privacy was designed for the scenario where a database server answers queries in a privacy-preserving manner by adding tailored Laplace-distributed
noise to the query results \cite{56,59}. In such a scenario, the database contains private data of multiple individuals. The participatory sensing scenario, where participants are data owners who publish data (instead of answering queries) about themselves alone, can be considered as a distributed version of the differential privacy scenario. In such a scenario, additional mechanisms are necessary if differential privacy is to be used. Evidence supporting the preceding claim can be found in many highly cited references including the following:

- Shi et al.’s scheme \cite{155} enables participants to upload encrypted values to a data aggregator, which computes the sum of the encrypted values. These values are perturbed with Laplace noise that satisfies \((\epsilon, \delta)\)-differential privacy, but the encryption relies on a trusted dealer allocating \(|\mathcal{U}| + 1\) secrets that sum to 0, to the data aggregator and the \(|\mathcal{U}|\) participants.

- Ács and Castelluccia’s scheme \cite{3} enables smart meters, organised into clusters, to send Laplace noise-tainted readings to an electricity distributor; but it requires all meters in a cluster to share pairwise keys with each other.

In general, additive perturbation of a centralised dataset can be made differentially private, but additive perturbation of distributed datasets is treated with either differential privacy and homomorphic-like encryption \cite{3,143,155}, or data-correlated noise \cite{72,131,188} without differential privacy.

In view of the above observation, we adopt the alternative approach of multiplicative perturbation as exemplified by Liu et al. \cite{114}, Mangasarian et al. \cite{123}, Bhaduri et al. \cite{23} and others. Corresponding to multiplicative perturbation, we use an alternative privacy definition to differential privacy, which we state informally now but formally later: a perturbation scheme is privacy-preserving with respect to an attack \(\mathcal{A}\) and a data distribution \(p_D\) if only a small fraction of the original data, characterised by \(p_D\), can be recovered from the perturbed data through \(\mathcal{A}\). This definition has three major components: (i) the reference attack; (ii) the data distribution, which captures an aspect of the attacker’s auxiliary information; and (iii) the recovery rate, which captures the notions of “small fraction” and “recovered.”
5.3 Privacy Analysis

To specify the reference attack, we first consider attacks to linear multiplicative perturbation schemes. These types of schemes project a data vector (and hence the whole data matrix) to a lower dimensional space so that an attacker has only an ill-posed problem in the form of an underdetermined system of linear equations $Ty = z$ to work with, where $z$ is a projection of vector $y$. An underdetermined system cannot be solved for $y$ exactly, but given sufficient prior information about $y$, an approximation of the true $y$ might be attainable. We can characterise an attack by the extent of prior information available to the attacker [77].

In a known input-output attack, the attacker has some input samples (i.e., some samples of the original data) and all output samples (i.e., all samples of the perturbed data), and knows the mapping between input samples and output samples [42, 77, 112, 148]. In the collaborative learning scenario where the data miner may collude with one or more participants to unravel other participants’ data, the known input-output attack is an immediate concern. In the following, our privacy analysis is conducted with respect to one of the major input-output attacks, based on Bayesian estimation.

Besides the reference attacks, we also need to define the recovery rate. If for a data vector $x$ the recovered copy is $\hat{x}$, then the relative error is $\xi \overset{\text{def}}{=} \|\hat{x} - x\|_2 / \|x\|_2$, where $\|\cdot\|_2$ is the Euclidean norm. Denote the joint distribution of $\xi$ and $x$ by $p_{\xi,x}(\xi, x)$, then we define the $\epsilon$-recovery rate with respect to the perturbation algorithm and attack as

$$r_\epsilon(A, p_D) \overset{\text{def}}{=} \int_{\xi=0}^\epsilon \int_{x \in D_x} p_{\xi,x}^A(\xi, x) \, dx \, d\xi,$$

(5.1)

where $D_x$ is the domain of the data vector, and $x$ is normalised. The joint distribution $p_{\xi,x}^A$ depends on the attack $A$ and data distribution $p_D$. At this point, we state the privacy definition formally as follows.

A probabilistic algorithm that takes $p_D$-distributed $x \in \mathbb{R}^n$ as input and produces $z \in \mathbb{R}^m$ as output is $(\epsilon, \delta)$-recovery resistant with respect to $p_D$ and attack algorithm $A$ if $r_\epsilon(A, p_D) = \delta$.

Suppose the attacker is targeting a particular participant by trying to solve

$$Z = \tilde{T}Y = (T + \delta T)Y.$$
for $Y$. In the analysis below, let $z \sim Z$ represent a column of $Z$, and $y \sim Y$ represent a column of $Y$.

5.3.1 Attack Based on Bayesian Estimation

We consider two scenarios: where $\hat{T}$ is known, and where $\hat{T}$ is unknown.

**Scenario where $\hat{T}$ is known**: This is the worst-case scenario, and here we assume the attacker somehow knows $\hat{T}$ exactly but not $Y$, for example when the attacker manages to predict the output of the victim’s improperly initialised pseudorandom number generator (in fact, such a vulnerability was discovered on the Android mobile platform in mid-2013). In a Bayesian formulation, maximum a posteriori (MAP) estimation is a more general approach than maximum likelihood estimation because the former takes the prior distribution into account—in this case, the distribution of the original data vectors. The MAP estimate of $y$, given $\hat{T}$ and $z$, is

$$\hat{y} = \arg \max_y p(y|z, \hat{T}) = \arg \max_y \frac{p(z|\hat{T}, y)p(\hat{T})p(y)}{p(z|\hat{T})p(\hat{T})}$$

$$= \arg \max_y \frac{p(y)}{\int_{\mathbb{R}^n} p(z|\hat{T}, y) dy} = \arg \max_{y \in Y} p_Y(y),$$

(5.2)

where $Y = \{y : z = \hat{T}y\}$. Note that:

- The factor $p(z|\hat{T}, y)$ translates to the constraint $y \in Y$.
- The integral in the denominator does not contribute towards maximising $y$.

If $Y$ is an $n$-variate Gaussian with a positive definite covariance matrix, then Eq. (5.2) becomes a quadratic programming problem with solution \[148\] Theorem 1:

$$\hat{y} = \bar{y} + \Sigma_Y \hat{T}' \Sigma_Z^{-1}(z - \hat{T}\bar{y}),$$

(5.3)

where $\bar{y}$ and $\Sigma_Y$ are the sample mean and sample covariance matrix of $\hat{y}$ respectively, and $\Sigma_Z$ is the sample covariance matrix of $Z$. Note that $\Sigma_Y$ is positive definite, provided the covariance matrix has full rank and there are more samples of $Y$ than dimensions of $\hat{y}$ \[60\].

In this case, we can write $\Sigma_Y = AA'$, where $A$ is a nonsingular matrix. Furthermore,
\[ \Sigma_Z = \tilde{T}\Sigma_y\tilde{T}' = \tilde{T}AA'\tilde{T}'. \] Since \( \tilde{T} \) is nonsingular, \( \Sigma_Z^{-1} \) and therefore the solution to Eq. (5.3) exists.

The analysis above suggests that to thwart MAP estimation, we cannot hope to generate \( \tilde{T} \) such that \( \Sigma_Z \) is singular. Instead, the key is to design a nonlinear function \( \mathcal{N} \) that transforms a potentially Gaussian or Laplace (typical of sparse datasets) data distribution to a distribution that deters an accurate solution of Eq. (5.2).

As the nonlinear function, Bhaduri et al. [23] proposed \( \tanh \), i.e., \( \mathcal{N}(x) = \tanh(\beta_t x) \), where \( \beta_t \) is a tuneable parameter. We propose

\[ \beta_t = \arg \min_{\beta} \int_0^1 [\tanh(\beta x) - x]^2 \, dx \approx 1.23. \tag{5.4} \]

In Chapter 4, we proposed the double logistic function as the nonlinear function, i.e.,

\[ \mathcal{N}(x) = \text{sgn}(x)[1 - \exp(-\beta_{dl} x^2)], \]

where

\[ \beta_{dl} = \arg \min_{\beta} \int_0^1 [1 - \exp(-\beta x^2) - x]^2 \, dx \approx 2.81. \tag{5.5} \]

In this chapter, we propose a 7th-order polynomial curve

\[ \text{poly7}(x) \overset{\text{def}}{=} \sum_{j=0}^{7} a_j x^j, \tag{5.6} \]

having (i) the desirable characteristics of \( \tanh \) in terms of protecting anomalies [23], and (ii) the desirable characteristics of the double logistic function in terms of protecting normal data (see Section 5.5). The design is based on 7 constraints:

\[ \text{poly7}(0) = 0, \tag{5.7} \]

\[ \frac{d}{dx} (\text{poly7}(x)) \bigg|_{x=0} = \frac{d}{dx} (\tanh(\beta_t x)) \bigg|_{x=0}, \tag{5.8} \]

\[ \text{poly7}(0.45) = 0.50, \tag{5.9} \]

\[ \text{poly7}(0.50) = 0.50, \tag{5.10} \]

\[ \text{poly7}(0.55) = 0.50, \tag{5.11} \]

\[ \frac{d^2}{dx^2} (\text{poly7}(x)) \bigg|_{x=0.50} = 0, \tag{5.12} \]
Table 5.1: Suitable Values for the Coefficients of Poly7

<table>
<thead>
<tr>
<th>$a_7$</th>
<th>$a_6$</th>
<th>$a_5$</th>
<th>$a_4$</th>
<th>$a_3$</th>
<th>$a_2$</th>
<th>$a_1$</th>
<th>$a_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1928</td>
<td>-9.4086</td>
<td>8.1029</td>
<td>16.2265</td>
<td>-20.8269</td>
<td>5.4659</td>
<td>1.2300</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: The stated values are rounded to four decimal places.

\[
poly7(0.95) = 0.95, \tag{5.13}
\]

\[
\left. \frac{d}{dx} (poly7(x)) \right|_{x=1} = 0. \tag{5.14}
\]

Constraints Eq. (5.7) to Eq. (5.8) give poly7 the slope of \(\tanh(\beta_{i}x)\) at the origin for protecting near-origin anomalies (given that the data is scaled so that the mean occurs around \(x = 0.5\)). Constraints Eq. (5.9) to Eq. (5.12) emulate the inflection of the double logistic function and create a plateau for protecting normal data. Constraints Eq. (5.13) to Eq. (5.14) give poly7 a slope that resembles the slope of \(\tanh(\beta_{i}x)\) at \(x = 1\). Solving the constraint equations gives us values of \(a_i\) in Table 5.1 and Figure 5.1 plots poly7 using these values.

Figure 5.1: Plot of the tanh, the double logistic and the poly7 functions.

Since transforming the data pdf is crucial, Figure 5.2 is provided to visualise the resultant distributions when a bivariate Gaussian pdf is transformed using tanh, the double
Figure 5.2: Distributions of $y = \mathcal{N}(x)$ (note here that in this illustration $x$ is Gaussian distributed with zero mean and covariance matrix diag[1, 1]): (a) $\mathcal{N}$ is the identity function; (b) $\mathcal{N}$ is tanh. (c) $\mathcal{N}$ is the double logistic function; (d) $\mathcal{N}$ is poly7—note the PDF is uniformly close to zero but not zero.

Figure 5.3: Distributions of $y = \mathcal{N}(x)$ (note here that in this illustration $x$ is Laplace distributed with zero mean and covariance matrix diag[1, 1]): (a) $\mathcal{N}$ is the identity function; (b) $\mathcal{N}$ is tanh. (c) $\mathcal{N}$ is the double logistic function; (d) $\mathcal{N}$ is poly7—note the PDF is uniformly close to zero but not zero.

logistic function, and poly7. Figure 5.3 shows the transformed distributions when the original pdf is a bivariate Laplace. Both Figure 5.2 and Figure 5.3 show that poly7 essentially flattens the pdf of the transformed data, whether the original pdf is Gaussian or Laplace; this is expected to deter the accurate solution of the maximisation problem of Eq. (5.2).

We note that the pdf of a transformed random variable can be obtained analytically, provided the pdf of the original random variable and the inverse transformation function can be obtained analytically [79]. Whereas tanh and the double logistic function are readily invertible, poly7 is not, thus making an analytical expression of the transformed pdf unavailable even if the pdf of the original random variable is available. The implication is significant: there does not exist a distribution of $x$ for which the attacker can find an analytical expression for $p(y)$ in Eq. (5.2), where $y = poly7(x)$. 
Scenario where $\tilde{T}$ is unknown: In the basic scheme, the attacker knows $T$ and the relationship between $T$ and $\tilde{T}$ (see Eq. 4.4). In the TTP-mediated scheme, the attacker does not know $T$, but can estimate $T$ as $\hat{T} = \frac{1}{|U|} \sum_{v \in U} T_v$. To simplify analysis, we assume the attacker knows $T$ exactly. Note that even with precise knowledge of $T$ and $\alpha$, without further information, any matrix value between $T - \alpha 1$ and $T + \alpha 1$ can be an estimate of the victim’s matrix $T$. According to Lemma 5.1, for every element of $T$, there is a 50% chance of guessing its value wrong by at least $(2 - \sqrt{2})\alpha$.

Lemma 5.1. Let $D$ be the difference between two $\mathbb{U}(-\alpha, \alpha)$-distributed random variables. Then for $0 \leq d \leq 2\alpha$,

$$\Pr[|D| \geq d] = \frac{(2\alpha - d)^2}{(4\alpha^2)}.$$  

Proof. Let $A$ and $B$ be two $\mathbb{U}(-\alpha, \alpha)$-distributed random variables. Then the pdf of $D = A - B$ is given by the convolution

$$p_D(d) = \frac{1}{2\alpha} \int_{-\alpha}^{\alpha} p_{\mathbb{U}(-\alpha, \alpha)}(d + b)db$$

$$= \begin{cases} 
\frac{1}{2\alpha} \left(1 + \frac{d}{\alpha}\right) & -2\alpha \leq x < 0, \\
\frac{1}{2\alpha} \left(1 - \frac{d}{\alpha}\right) & 0 \leq x < 2\alpha, \\
0 & \text{elsewhere.}
\end{cases}$$

To get $\Pr[|D| \geq d]$ where $0 \leq d \leq 2\alpha$, we integrate the expression above from $-2\alpha$ to $-d$, and from $d$ to $2\alpha$. With a little algebra we get $\Pr[|D| \geq d] = \frac{(2\alpha - d)^2}{4\alpha^2}$. □

MAP estimation can be used to estimate both $Y$ and $\tilde{T}$. The MAP estimates of $Y$ and $\tilde{T}$, given $Z$, are

$$\hat{(T, Y)} = \arg\max_{T, Y} p(T, Y|Z)$$

$$= \arg\max_{T, Y} \frac{p(Z|\tilde{T}, Y)p(\tilde{T})p(Y)}{\int \int p(Z|T, Y)p(T)p(Y)dTdY}$$

$$= \arg\max_{(T, Y) \in \Theta} p_\gamma(T)p_\gamma(Y),$$

(5.15)

where $\Theta \in \{(\tilde{T}, Y) : Z = \tilde{T}Y\}$. Substituting $p_\gamma$ with $\mathbb{U}_{|w \times n}(T - \alpha 1, T + \alpha 1)$ and assuming
p_y to be zero-mean Gaussian converts Eq. (5.15) to:

$$\min_{\tilde{T}, Y} \sum_{j=1}^{m} y_j / \Sigma_y^{-1} y_j$$

(5.16)

s.t. \( Z = \tilde{T}Y, T - a1 \preceq \tilde{T} \preceq T + a1 \).

In Eq. (5.16), \( y_j (j = 1, \ldots, m) \) are columns of \( Y \). Note that in the equality constraint, both \( \tilde{T} \) and \( Y \) are optimization variables, so even the Gaussian assumption does not reduce Eq. (5.16) to a convex problem. As previously explained, ERMP’s stage-1 nonlinear perturbation converts a potentially Gaussian (Laplace) data distribution to a non-Gaussian (non-Laplace) one. This hampers the attacker’s solution of not only Eq. (5.2) but also Eq. (5.15), which is a harder problem than Eq. (5.2).

### 5.4 Experimental Results on Accuracy

In this section we present our experimental results on the accuracy of data mining operations on data perturbed by ERMP. More specifically, we evaluate the impact of the two-stage transformation as well as the impact of added noise. ERMP is designed for anomaly detection, but is not limited to a single type of anomaly detection algorithm.

**Data mining algorithm:** For our study, we make use of two widely used anomaly detection methods, an autoencoder and a one-class SVM. The autoencoder (AE) is implemented in MATLAB with mini-batch learning, following the original scheme in [145]. In our experiments, a five-layered autoencoder, with tied weights and a sigmoid activation function for both the encoder and decoder is used. Initially an autoencoder was trained based on greedy layer-wise pre-training (i.e., training one layer at a time) to extract features, and then using these features to train the next layer (as discussed in Section 2.3.2). Once the network was trained, the learnt parameter values (weights and bias) were used as initialisation values of a multilayer perceptron with the same number of inputs and outputs. Then the network was fine-tuned by gradient descent to adjust the weights and minimise the reconstruction error \( l \). The whole process of pre-training and fine-tuning was performed in an unsupervised manner for anomaly detection. The hyperparameters
of the autoencoders, namely, learning rate (for pretraining 0.001–0.01, for fine tuning 0.1–1), number of epochs (for pretraining 5–10, for fine tuning 10–30), and number of hidden units \( h \leq n \), are set based on the best performance on a validation set. For the one-class SVM the Dd-tools \(^{[166]}\) and LIBSVM toolboxes are used, and their parameters, width \( \nu \) (from 0 to 1), and \( \sigma \) (from 1 to \( \infty \)), are selected via a grid-search.

**Datasets:** Experiments are conducted on five real datasets from the UCI Machine Learning Repository (all collected from sensor networks except the fourth): (i) Abalone (ii) Gas Sensor Array Drift (Gas), (iii) Opportunity Activity Recognition (OAR), (iv) Daily and Sport Activity (DSA) (v) Human Activity Recognition using Smartphones (HAR), with a dimensionality of 8, 128, 242, 315\(^{[1]}\) and 561, respectively. We also use two synthetic datasets. One is “Smiley-face” dataset, generated from a mixture of two compact Gaussians and an arc shaped distribution. The dataset contains 20 dimensions and in any two dimensions the components of the face are randomly moved. The other is the “GME” dataset, which is a mixture of four separated Gaussians in 100 dimensions. In our experiments we used the first 2000 records, 80% for training and 20% for testing. Training and testing records, respectively, were mixed with 5% and 20% anomalies, randomly drawn from \( \mathcal{U}(0, 1) \). These records are horizontally partitioned among the participants in batches of 30 records (one participant has less than 30 records).

### 5.4.1 Impact of Two-Stage Transformations

The primary objective of combining linear and nonlinear transformations is to prevent the privacy threats to participants’ data. Correspondingly, the two-stage transformation is designed to collapse the distances between normal records as well as anomalous records. Figure 5.4 demonstrates the effect of the two-stage transformation on a synthetic dataset, known as the Banana dataset. For illustration purposes this example shows how the data from a single participant is perturbed, following Eq. (4.1) and (4.5), using the same linear transformation \( T \), and three different nonlinear transformations \( N \), i.e., poly7, tanh, and double logistic function. The blue dots represent normal records.

\(^{[1]}\)DSA is a large dataset comprising the time series measurements from 45 wearable sensors for 19 activities. We select a portion of the time series for each of the first 7 activities, yielding a total of 315 concatenated time series features.
5.4 Experimental Results on Accuracy

Figure 5.4: Demonstration of the effect of random projection on normal (star) and anomalous (circle) records.

and red circles are random anomalies. Notice that in the perturbed datasets the normal records are closer together and the anomalies are further away from the majority of the data, i.e., the two-stage transformation amplifies the distance between normal records and anomalies. Then it is expected that these transformations enhance anomaly detection.

Projecting the data to a lower dimensional space with a linear random transformation can be beneficial for anomaly detection, since the transformations act as a regulariser, which enhances the accuracy of anomaly detection (discussed in Chapter 8). To investigate the effect of the combined transformations on the accuracy of anomaly detection, each nonlinear transformation is combined with the random transformation, and their AUC values are compared with unperturbed data records (“Raw”) in Table 5.2.
Table 5.2: Comparing the impact of different nonlinear transformations on the AUC of anomaly detection techniques.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>1SVM</th>
<th>AE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Raw</td>
<td>Tanh</td>
</tr>
<tr>
<td>Abalone</td>
<td>0.94</td>
<td>0.97</td>
</tr>
<tr>
<td>Smiley</td>
<td>0.85</td>
<td>0.98</td>
</tr>
<tr>
<td>Gas</td>
<td>0.91</td>
<td>0.99</td>
</tr>
<tr>
<td>OAR</td>
<td>0.91</td>
<td>0.96</td>
</tr>
<tr>
<td>DSA</td>
<td>0.84</td>
<td>0.98</td>
</tr>
<tr>
<td>HAR</td>
<td>0.88</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Note: “Raw” indicates non-privacy preserving anomaly detection on the non-perturbed data.

Table 5.3: Comparing the impact of the added noise level $\alpha$ on the AUC of anomaly detection techniques.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>1SVM</th>
<th>AE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Raw</td>
<td>0.2</td>
</tr>
<tr>
<td>Abalone</td>
<td>0.94</td>
<td>0.97</td>
</tr>
<tr>
<td>Smiley</td>
<td>0.85</td>
<td>0.96</td>
</tr>
<tr>
<td>Gas</td>
<td>0.91</td>
<td>0.98</td>
</tr>
<tr>
<td>OAR</td>
<td>0.91</td>
<td>0.96</td>
</tr>
<tr>
<td>DSA</td>
<td>0.84</td>
<td>0.98</td>
</tr>
<tr>
<td>HAR</td>
<td>0.88</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Note: “Raw” indicates non-privacy preserving anomaly detection on the non-perturbed data, and $\alpha$ values indicate the level of imposed noise in ERMP anomaly detection.

periment was conducted for the 1SVM and AE, and between these two methods it was observed that 1SVM reveals a greater improvement in accuracy, e.g., experiencing a more than 10% increase in AUC for the Smiley, DSA and HAR datasets. This might be due to the depth of the AE architecture being better able to capture the generalisation of the underlying distribution. Comparing the nonlinear functions, the AUC results of poly7 are slightly better than the two others. A clearer separation by using poly7 is also reflected in the top right subfigure of Figure 5.4 where $N = \text{poly7}$, which indicates the suitability of poly7 for anomaly detection.
5.4.2 Impact of Imposed Noise

To maintain participants’ privacy, ERMP combines linear and nonlinear transformations, and personalises the perturbation by imposing random noise. Recall from the previous section that applying these transformations can enhance the anomaly detection results. Consequently, the effect of added noise requires further investigation. Table 5.3 studies this effect by comparing the AUC values of unperturbed data records along with the corresponding results using the privacy preserving scheme of ERMP with injected noise in the range of [0, 0.6].

As can be seen from the table, the AE presents a more consistent result, even when adding a relatively high level of noise. For example, with $\alpha = 0.4$, AE shows little or no loss of accuracy. Its resilience to the noise is due to its deep architecture and the applied weight penalty. SVDD, on the other hand, is more vulnerable to noise. When data are perturbed with a low level of noise $\alpha = 0.2$, the accuracy loss of SVDD is negligible (less than 1%), while its AUC decreases faster for higher levels of noise.

Given that the two-stage perturbation with $\alpha = 0.2$ satisfies the privacy requirements and does not exert significant impact on the AUC values, either of the anomaly detection techniques can be used. If a larger level of noise is required, e.g., for the sake of greater privacy, adopting a more noise resilient technique such as an AE is more appropriate.

5.5 Experimental Results on Privacy

Experimental results are provided in this section on the recovery resistance of ERMP against the MAP estimation attack, in terms of the $\epsilon$-recovery rate defined in Eq. (5.1). In the absence of an analytical expression for Eq. (5.1), we estimate the $\epsilon$-recovery rate as the fraction of test data that can be recovered to within a relative error of $\epsilon$:

$$
\frac{\# \left\{ \hat{x}_i : \frac{\| \hat{x}_i - x_i \|_2}{\| x_i \|_2} \leq \epsilon, i = 1, \ldots, m \right\}}{m},
$$

(5.17)

where $x_i$ and $\hat{x}_i$ are the $i$th original data record and its attacker-estimated value, respectively.
Table 5.4: Evaluated schemes

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Nonlinear perturbation function (Stage 1)</th>
<th>Linear projection matrix (Stage 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP [114]</td>
<td>None</td>
<td>$T \sim \mathcal{N}_{w \times n}(0, 4)$</td>
</tr>
<tr>
<td>tanh+RT</td>
<td>tanh [23]</td>
<td>$T \sim \mathcal{U}_{w \times n}(0, 1)$</td>
</tr>
<tr>
<td>DL+RT (Chapter 4)</td>
<td>Double logistic</td>
<td>$T \sim \mathcal{U}_{w \times n}(0, 1)$</td>
</tr>
<tr>
<td>poly7+RT (ERMP)</td>
<td>poly7</td>
<td>$T \sim \mathcal{U}_{w \times n}(0, 1)$</td>
</tr>
</tbody>
</table>

To execute MAP estimation, the attacker can either apply the formulate in Eq. (5.3) formula, provided the original data is multivariate Gaussian distributed; or solve the constrained optimisation problem (5.2). To solve optimisation problem (5.2), the attacker needs to evaluate an objective function that is the pdf of the original data. For this, the attacker can estimate the pdf of the original data as the pdf of the leaked input samples, using multivariate kernel density estimation (KDE). For KDE, we use Ihler and Mandel’s Kernel Density Estimation Toolbox for MATLAB. Among the kernels supported, we use the Epanechnikov kernel — which is optimal in the sense of the asymptotic mean integrated squared error — with uniform weights.

Four schemes as defined in Table 5.4 are evaluated in the worst-case scenario where the attacker knows precisely the victim’s perturbation matrix. The evaluation is performed using purely Gaussian datasets, purely Laplace datasets, and an assortment of synthetic and real datasets.

- Purely Gaussian datasets: Figure 5.5 shows that poly7+RT provides significantly higher recovery resistance for normal data compared to the other schemes. For protecting anomalous data, poly7+RT is not as good as RP or tanh+RT, but is better than DL+RT. More importantly, even in the worst case, poly7+RT keeps the 0.2-recovery rate at a maximum of 10%, and the 0.1-recovery rate at below 10%.

- Purely Laplace datasets: Figure 5.6 shows that although the 0.2-recovery rate of MAP estimation against poly7+RT reaches 60% for normal data, this rate is significantly lower than those of the other schemes. Furthermore, the 0.1-recovery rate against poly7+RT is well below 10%, which is again significantly lower than those

5.5 Experimental Results on Privacy

Figure 5.5: Recovery rates of MAP estimation attacks against the evaluated schemes, on $w \times 1000$ data projected from $15 \times 1000$ normalised Gaussian-distributed data (zero mean, identity covariance matrix).

Figure 5.6: Recovery rates of MAP estimation attacks against the evaluated schemes, on $w \times 1000$ data projected from $15 \times 1000$ normalised Laplace-distributed data (zero mean, unity scale).
Figure 5.7: 0.1-recovery rates of the MAP estimation attack against the evaluated schemes, on various datasets. The rank of the perturbation matrix, $w$, is set as \( \lfloor (n + 1)/2 \rfloor \), where $n$ is the number of attributes. Note that the recovery rates are zero in many cases.
of the other schemes. For anomalous data, poly7+RT is more recovery-resistant than RP and DL+RT, and is only less so than tanh+RT.

- Assorted synthetic and real datasets: Consistent with the results for purely Gaussian and purely Laplace datasets, as shown in Fig. 5.7, poly7+RT outperforms tanh+RT and DL+RT in terms of recovery resistance for normal data. For protecting anomalous data, poly7+RT is worse than tanh+RT but better than DL+RT. Note that in many cases zero recovery rates occur. For example, poly7 achieves (0.1, 0)-recovery resistance for the Adult, Gas, OAR, DSA and HAR datasets.

In summary, for protecting normal data, poly7+RT is significantly better than other schemes, whereas for protecting anomalous data, poly7+RT is not as recovery-resistant as tanh+RT. By being able to achieve zero or near-zero recovery rates for more cases than the other schemes, poly7+RT is the overall best privacy-preserving scheme among the evaluated schemes.

5.6 Conclusion

Anomaly detection in large distributed networks raises concerns for the participants in terms of revealing the content of their records, and for the data miner in terms of improving the accuracy of the generated anomaly detection model that is derived from the perturbed records. Increasing numbers of smart devices creates further challenges in terms of the efficiency and scalability of existing solutions. To overcome these challenges, we proposed ERMP, a privacy-preserving anomaly detection scheme that avoids computationally expensive cryptographic approaches and builds on a light weight transformation. ERMP comprises a combination of nonlinear and participant-specific linear perturbations. The nonlinear function, poly7, is designed to deliver a higher level of privacy. We have provided both analytical and empirical analysis of the effectiveness of ERMP. Our accuracy evaluation demonstrates the suitability of the two-stage transformation for anomaly detection in terms of its ability to enhance the accuracy of two different types of anomaly detection methods. Our privacy evaluation reveals that poly7 is significantly better than the tanh and double logistic functions in protecting normal
data, whereas it is not as recovery-resistant as tanh for protecting anomalous data when the dimensionality of the data is low.

So far we have focussed on the case where the random perturbation matrix for each participant is generated from a public perturbation matrix. In the next chapter we study how we can assign completely random transformations to each participant while maintaining the accuracy of the anomaly detection model.
Chapter 6
IRMP: Individually Random Multiparty Perturbation

In Chapter 4 we introduced a privacy-preserving anomaly detection technique for PSNs, called RMP, and in Chapter 5 we introduced an enhanced version of RMP, called ERMP. The design of ERMP (and RMP) enables collaborative learning in an untrusted environment. It relaxes the existing assumption (i.e., sharing the same transformation matrix) in the literature of PPDM and allows participants to generate a unique transformation matrix by imposing random noise, while enabling the server to generate an accurate model. As previously discussed in Section 4.2.1, however, ideally we would like to assign an entirely different perturbation matrix to each user. So far this goal has been restricted due to the impact this would have on accuracy.

In this chapter, we study this challenge and investigate how to apply an individually random transformation matrix for each participant while maintaining a high level of accuracy. We propose Individually Random Multiparty Perturbation (IRMP), a lightweight perturbation scheme consisting of a nonlinear stage and a linear stage, that enables participants to randomly perturb their records. IRMP is an extension of RMP but with greater flexibility in the choice of transformations. Empirical evaluations on the performance of IRMP show no decrease in the accuracy of anomaly detection. The reason lies in the impact of our transformation on the data and the use of a deep autoencoder, known as a Contractive Autoencoder. Finally, we conduct an experimental analysis of the scalability and efficiency of our random transformation method, by varying the number of participants and records.

The work arising from this chapter is under review as paper P6.

6.1 Introduction

Data randomisation is a promising approach for privacy-preserving data mining in large collaborative networks such as participatory sensing. Unlike distance-preserving trans-
formations, random transformation does not preserve Euclidean distances and inner products between data instances, hence it thwarts distance inference attacks. However, if this method is applied to collaborative learning, all the participants must agree on the same perturbation matrix, and then collusion attacks may succeed. Furthermore, the mining models generated from the perturbed records are specific to the participants.

Correspondingly, in the previous chapters we have proposed RMP, a privacy-preserving collaborative anomaly detection scheme, which uses a combination of a nonlinear function and a linear perturbation. In RMP, each user \( u_i \) perturbs their data as \( T_i \mathcal{N}(X) \), where \( \mathcal{N} \) is a nonlinear function and \( T_i \) is a participant-specific matrix. Instead of sharing a common random matrix, each participant generates a unique perturbation matrix \( T_i = T + \delta T_i \), derived from a publicly available random matrix \( T \sim \mathcal{U}(0, 1) \) and additive noise \( \delta T_i \) randomly drawn from \( \mathcal{U}(-\alpha, \alpha) \), where \( 0 < \alpha < 1 \). Clients can then use the public matrix \( T \) and RMP’s generated model for testing. Though relaxing the existing assumption of applying the same perturbation matrix and enhancing privacy, RMP still requires the participants to share \( T \). This means that their privacy is tied to the level of the added noise \( \alpha \) in the personalised transformation matrix \( T_i \) (see Lemma 5.1).

To overcome this issue, in this chapter we propose a privacy-preserving anomaly detection scheme called Individually Random Multiparty Perturbation (IRMP), in which participants use their own unique, randomly generated perturbation matrix to randomise their data. Here, we challenge the utility of requiring a shared transformation matrix, and raise the bar by removing this constraint in IRMP. Our new scheme is designed so that the data miner can build an accurate anomaly detection model from the aggregated records of all participants. This model then can be distributed to clients who want to test for anomalies in their local data. To the best of our knowledge, IRMP is the first collaborative privacy-preserving scheme in which participants can use their own individual perturbation matrix to randomise their data, in contrast to previous works that require a shared perturbation matrix for the sake of accuracy. A key challenge in this context is how to perform anomaly detection on data with individual random perturbations. We propose to address this challenge by using a deep learning model based on a contractive autoencoder, which is designed to include a regularisation term in its optimisation.
that makes it resilient to small perturbations in the data. More specifically, we show that by using a deep learning architecture for our anomaly detector based on a contractive autoencoder, our anomaly detection scheme can generate an accurate model from randomly transformed data, when each participant’s data has been perturbed by individually unique perturbation matrices \([78, 144]\). We also show that our proposed approach outperforms traditional anomaly detection methods such as 1SVMs in the presence of randomly perturbed data.

IRMP also offers the following desirable properties:

(i) It is resistant to collusion, since participants individually generate their own random perturbations (linear and nonlinear).

(ii) It imposes no restrictions on the kind or the number of clients that access the anomaly detection function generated by IRMP, i.e., the model can be used by any party who understands the two-stage perturbation.

(iii) It obtains comparable accuracy to non privacy-preserving anomaly detection schemes on a variety of benchmark datasets.

The remainder of the chapter is organised as follows. Section 6.2 presents our proposed privacy-preserving anomaly detection approach IRMP. Section 6.3 demonstrates the experimental analysis concerning the privacy and performance of IRMP on various real-world and synthetic datasets. Section 6.4 concludes the chapter.
6.2 IRMP Privacy-Preserving Anomaly Detection Scheme

An ideal privacy solution in PSN’s is to generate an independent perturbation matrix $T_i$ for each participant $u_i$, i.e., $|U|$ different participation matrices corresponding to the $|U|$ participants. However, building an accurate mining model with traditional anomaly detection techniques requires consistency among the perturbation matrices. The proposed approach, IRMP, overcomes this challenge and introduces a more flexible privacy-preserving scheme that enables participants to choose an arbitrary perturbation matrix. Moreover, it maintains accuracy by using a contractive autoencoder, which is invariant to a range of transformations in the input. This section formally presents the IRMP scheme and then elaborates how we can avoid accuracy loss when applying individual transformations.

6.2.1 Privacy-Preserving Random Perturbation

IRMP’s two-stage data perturbation is designed to fulfil the aforementioned requirements. Suppose that participant $u_i$ intends to contribute its captured measurements to the server $S$ for anomaly detection. The participant first simply transforms the dataset $X_i \in \mathbb{R}^{n \times m_i}$ to $Z_i \in \mathbb{R}^{w \times m_i}$ in two stages:

**Stage I:** Using a nonlinear function $N$, e.g., a $7^{th}$-order polynomial function described in Section 5.3.1 as $poly7$, the participant transforms $X_i$ to $Y_i$:

$$Y_i \overset{\text{def}}{=} N(X_i). \quad (6.1)$$

**Stage II:** The participant generates a random transformation matrix $T_i$, e.g., drawn from $U(0, 1)$, and transforms $Y_i$ to $Z_i$

$$Z_i \overset{\text{def}}{=} T_i Y_i. \quad (6.2)$$

Note that to generate $T_i$ in IRMP no transformation matrix is shared and no noise is applied.

Data collection and data modelling on the server is the same as stage-III of RMP,
6.2 IRMP Privacy-Preserving Anomaly Detection Scheme

The learned model $M$ can then be exercised by clients to identify anomalies in their test records, by following the two stages in Eq. (6.1) and Eq. (6.2). Next we illustrate how to perform anomaly detection when each user applies their own random perturbation matrix.

6.2.2 Privacy-preserving Anomaly Detection with CAE

An autoencoder is a learning algorithm that discovers multiple levels of representation, and constructs higher-level features to abstract the input. The trained model $M$ for an autoencoder should result in low reconstruction error for points sampled from the same distribution as the training examples, and a high error otherwise. In principle, good generalisation can be achieved by stacking up several autoencoders in a deep model [16]. Empirically it has been observed that deeper networks often capture more abstract features than their shallow counterparts, such as SVMs [105], because more abstract features can often be captured in terms of less abstract ones.

Invariance to small perturbations of the input is a highly desirable property for many anomaly detection and classification tasks, and some deep autoencoders have shown promising results in automatically learning invariant features [17, 18, 22]. The concept of invariance implies extracting features that are insensitive to variations in the input, such as translation, rotation, scaling, etc. [78, 144]. To favour learning models that are insensitive to such variations, a regularisation term can be added to the autoencoder’s objective function, see Eq. (6.3). Recent works suggest that regularised autoencoders can be efficient at capturing the structure of the input distribution based on noisy training examples [9].

A promising type of regularised autoencoder is the contractive autoencoder (CAE) proposed by Rifai et al. [144, 145]. In order to learn models that are robust to changes in the underlying data distribution [145], a CAE penalises its sensitivity to the input by measuring the Frobenius norm of the Jacobian $J_f(x)$ of the mapping in Eq. (2.3.2) [144]. This is achieved using the following objective function

$$
\mathcal{L}_{\text{CAE}}(\theta) = \sum_{x \in \mathcal{X}} L(x, x') + \lambda \| J_f(x) \|_F^2,
$$

(6.3)
where $\lambda$ is a hyperparameter that controls the strength of the regularisation. The added penalty term $\| J_f(x) \|_F^2$ encourages the mapping to the feature space to be as insensitive as possible to changes in the input.

The added penalty, called the contraction penalty, reduces the number of effective degrees of freedom of the learnt representation $h(z)$ by making the encoder $f(z)$ contractive, i.e., making its partial derivative $\| \frac{\partial f(z)}{\partial z} \|_F^2$ small. As a consequence, the objective function encourages the learnt representation $h(z)$ to be as invariant as possible to small perturbations in the input $z$, while it is still able to reconstruct different training examples when it is combined with the reconstruction error. The resulting representation is faithful to variations in the input space in the directions of the manifold near which training examples are more concentrated, but it is highly contractive in the other directions.

Our IRMP approach exploits the invariance of CAE to small variations of the input around the training examples, to train an anomaly detection model from the independently perturbed records that are provided by the participants. The advantages of using CAE for our privacy-preserving scheme are twofold: first, it enhances the accuracy of anomaly detection; second, it makes the scheme invariant to arbitrary changes in the transformation matrices. This enables each participant $u_i$ to independently generate their own random perturbation matrix $T_i$. Note that while we present the use of a CAE in our architecture IRMP, similar results can be achieved using a denoising autoencoder (DAE) [171].

### 6.3 Simulations and Comparison

This section evaluates the performance and quality of the proposed privacy-preserving anomaly detection scheme IRMP in terms of accuracy, scalability and efficiency. The experiments on accuracy measure the trade-off in the accuracy of our anomaly detection algorithm as a result of maintaining the participants’ privacy (in Section 6.3.1). They make a comparison between the performance of CAE, AE and 1SVM [165] — one of the most popular unsupervised anomaly detection algorithms, which generates a nonlinear decision boundary by mapping the data to a higher dimension space. The scalability
of IRMP is evaluated with regards to the number of participants (in Section 6.3.2) and its efficiency is assessed with respect to the training/testing time of the algorithm (in Section 6.3.3).

**Experimental Setup:** For the one-class SVM the `svdd` command from Dd-tools [166] is used, and its parameters, width $\nu$ (from 0 to 1), and $\sigma$ (from 1 to $\infty$), are selected via a grid-search. The autoencoders, AE and CAE, are implemented in MATLAB with mini-batch learning, following the original scheme of Rifai et al. [145]. In our experiments, a five-layered autoencoder, with tied weights and a sigmoid activation function for both the encoder and decoder is used. Initially an autoencoder was trained based on greedy layer-wise pre-training (i.e., training one layer at a time) to extract features, and then using these features to train the next layer. Once the network was trained, the learnt parameter values (weights and bias) were used as initialisation values of a multilayer perceptron with the same number of inputs and outputs. Then the network was fine-tuned by gradient descent to adjust the weights and minimise the reconstruction error $l$. The hyperparameters of the autoencoders, learning rate (for pretraining 0.001–0.01, for fine tuning 0.1–1), number of epochs (for pretraining 5–10, for fine tuning 10–30), and number of hidden units ($h \leq n$), are set based on the best performance on a validation set. Anomalies can be identified by the autoencoder based on the integrated squared error between the inputs and outputs of the training records. Note that the whole training process in the one-class SVM and autoencoders is performed in an unsupervised manner for anomaly detection, and the labeled data is only used for testing.

**Datasets:** Experiments are conducted on seven real datasets from the UCI Machine Learning Repository: (i) Abalone, (ii) Diabetes 130-US hospitals for years 1999-2008 dataset (Diabetes), (iii) Forest, (iv) Adult, (v) Gas Sensor Array Drift (Gas), (vi) Opportunity Activity Recognition (OAR), (vii) Daily and Sport Activity (DSA), and (viii) Human Activity Recognition using Smartphones (HAR), with dimensionalities of 8, 45$^1$, 54, 123, 128, 242, 315$^2$, and 561 features, respectively. We also use three synthetic datasets. One

---

$^1$The Diabetes dataset includes over 50 attributes, but our evaluation was conducted on only 45 attributes. Some features such as encounter ID, and patient number are excluded due to uninformative information, and some feature such as weight, payer code, and medical speciality are excluded due to a large proportion of missing values (more than 50%).

$^2$DSA is a large dataset comprising the time series measurements from 45 wearable sensors for 19 activi-
is the “Smiley” dataset, generated from a mixture of two compact Gaussians and an arc shaped distribution, resembling a smiley-face. The dataset contains 100 dimensions, and in any two dimensions the components of the face are randomly moved. The other is the “GME” dataset that is a mixture of four separated Gaussians in 100 dimensions. Finally, we use the “Banana” dataset, which is a mixture of two banana shaped distributions randomly moved in 100 dimensions.

The datasets are normalised between [0,1], and 80% of the records are used for training and 20% for testing. Training and testing records are respectively, mixed with 5% and 20% (of the dataset size) anomalies uniformly scattered in [0,1]. In all experiments, except those stated, the first 2000 records of each are horizontally partitioned among the participants in batches of 30 records (one participant gets less than 30 records), which corresponds to around 70 participants. Note that some of the experiments are conducted with other numbers of records, participants and batch sizes.

Visualisation tool: For visualisation purposes, a tool called improved Visual Assessment of cluster Tendency (iVAT) [174] is applied to help visualise the possible number of clusters in, or the cluster tendency of, a set of objects. iVAT reorders the dissimilarity matrix of a given set of objects so that it can display any clusters as dark blocks along the diagonal of the image.

### 6.3.1 Accuracy Evaluation

Comparing the impact of tanh, double logistic and poly7 on accuracy

To investigate the impact of the studied nonlinear transformations on accuracy, the transformation matrix $T$ is kept constant (for all participants), and one of the nonlinear functions is used at each trial to perturb the datasets. The detailed results were not included, since they show that all the nonlinear functions are comparable (within 2%). More specifically, poly7 gives a slight improvement in accuracy compared to the two other approaches (tanh and double logistic function).
Comparing the performance of CAE with AE and 1SVM with the IRMP two-stage perturbation

To study the performance of these methods, two experiments are conducted. In the first experiment, the nonlinear function is set to $N = \text{poly7}$ with the coefficients stated in Table 5.1, while different $T_i \sim U(0,1)$ are randomly chosen for each participant $u_i$. In the second experiment, in addition to varying the transformation matrix $T$, the coefficients of the polynomial function are also randomly modified for each participant. The coefficients are combined with some random noise $a_j + \delta$, for $j = 0, \ldots, 7$, where $\delta$ is randomly drawn from a Gaussian distribution $N(0,1)$. Table 6.1 compares the performance of the 1SVM, AE and CAE anomaly detectors on the unperturbed data (“Raw”) along with the corresponding results using the two-staged perturbation in the above experiments.

Under the first setting that uses a fixed nonlinear function $N$ and a random linear transformation $T$, denoted with $T^*$ in Table 6.1, CAE demonstrates the best performance, followed by AE. The AUC of CAE shows little sign of being affected by using an independent transformation for each participant. Specifically the difference in AUC between the
Figure 6.2: Comparing the impact of different transformations on normal and anomalous records.
results on the raw data and those on the data with a random linear transformation is less than 1%. As expected, 1SVM suffers the most from this change. The experimental results on the benchmark datasets show that for all the datasets, except GME, 1SVM experiences a substantial loss in accuracy — more than 10% for some datasets (i.e., datasets DSA and HAR).

The second experiment considers the effect of randomising both $T$ and $\mathcal{N}$, shown under the heading $(T', \mathcal{N}')$ in Table 6.1, as opposed to randomising $T$ and using a constant $\mathcal{N}$. While randomising the coefficients of the nonlinear transformation $\mathcal{N}$ might be desirable, e.g., to increase privacy, it is interesting to assess its impact on accuracy. Similar to the previous experiment, imposing noise on $\mathcal{N}$, in the manner described earlier, has little effect on the accuracy of CAE$^3$ i.e., reducing the accuracy by about 1% compared to using a fixed $\mathcal{N}$. However, this is not the case for the two other methods. The accuracy of AE drops by 4%, whereas 1SVM experiences a much larger decrease in AUC (up to 8%).

Comparing the impact of different random transformations on the dataset

To better understand the effect of random transformation and the reason behind the accuracy loss in the above experiments, iVAT is used. Figure 6.2 shows the iVAT images of two datasets (for brevity the images of the other datasets are not shown), Banana and HAR, with the following order from the top: raw dataset, transformed dataset with constant $T$ and $\mathcal{N}$, transformed dataset with random $T$ and constant $\mathcal{N}$, and transformed dataset with random $T$ and $\mathcal{N}$ ($T', \mathcal{N}'$). As can be seen in the image of the raw datasets, normal records (first 1000 records) appear in dense blocks, while the anomalous records (last 50 records) are shown in gray shadow (since they are distributed across the dataset). Projecting the data to a lower dimensional space results in a concentration of the data around its mean $^4$. Looking at the iVAT images of the transformed data, a small cluster appeared as a result of the transformation. This effect can be advantageous for anomaly detection, since it separates normal records from anomalies. Transforming data with the same $T$ and $\mathcal{N}$ is more beneficial as a clearer separation appears between the normal records

$^3$Although we present only the AUC results of varying the poly7’s coefficients, in our empirical experiments we obtained similar results with different nonlinear functions, e.g., higher or lower degree polynomial functions, in conjunction with CAE.
Figure 6.3: Impact of dimensionality reduction in terms of size of $w$ on AUC.

and anomalies, as reflected in the improved clarity and contrast in the block structure. However, the concern arises when random perturbations ($T^*$ or $N^*$) are adopted. The separation still appears, but the randomness of the perturbations introduces significant noise to the data. In this case only those anomaly detection methods that are more resistant to noise effects, e.g., CAE, overcome this challenge, following the discussion in Section 6.2.2.

**Impact of the size of $w$ on accuracy**

Reducing the data dimensionality from $n$ to $w \leq \frac{(n+1)}{2}$ ensures that no linear filter can separate the transformed data. On the other hand, this raises the concern of accuracy loss. Figure 6.3 illustrates the impact of reducing $w$ (i.e., the number of dimensions in the perturbed dataset) on the accuracy of anomaly detection. The x-axis shows the percentage reduction in the number of dimensions $w$, and the y-axis shows the corresponding accuracy. As demonstrated in the figure, reducing the data dimensionality by 50% decreases the detection rate by less than 2%.

**6.3.2 Scalability Evaluation**

IRMP scales to any number of participants and clients without concern of collusion attack, or loss of accuracy. The property that makes randomisation approaches appeal-
Figure 6.4: Influence of number of participants on AUC.

Figure 6.5: Influence of data dimensionality on training and testing time.

ing for participatory sensing applications is that inherently they are not bounded by the number of participants and clients. In existing schemes, e.g., [23, 42, 107, 114], due to the vulnerability of the underlying privacy scheme against collusion attacks, participants are enforced to apply the same perturbation matrix to their data. Although this requirement restricts the usage of such schemes to parties who are aware of the perturbation matrix, it does not imply that randomisation approaches themselves suffer from scalability issues. Instead it indicates that privacy-preserving schemes built on a single perturbation matrix are applicable only if all the collaborating parties are trusted (semi-honest), which may not be a valid assumption in more real-life scenarios like participatory sensing. In addition to privacy-related issues, we need to consider whether using multiple perturbation matrices is feasible in terms of any potential accuracy loss, due to the noise introduced into the data, as shown in Figure 6.2.
Results from Table 6.1 demonstrate that the accuracy of IRMP is not significantly influenced by the variation in transformation matrices. Note that these experiments were conducted on 2,000 records, which corresponds to about 70 participants. A remaining question is whether IRMP maintains its performance for greater numbers of participants. A direct effect of increasing the number of participants may be reflected in the AUC of the generated model. Figure 6.4 studies this matter for a medium size network with 100 to 1,500 participants (contributing 3,000 to 45,000 records), and a large size network with 50,000 to 500,000 participants (contributing 150,000 to 1,500,000 records). In this experiment two medium size datasets, Forest and Adult, and two larger datasets, Smiley and Banana are used. As shown in Figure 6.4, this experiment demonstrates that the AUC values for all the datasets remains more or less constant.

### 6.3.3 Efficiency Evaluation

A major advantage of IRMP, in addition to privacy, accuracy and scalability, is its efficiency. The two-stage perturbation for participants is a matrix multiplication with a complexity of \( O(mnw) \), a simple computation that can be executed in resource constrained sensors. Since anomaly detection is our main concern, we are not interested in preserving the distance among different classes, hence the dimensionality of the records can be dramatically reduced \( w \ll n \). Reducing the data dimensionality clearly lessens the communication cost between participants and the server. Moreover, projecting the data to
6.4 Discussion and Conclusions

Collaborative anomaly detection schemes mainly deal with balancing the trade-off between accuracy and privacy. Large applications such as participatory sensing networks with resource constrained devices raise the challenges of scalability and efficiency as well. To address these challenges we proposed IRMP, a lightweight scalable anomaly detection scheme that preserves the privacy of both normal and anomalous records for numerical datasets. IRMP delivers these salient properties through a combination of nonlinear and linear perturbations. The nonlinear stage thwarts Bayesian estimation attacks, whereas the linear stage prevents independent component analysis, and collusion attacks.

For the nonlinear perturbation, a 7th-order polynomial function, poly7, was designed. The 7th-order polynomial curve has been chosen since the number of constraints, in the form of Eq. (5.7) to Eq. (5.14), have been empirically found to be effective in hampering Bayesian estimation attacks. The eight constraints that define the 7th-degree polynomial play the following roles: constraints Eq. (5.7) and Eq. (5.8) give poly7 the slope of zero at the origin for protecting near-origin anomalies; constraints Eq. (5.9) to Eq. (5.12) make poly7 hard to invert for normal data; and constraints Eq. (5.13) and Eq. (5.14) give poly7 a slope that resembles the slope of \( \tanh(\beta_t x) \) at \( x = 1 \). For the linear transformation, a perturbation matrix is drawn from \( \mathcal{U}(0, 1) \). This matrix projects the data to a lower dimensional space to prevent ICA attacks, by creating an ill-posed problem in the form of an under-determined system of linear equations. To thwart collusion among parties, par-
participants randomly draw their individual perturbation matrix, unlike current schemes that require the same data transformation to be used among all participants. Additionally, poly7 avoids distance inference attacks since neither the nonlinear function nor the linear transformation aim for distance preservation.

Our empirical analysis on Bayesian estimation attacks suggests that the two-stage transformation, poly7+RT, maintains privacy of normal and anomalous records, though the support for normal records is somewhat stronger than it is when the data are anomalous. Poly7 delivers the lowest recovery rate for normal records, but is slightly less effective than tanh for anomalous records in low dimensional datasets. Achieving zero or near-zero recovery rates for most of the studied cases suggests that poly7+RT is the best recovery-resistant scheme among the evaluated approaches.

A distinctive property of IRMP is its ability to deliver comparable accuracy to non-privacy preserving schemes, despite the individually random data perturbation. The effect of applying random perturbations can be observed from the various visualisations and performance evaluations that we made. Individual random transformations introduce significant noise to the data (see Fig. 6.2), which challenges the efficiency of many anomaly detection techniques with shallow architectures, such as one-class SVMs. Alternatively, IRMP adopts a contractive autoencoder since its objective function encourages the learnt representation to be as invariant as possible to small perturbations in the input data. Our accuracy and scalability evaluation determine the suitability of CAE as an anomaly detector for IRMP. It maintains the accuracy level to be comparable to non-privacy schemes even for networks with large numbers of participants and large volumes of records.

The idea of using individually random perturbation is attractive because it minimises the risk of disclosures by semi-honest parties. Therefore we aim to extend the framework of IRMP to other data types and data mining algorithms, e.g., categorical data, time series datasets, and multi-class classification algorithms.
Part II

High-Dimensional and Large-Scale Anomaly Detection
Introduction to Part II

In the second part of this thesis, we address the problem of high-dimensional anomaly detection, the second challenge introduced in Chapter 1. Chapter 7 presents a hybrid anomaly detection architecture based on a combination of a deep belief net and a one-class SVM. Chapter 8 presents an unsupervised anomaly detection technique that approximates a nonlinear 1SVM by modifying the original linear 1SVM method based on a randomised nonlinear projection of the data. Chapter 9 summarises the thesis, and highlights a number of possible directions for future research that arise from our work.
Chapter 7
High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning

In this chapter, we focus on the problem of anomaly detection in high-dimensional problem domains. The presence of irrelevant features can conceal the presence of anomalies. This problem, known as the ‘curse of dimensionality’, is an obstacle for many anomaly detection techniques. Building a robust anomaly detection model for use in high-dimensional spaces requires the combination of an unsupervised feature extractor and an anomaly detector. While one-class support vector machines are effective at producing decision surfaces from well-behaved feature vectors, they can be inefficient at modelling the variations in large, high-dimensional datasets. Architectures such as deep belief networks (DBNs) are a promising technique for learning robust features. We present a hybrid model where an unsupervised DBN is trained to extract generic underlying features, and a one-class SVM is trained from the features learned by the DBN. Since a linear kernel can be substituted for nonlinear ones in our hybrid model without loss of accuracy, our model is scalable and computationally efficient.

By means of various statistical analyses we demonstrate the synergy obtained from the combination of the deep architecture and a one-class SVM. More specifically the experiments show that the proposed model yields comparable anomaly detection performance with a deep autoencoder, while reducing its training and testing time by a factor of 3 and 1000, respectively.

The work arising from this chapter is under review as the paper P 7.

7.1 Introduction

A key challenge in the development of participatory sensing applications is how to model and interpret the large volumes of high-dimensional data that are generated in such do-
High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning

mains [80]. Further, the lack of ground truth (labels) in the data that are collected from large-scale networks in participatory applications require unsupervised algorithms to process the data. Anomaly detection aims to detect unusual behaviours caused by either faulty devices or events of interest in the monitoring environment, and thus is of great importance in participatory applications. However, a major challenge for anomaly detection in such domains is how to cope with noisy, large-scale datasets [100]. In this work we address this challenge by proposing an unsupervised hybrid architecture for anomaly detection in large-scale high-dimensional problem domains.

As elaborated in Section 2.3.3, high-dimensional datasets pose a challenge for anomaly detection due to various factors [190] such as: (i) Exponential search space (ii) Data-snooping bias (iii) Irrelevant features. Our objective is to find a large-scale, high-dimensional anomaly detection algorithm that is robust, i.e., generates an accurate model for data drawn from a wide range of probability distributions, and is not unduly affected by small departures from the trained model. In addition, it is desirable that the algorithm be efficient in terms of time complexity, memory complexity and the required number of labelled records.

One-class Support Vector Machines (1SVMs) [150, 165, 173] are a popular technique for unsupervised anomaly detection. Generally, they aim to model the underlying distribution of normal data while being insensitive to noise or anomalies in the training records. A kernel function implicitly maps the input space to a higher dimensional feature space to make a clearer separation between normal and anomalous data. When properly applied, in principle a kernel-based method is able to model any non-linear pattern of normal behaviour. For clarity in the rest of the chapter, the notation of 1SVM is used to denote (an unsupervised) one-class SVM; lSVMs— short for labeled SVM — to denote (supervised) binary and multi-class SVM classifiers; and SVMs when both 1SVMs and lSVM are considered.

SVMs are theoretically appealing for the following reasons [1,12]: they provide good generalisation when the parameters are appropriately configured, even if the training set has some bias; they deliver a unique solution, since the loss function is convex; and in principal they can model any training set, when an appropriate kernel is chosen.
In practice, however, training SVMs is memory and time intensive. SVMs are non-parametric learning models, whose complexity grows quadratically with the number of records [170]. Thus, they are best suited to small datasets with many features. Moreover, large numbers of input features result in the curse of dimensionality phenomenon, which causes the generalisation error of SVMs to increase with the number of irrelevant and redundant features. The curse of dimensionality implies that to obtain good generalisation, the number of training samples must grow exponentially with the number of features. So far large-scale training on high-dimensional records has been limited with SVMs [90]. Furthermore, SVMs are a shallow architecture (discussed in Section 2.3.4), i.e., they have practical limitations for efficient representation of certain types of function families [21].

An alternative class of classification algorithms that have emerged in recent years are Deep Belief Nets (DBNs), which have been proposed as a multi-class classifier and dimensionality reduction tool [87, 89]. DBNs are multi-layer generative models that learn one layer of features at a time from unlabelled data. The extracted features are then treated as the input for training the next layer. This efficient, greedy learning can be followed by fine-tuning the weights to improve the generative or discriminative performance of the whole network.

DBNs have a deep architecture, composed of multiple layers of parameterised non-linear modules. There are a range of advantageous properties that have been identified for DBNs [21]: they can learn higher-level features that yield good classification accuracy; their training time scales linearly with the number of records; and they can use unlabelled data to learn from complex and high-dimensional datasets.

A major limitation of DBNs is that their loss function is non-convex. Therefore the model often converges on local minima and there is no guarantee that the global minimum will be found. In addition, DBN classifiers are semi-supervised algorithms, and require some labelled examples for discriminative fine-tuning, hence an unsupervised generative model of DBNs, known as autoencoders, are used for anomaly detection.

The open research problem we address is how to overcome the limitations of one-class SVM architectures on complex, high-dimensional datasets. We propose the use of
DBNs as a feature reduction stage for one-class SVMs, to give a hybrid anomaly detection architecture. While a variety of feature reduction methods — i.e., feature selection and feature extraction methods — have been considered for SVMs (e.g., [36, 130, 153, 177, 178] — see [81] for a survey) none have studied the use of DBNs as a method for deep feature construction in the context of anomaly detection, i.e., with a one-class SVM. In this chapter, we design and evaluate a new architecture for anomaly detection in high-dimensional domains. To the best of our knowledge, this is the first method proposed for combining DBNs with one-class SVMs to improve their performance for anomaly detection.

The contributions of this chapter are two-fold. The performance of DBNs against one-class SVMs is evaluated for detecting anomalies in complex high-dimensional data. In contrast, the reported results in the literature from DBN classification performance only cover multi-class classification, e.g., [14, 82, 90, 115]. A novel unsupervised anomaly detection model is also proposed, which combines the advantages of deep belief nets with one-class SVMs. In our proposed model an unsupervised DBN is trained to extract features that are reasonably insensitive to irrelevant variations in the input, and a 1SVM is trained on the feature vectors produced by the DBN. More specifically, for anomaly detection we show that computationally expensive non-linear kernel machines can be replaced by linear ones, when aggregated with a DBN. To the best of our knowledge, this is the first time these frameworks have been combined this way. The results of experiments conducted on several benchmark datasets demonstrate that our hybrid model yields significant performance improvements over the stand-alone systems. The combination of the hybrid DBN-1SVM avoids the complexity of non-linear kernel machines, and reaches the accuracy of a state-of-the-art autoencoder while considerably lowering its training and testing time.

The remainder of the chapter is organised as follows. Section 7.2 gives a background to DBNs and motivates the requirements for the hybrid model by considering the shortcomings of SVMs for processing large datasets. Section 7.3 presents our proposed unsupervised anomaly detection approach DBN-1SVM. Section 7.4 describes the empirical analysis and provides a detailed statistical comparison of the performance of autoen-
7.2 Background DBN and Hybrid DBN-SVM

In Section 2.3 we gave an introduction to deep architectures, their strengths and weaknesses compared to their shallow counterparts. We also reviewed some of the leading 1SVM methods. In this section we motivate the need for the hybrid model by considering the shortcomings of SVMs in processing large datasets.

In contrast to SVMs, deep belief nets are a relatively new type of multi-layer neural network. A DBN is trained in an unsupervised way to learn a hierarchical representation of the training data, which corresponds to a high-dimensional manifold. DBNs are trained one layer at a time, i.e., the latent variables at each layer are treated as the input for training the next layer, as discussed in Section 2.3.2. This efficient, greedy layer-wise training [20] can be followed by a stage of supervised fine-tuning, e.g., adding softmax or a logistic regression classifier, to improve the discriminative performance of the network. Semi-supervised DBNs are commonly used for multi-class classification. These methods have been shown to exhibit low complexity and high classification performance on complex datasets, e.g., for electroencephalography waveform clarification [181] and 3D object
High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning

recognition [129], in comparison to other (shallow) classifiers such as SVMs.

Deep belief nets have been demonstrated to be effective at learning invariant features from complex and high-dimensional datasets. SVMs, on the other hand, with their fixed kernel function can face difficulties in learning complicated invariances, but can learn robust decision surfaces when applied to well-behaved feature vectors. Given these complementary strengths, it is appealing to investigate the scope for using a hybrid model of these two architectures for anomaly detection.

Hybrid DBN-SVM methods (i.e., a DBN in combination with labelled SVMs such as binary or multiclass SVMs) have been used in some application domains, such as object classification [90], text classification [115], music classification [82], face expression recognition [163], and speech separation [175]. These studies have observed that a hybrid model yields noticeable performance improvement over the stand-alone methods for supervised classification, but have not considered unsupervised anomaly detection. The next section presents our hybrid model that combines DBN and one-class SVM architectures, called DBN-1SVM, for unsupervised anomaly detection.

7.3 DBN-1SVM Hybrid Model

To benefit the complementary strengths of DBNs and kernel machines, we propose a novel unsupervised hybrid architecture DBN-1SVM (see Figure 7.1), in which a DBN is trained to extract features that are relatively invariant to irrelevant variations in the input, so that the 1SVM can effectively separate the normal data from anomalies in the learned feature space. A DBN is trained as a non-linear dimensionality reduction algorithm to transform the high-dimensional data into a low-dimensional set of features [87]. The derived feature sets from the training samples form the input to train the one-class SVM. Subsequently, a hybrid of the generated models from these two algorithms constructs the ultimate anomaly detection model and can be used for testing. Our experiments show that this hybrid system not only improves the detection rate significantly, but also alleviates the computational complexity of training and testing.

In the following, we first describe the use of the DBN as a dimensionality reduction
7.3 DBN-1SVM Hybrid Model

algorithm, and then elaborate on how the output of the DBN can be taken as input for a one-class SVM. Based on the explanation below, two of the most common 1SVM algorithms are chosen, a hypersphere-based 1SVM (known as Support Vector Data Description (SVDD)) by Tax and Duin [165], and a Plane-based 1SVM (PSVM) by Schölkopf et al. [149], see Figure 2.2; correspondingly, their hybrid models are referred to as a DBN-SVDD (DSVDD) and a DBN-PSVM (DPSVM), respectively. Note that the other types of 1SVMs that are described in Section 2.3.2 can also be applied in a similar way to our hybrid model.

7.3.1 Deep Belief Nets (DBNs)

DBNs are multi-layer generative models that learn one layer of features at a time from unlabelled data. Two significant properties of DBNs are their ability to perform non-linear dimensionality reduction on very large datasets and to learn high-dimensional manifolds from the data.

A DBN can be trained efficiently in a greedy layer-wise fashion by using a Restricted Boltzmann Machine (RBM). An RBM is a bipartite graph, with visible units \( v \) representing observations and hidden units \( h \) learning to represent features, i.e., it maps the input vectors \( v \) from an input space of dimension \( n \) to a feature space of dimension \( d = |h| \), where \( d < n \). Given a dataset \( D_{m \times n} \) as input, an RBM maps it to \( X_{m \times d} \). RBMs are restricted in the sense that there are no connections between units at the same level, i.e., visible-visible or hidden-hidden connections, and the two layers of the graph are connected with symmetric weights \( W \) between pairs of hidden and visible units.

As in the original Boltzmann machine architecture, the joint distributions over hidden and visible vectors \( p(v, h) \) are defined in terms of an energy function \( E(v, h) \), as \( p(v, h) = \frac{e^{-E(v,h)}}{Z} \), where \( Z \) is a normalisation factor called the partition function and is calculated as \( Z = \sum_{v, h} e^{-E(v,h)} \). The energy of the joint configuration is determined with respect to the values of the network parameters \( \theta = (W, b, c) \), where \( b \) and \( c \) are biases to the hidden layer and visible layers, respectively. The hidden layer \( h \) is binary and hidden units are Bernoulli random variables, whereas the input units can be either binary or real-valued. Assuming the input vectors are Gaussian random variables with variance \( \sigma \), the energy
function for this Gaussian-Bernoulli RBM can be obtained as:

$$E(v, h) = \frac{1}{2} \sum_i (v_i - c_i)^2 - \sum_j b_j h_j - \sum_{i,j} w_{i,j} v_i h_j,$$

(7.1)

where $v_i$ and $h_j$ are the $i$th and $j$th units of the visible $v$ and hidden $h$ layers with the symmetric weight of $w_{i,j}$, and the corresponding biases $c_i$ and $b_j$. Given a binary hidden unit $h_j$, since there is no connection between hidden units it is straightforward to calculate the conditional distribution $p(h|v)$,

$$p(h|v) = \prod_j p(h_j|v) = \prod_j \text{Sigm}(b_j + \sum_i w_{i,j} v_i),$$

(7.2)

and similarly since there is no connection between visible units, $p(v|h)$ factorises to

$$p(v|h) = \prod_i p(v_i|h) = \prod_i \mathcal{N}(c_i + \sum_j w_{i,j} h_j, \sigma),$$

(7.3)

where $\text{Sigm}(x) = \frac{1}{1+e^{-x}}$ is the logistic sigmoid function and $\mathcal{N}(\mu, \sigma)$ denotes a Gaussian distribution with mean $\mu$ and variance $\sigma$.

Training an RBM implies finding the values of the parameters $\theta$ such that the energy is minimised. One possible approach aims at maximising the log-likelihood of $v$ that is estimated by its gradient with respect to the model parameters,

$$\frac{\partial \log p(v)}{\partial \theta} = E_{p(h|v)} \left[ \frac{\partial E(v, h)}{\partial \theta} \right] - E_{p(v|h)} \left[ \frac{\partial E(h, v)}{\partial \theta} \right].$$

(7.4)

Since an exact calculation of the second term of the log-likelihood is intractable, the gradient can be estimated using a method known as Contrastive Divergence (CD) [88]. CD approximates the expectation through $k$ iterations of Gibbs sampling (often $k = 1$) to update the network weights, i.e.,

$$\frac{\partial \log p(v)}{\partial w_{i,j}} \approx \left\langle v_i h_j \right\rangle^0 - \left\langle v_i h_j \right\rangle^k,$$

(7.5)

where $\langle . \rangle^I$ represents the average value at contrastive divergence iteration $I$. After train-
7.3 DBN-1SVM Hybrid Model

An RBM, another RBM can be stacked on top of the first one, i.e., the inferred states of the hidden units, $X_{m \times d_2}$, are used as the visible units for training the new RBM. The upper layers can be a Bernoulli-Bernoulli RBM, in which the main difference to the first layer lies in the binary visible units and the energy function. Stacking RBMs enables us to model the significant dependencies between the hidden units of the earlier RBM. More specifically, multiple layers of RBMs can be stacked to produce different layers of non-linear feature detectors, which represent progressively more complex statistical structure in the data. In a stack of RBMs, the bottom-up recognition weights of the resulting DBN are used to initialise the weights of a multi-layer feed-forward neural network. This network can be employed as a tool for dimensionality reduction, or topped with a logistic regression layer and discriminatively fine-tuned by backpropagation for classification.

7.3.2 One-class SVM

For the second stage of our DBN-1SVM architecture, we avoid discriminative fine-tuning of the DBN and feed the output of the last hidden layer (bottleneck) $X$ to a one-class SVM (see Figure 7.1(c)).

Support Vector Data Description (SVDD)

SVDD essentially finds the smallest possible hypersphere around the majority of the training records, while leaving out some points to be excluded as anomalies. Given $X = \{x_l : l = 1, \ldots, m\}$ and $X \in \mathbb{R}^d$, SVDD finds the most tightly fitting hypersphere that encompasses most of the data points, as shown in Figure 2.2. Denoting the centre of the hypersphere by $a$ and its radius by $R$, this hypersphere formulation involves solving the following quadratic programming optimisation problem,

$$\begin{align*}
\min_{a, R, \xi} & \quad R^2 + \frac{1}{m} \sum_{l=1}^{m} \xi_l \\
\text{s.t.} & \quad \| \phi(x_l) - a \|^2 \leq R^2 + \xi_l, \\
& \quad \forall l = 1, \ldots, m, \xi_l \geq 0.
\end{align*}$$

(7.6)
The term $\phi(.)$ is a non-linear function that maps data to a higher dimensional space $\mathbb{R}^d \to \mathbb{R}^q, d < q$. The term $\nu$ is a user predefined regularisation parameter that governs the trade-off between the size of the hypersphere and the fraction of data points falling outside the hypersphere, i.e., the fraction of training examples that can be classified as anomalies. The terms $\xi_l, l = 1, \ldots, m$, are the slack variables that allow some of the data points to lie outside the hypersphere.

Let $\alpha = [\alpha_1, \ldots, \alpha_m]^T$ and $0 \leq \alpha_l \leq \frac{1}{\nu m}$. The dual problem for the above primary problem in (7.6) is as follows

$$
\max_{\alpha} \sum_{l=1}^{m} \alpha_l (x_l \cdot x_l) - \sum_{l,t} \alpha_l \alpha_t (x_l \cdot x_t),
$$
$$\forall 1 \leq l, t \leq m,
$$
$$s.t. \quad 0 \leq \alpha_l \leq \frac{1}{\nu m}.
$$

Maximising this optimisation (7.7) gives a set of $\alpha_l$. For the training samples $x_l$ that satisfy the inequality $\| \phi(x_l) - a \|^2 < R^2 + \xi_l$, the corresponding Lagrange multiplier will be zero, i.e., $\alpha_l = 0$. For examples that satisfy the equality $\| \phi(x_l) - a \|^2 = R^2 + \xi_l$, the multiplier will become greater than zero, i.e., $\alpha > 0$. In summary,

$$
\| \phi(x_l) - a \|^2 < R^2 + \xi_l, \quad \alpha = 0 \quad \text{inlier},
$$
$$
= R^2 + \xi_l, \quad 0 < \alpha < \frac{1}{\nu m} \quad \text{border SVs},
$$
$$
> R^2 + \xi_l, \quad \alpha = \frac{1}{\nu m} \quad \text{outlier}.
$$

The centre of the hypersphere is a linear combination of the training examples, i.e., $a = \sum (a_i, x_i)$. Among the input vectors $x_i$, only those with $a_l > 0$ (the support vectors (SVs)) are required. After solving the above optimisation problem, a test instance $z$ is detected as an anomaly if its distance to $a$ is larger than the radius $R$,

$$
\| \phi(z) - a \|^2 > R^2,
$$
where $\| \phi(z) - a \|^2 = (z \cdot z) - 2 \sum a_l (z \cdot x_l) + \sum a_l a_t (x_l \cdot x_t)$.
7.3 DBN-1SVM Hybrid Model

Plane-based One-class Support Vector Machine (PSVM)

An alternative to SVDD for the hybrid model is a plane-base 1SVM [122, 149]. PSVM identifies anomalies in the feature space by finding a hyperplane that best separates the data from the origin. In other words, the decision function of PSVM returns +1 in a region where most of the data points occur (i.e., where the probability density is high), and returns −1 elsewhere. To separate the data from the origin, PSVM solves the following quadratic optimisation function:

\[
\begin{align*}
\min_{s, \xi, \rho} & \quad \frac{1}{2} \|s\|^2 + \frac{1}{m\nu} \sum_{i=1}^{m} \xi_i - \rho \\
\text{s.t.} & \quad (s.x_i) \geq \rho - \xi_i, \\
& \quad \forall l = 1, \ldots, m, \xi_l \geq 0.
\end{align*}
\]  

(7.8)

Since the non-zero slack variables \( \xi_l \) are penalised in the objective function, the PSVM estimates a decision function \( f_{s,\rho}(x) = \text{sgn}(s.\phi(x) - \rho) \) that maximises the distance of all the data points (in the feature space) from the hyperplane to the origin, parameterised by a weight vector \( s \) and an offset \( \rho \).

By introducing the Lagrange multipliers and setting the primal variables \( s, \xi \) and \( \rho \) equal to zero, the quadratic program can be derived as the dual of the primal program in Eq. (7.8):

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \sum_{i,j} k(x_i, x_j) \\
\text{s.t.} & \quad 0 \leq \alpha_i \leq \frac{1}{m\nu}, \sum_i \alpha_i = 1.
\end{align*}
\]  

(7.9)

Using the Karush-Kuhn-Tucker optimality conditions (KKT conditions) the data vectors can be characterised in terms of whether they fall below, above, or on the hyperplane boundary in the feature space depending on the corresponding \( \alpha_i \) values. Data vectors with positive \( \alpha_i \) values are the support vectors. Further, for \( 0 < \alpha_i < 1/\nu n \), the data vectors fall on the hyperplane and hence \( \rho \) can be recovered using these vectors, vis-a-vis
\[ \rho = \langle s, \phi(x_i) \rangle = \sum_j \alpha_j k(x_j, x_i). \] Therefore, the decision function can now be written as

\[ f_{s,\rho}(x) = \text{sgn}(s, \phi(x) - \rho) = \text{sgn}(\alpha_i k(x_i, x) - \rho). \quad (7.10) \]

### 7.4 Evaluation and Discussion

This section evaluates the performance of the proposed hybrid approaches through various experiments. More specifically the aims of these experiments are as follows:

- Evaluate the effect of kernel choice on the performance of 1SVM-based methods (Section 7.4.2).
- Compare the performance of DBN-1SVMs against standalone approaches (i.e., autoencoder, SVDD and PSVM), as well as PCA-SVDD and PCA-PSVM (a hybrid model consisting of Principle Component Analysis (PCA) and a 1SVM) on several high-dimensional datasets. Note that PCA is used as a reference feature extraction algorithm as it is one of the most prevalent approaches used in conjunction with SVMs\(^1\) (Section 7.4.3).
- Measure the effect of the network depth (number of hidden layers) on the performance of DBN-1SVMs (Section 7.4.4).
- Determine the effect of the number of training records on the performance of the hybrid model (Section 7.4.5).
- Assess the time- and memory-complexity of the hybrid models (Section 7.4.6).

#### 7.4.1 Experimental Methodology

**Experimental setup:** DBN-based algorithms are implemented in MATLAB with mini-batch learning, following the original scheme in [87, 89]. Each RBM was first individually

\(^1\) Kernel-PCA (KPCA) is another well-known nonlinear approach that can also be used as a dimensionality reduction tool. However, similar to other kernel-machines, KPCA suffers from high computational complexity (\(O(N^3)\)), i.e., it starts by projecting data to a higher dimensional space. Moreover, the main difficulty in KPCA is the choice of an appropriate kernel and the corresponding parameters, where choosing an inappropriate kernel may lead to an increase in dimensionality [106].
trained using one step Contrastive Divergence \( (k = 1) \) \[^2\text{88}\], to extract features, and used the features to train the next layer. Once the network was trained, as described in Section 5.4, the learned parameter values were set as initialisation values of a multilayer perceptron and fine-tuned with backpropagation to minimise the mean-squared error between the inputs and outputs of the network, i.e., an autoencoder (AE). When the network was trained, its bottom half (i.e., a DBN) is used to extract feature sets, which are then taken as input to train a one-class SVM in the usual way. For the one-class SVMs (PSVM and SVDD) LIBSVM \[^3\text{38}\] is employed, and for the PCA, the standard native MATLAB command \texttt{princomp} is used.

The hyperparameters of DBN-based networks, learning rate (for pretraining 0.001 – 0.01, for fine tuning 0.1 – 1), number of epochs (for pretraining 5 – 10, for fine tuning 10 – 30), number of hidden units \((d \ll n)\), are set based on the best performance on a validation set. The SVDD parameters, width \(c (0 – 1)\), and \(\sigma (1 – \infty)\), are selected via a grid-search. In the case of PCA the number of components \(\kappa\) is set such that 95\% of the variance is retained.

The training time for the hybrid methods, DBN-1SVMs and PCA-1SVMs, includes the time to train the dimensionality reduction algorithm, extracting features and training a 1SVM with the features. Note that in the following reported results, the value given after the headings of DBN-based methods (i.e., DBN1, DBN2, D1x, D2x, AE1 and AE2) indicates the number of hidden layers for the corresponding experiment.

The reported training/testing times are in seconds based on experiments run on a machine with an Intel Core i7 CPU at 3.40 GHz and 8 GB RAM. The stated AUC values and training/testing times are the average of 1000 iterations for each experiment.

\textbf{Datasets:} The experiments are conducted on six real-life datasets and two synthetic ones. The real-life datasets are from the UCI Machine Learning Repository: (i) Forest (ii) Adult (iii) Gas Sensor Array Drift (Gas), (iv) Opportunity Activity Recognition (OAR), (v) Daily and Sport Activity (DSA), and (vi) Human Activity Recognition using smartphones (HAR), with dimensionalities of 54, 123, 128, 242, 315, and 561 features, respectively. One synthetic dataset is the “Banana” dataset, generated from a mixture of

\[^2\text{In this work we following the original scheme in} \text{87–89}\text{ to train an RBM, but other approaches like} \text{28}\text{ can also be applied.}\]
High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning

two banana shaped distributions, which are randomly moved in 100 dimensions. The other is a “Smiley” dataset, generated from a mixture of two compact Gaussians and an arc shaped distribution, resembling an smiley-face. The dataset contains 500 dimensions and in any two dimensions the components of the face are randomly moved. All the records in each dataset are normalised between [0,1].

**Statistical analysis:** Statistical analysis is conducted to identify the statistical significance of the performance results obtained from the various methods. Non-parametric tests are chosen in this chapter, according to the recommendation made in [52, 73, 74], where their use in the field of machine learning is strongly recommended. These tests are conducted since the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose credibility with these parametric tests [52].

Throughout the experiments several non-parametric tests are considered. The Wilcoxon signed-rank test [179] is used to perform pairwise comparisons between different methods. The Iman-Davenport test [154] is used to detect statistical differences among a group of studied methods. If a significant difference was detected then a post-hoc test, the Shaffer test [152], is performed to identify which methods are distinctive in an all-by-all comparison. The pos-hoc test is conducted to examine if a hypothesis of comparison of means could be rejected at a specified significance level $\alpha$. Therefore, the $p$-value associated with each comparison is computed, representing the lowest level of significance of a hypothesis that results in a rejection. This value allows one to identify if two algorithms have significantly different performance and to what extent. For all the comparisons in this study the significance level $\alpha$ is set to 0.1.

7.4.2 Choice of Kernel

In Section 7.3 two types of one-class SVMs, hypersphere and hyperplane SVM, were presented for combination with DBN. Given that each of the methods can be used in conjunction with different kernels, their performance for two common kernels, linear and RBF, are evaluated. Table 7.1 shows the performance results and their average over all the datasets. The best case, i.e., the highest AUC and lowest training/testing time, for
Table 7.1: Performance results.

<table>
<thead>
<tr>
<th>Kernel Method</th>
<th>Metric</th>
<th>Forest</th>
<th>Banana</th>
<th>Adult</th>
<th>GAS</th>
<th>OAR</th>
<th>DSA</th>
<th>Smiley</th>
<th>HAR</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear PSVM</td>
<td>AUC</td>
<td>0.83</td>
<td>0.77</td>
<td>0.81</td>
<td>0.89</td>
<td>0.87</td>
<td>0.78</td>
<td>0.79</td>
<td>0.80</td>
<td>0.813</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.0046</td>
<td>0.0121</td>
<td>0.0073</td>
<td>0.0857</td>
<td>0.0722</td>
<td>0.0156</td>
<td>0.0605</td>
<td>0.079</td>
<td>0.0421</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.0018</td>
<td>0.0029</td>
<td>0.0036</td>
<td>0.0063</td>
<td>0.0067</td>
<td>0.0085</td>
<td>0.0012</td>
<td>0.0082</td>
<td>0.0049</td>
</tr>
<tr>
<td>RBF SVDD PSVM</td>
<td>AUC</td>
<td>0.83</td>
<td>0.78</td>
<td>0.81</td>
<td>0.89</td>
<td>0.87</td>
<td>0.79</td>
<td>0.79</td>
<td>0.80</td>
<td>0.823</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.0073</td>
<td>0.2505</td>
<td>0.1880</td>
<td>0.3175</td>
<td>0.2764</td>
<td>0.8314</td>
<td>1.3215</td>
<td>1.6288</td>
<td>0.6111</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.0019</td>
<td>0.0012</td>
<td>0.0015</td>
<td>0.0027</td>
<td>0.0004</td>
<td>0.0013</td>
<td>0.0019</td>
<td>0.0016</td>
<td>0.0016</td>
</tr>
<tr>
<td>Linear PCA PSVM</td>
<td>AUC</td>
<td>0.97</td>
<td>0.92</td>
<td>0.86</td>
<td>0.91</td>
<td>0.90</td>
<td>0.85</td>
<td>0.85</td>
<td>0.87</td>
<td>0.8912</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.0254</td>
<td>0.0464</td>
<td>0.0611</td>
<td>0.0675</td>
<td>0.0475</td>
<td>0.1516</td>
<td>0.2417</td>
<td>0.3548</td>
<td>0.1236</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.0091</td>
<td>0.0085</td>
<td>0.2866</td>
<td>0.0132</td>
<td>0.0090</td>
<td>0.0275</td>
<td>0.0422</td>
<td>0.0763</td>
<td>0.0268</td>
</tr>
<tr>
<td>RBF SVDD PSVM</td>
<td>AUC</td>
<td>0.97</td>
<td>0.92</td>
<td>0.87</td>
<td>0.91</td>
<td>0.91</td>
<td>0.84</td>
<td>0.86</td>
<td>0.88</td>
<td>0.8950</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.02</td>
<td>0.02</td>
<td>0.04</td>
<td>0.04</td>
<td>0.02</td>
<td>0.07</td>
<td>0.102</td>
<td>0.26</td>
<td>0.0715</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
<td>0.031</td>
<td>0.08</td>
<td>0.0227</td>
</tr>
<tr>
<td>Linear DBN2 PSVM</td>
<td>AUC</td>
<td>0.99</td>
<td>0.97</td>
<td>0.99</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.99</td>
<td>0.9837</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.3217</td>
<td>0.518</td>
<td>0.564</td>
<td>0.99</td>
<td>0.67</td>
<td>0.63</td>
<td>0.988</td>
<td>1.02</td>
<td>0.6526</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.0021</td>
<td>0.0035</td>
<td>0.0033</td>
<td>0.0029</td>
<td>0.0035</td>
<td>0.0041</td>
<td>0.0011</td>
<td>0.0054</td>
<td>0.0039</td>
</tr>
<tr>
<td>RBF DBN2 PSVM</td>
<td>AUC</td>
<td>0.96</td>
<td>0.94</td>
<td>0.88</td>
<td>0.95</td>
<td>0.96</td>
<td>0.91</td>
<td>0.85</td>
<td>0.92</td>
<td>0.9213</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.1034</td>
<td>0.0228</td>
<td>0.1320</td>
<td>0.1647</td>
<td>0.0452</td>
<td>0.3252</td>
<td>0.1689</td>
<td>0.4944</td>
<td>0.1821</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.0176</td>
<td>0.0011</td>
<td>0.0240</td>
<td>0.0338</td>
<td>0.0055</td>
<td>0.0554</td>
<td>0.0030</td>
<td>0.0713</td>
<td>0.0265</td>
</tr>
<tr>
<td>Linear DBN2 PSVM</td>
<td>AUC</td>
<td>0.96</td>
<td>0.94</td>
<td>0.89</td>
<td>0.95</td>
<td>0.96</td>
<td>0.91</td>
<td>0.85</td>
<td>0.92</td>
<td>0.9225</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.1156</td>
<td>0.1121</td>
<td>0.0654</td>
<td>0.0784</td>
<td>0.1352</td>
<td>0.2126</td>
<td>0.2791</td>
<td>0.349</td>
<td>0.1684</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.0088</td>
<td>0.0031</td>
<td>0.0018</td>
<td>0.0062</td>
<td>0.0063</td>
<td>0.0206</td>
<td>0.0259</td>
<td>0.0256</td>
<td>0.0123</td>
</tr>
<tr>
<td>RBF DBN2 PSVM</td>
<td>AUC</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.9838</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.0768</td>
<td>0.0815</td>
<td>0.0989</td>
<td>0.0969</td>
<td>0.0803</td>
<td>0.1495</td>
<td>0.2151</td>
<td>0.2416</td>
<td>0.1289</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.0082</td>
<td>0.0095</td>
<td>0.0091</td>
<td>0.0082</td>
<td>0.0167</td>
<td>0.0092</td>
<td>0.00914</td>
<td>0.0096</td>
<td>0.0099</td>
</tr>
<tr>
<td>Linear DBN2 PSVM</td>
<td>AUC</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.9838</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.0768</td>
<td>0.0815</td>
<td>0.0989</td>
<td>0.0969</td>
<td>0.0803</td>
<td>0.1495</td>
<td>0.2152</td>
<td>0.2416</td>
<td>0.1289</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.018</td>
<td>0.0397</td>
<td>0.145</td>
<td>0.5597</td>
<td>0.5076</td>
<td>0.4738</td>
<td>0.7589</td>
<td>1.1496</td>
<td>0.4565</td>
</tr>
<tr>
<td>RBF DBN2 PSVM</td>
<td>AUC</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.9813</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.0903</td>
<td>0.1174</td>
<td>0.1256</td>
<td>0.1048</td>
<td>0.1379</td>
<td>0.1715</td>
<td>0.2602</td>
<td>0.2404</td>
<td>0.1560</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.007</td>
<td>0.0105</td>
<td>0.0086</td>
<td>0.0067</td>
<td>0.0032</td>
<td>0.0068</td>
<td>0.0118</td>
<td>0.0088</td>
<td>0.0079</td>
</tr>
<tr>
<td>RBF DBN2 PSVM</td>
<td>AUC</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.9838</td>
</tr>
<tr>
<td></td>
<td>Train</td>
<td>0.0909</td>
<td>0.0974</td>
<td>0.1212</td>
<td>0.1075</td>
<td>0.1566</td>
<td>0.1881</td>
<td>0.3192</td>
<td>0.3186</td>
<td>0.1749</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.121</td>
<td>0.187</td>
<td>0.143</td>
<td>0.183</td>
<td>0.126</td>
<td>0.258</td>
<td>0.194</td>
<td>0.163</td>
<td>0.1719</td>
</tr>
</tbody>
</table>

Note: Training and testing times are in seconds, except for testing time rows marked with a *, which are in milliseconds. The results of the underscored methods are only considered in Section 7.4.2.
Table 7.2: Wilcoxon test to compare the performance of the PSVM and SVDD based methods with respect to the choice of kernel.

<table>
<thead>
<tr>
<th>PSVM vs. SVDD</th>
<th>Accuracy</th>
<th>Training Time</th>
<th>Test Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$R^+$</td>
<td>$R^-$</td>
<td>$p$-value</td>
</tr>
<tr>
<td>Linear</td>
<td>3</td>
<td>15</td>
<td>0.1250</td>
</tr>
<tr>
<td>RBF</td>
<td>3</td>
<td>7</td>
<td>0.3438</td>
</tr>
</tbody>
</table>

Note: $R^+$ corresponds to the sum of the ranks for the method on the left and $R^-$ for the right.

Each dataset is stressed through **bold-face**.

Before taking the study further, to improve the clarity in the reporting results only the best pair of kernel and 1SVM method is taken. The Wilcoxon test is conducted to perform a pair-wise comparison among the performance values of PSVM and SVDD when used with linear and RBF kernels. Table 7.2 demonstrates the output of this comparison on all the hybrid and stand-alone 1SVM results from Table 7.1. The returned $p$-values in either case fail to reject the null hypothesis for the accuracy measure with a level of significance of $\alpha = 0.1$, i.e., the accuracy of PSVM and SVDD are comparable for both kernels. Similar results are obtained for the training time of the 1SVM methods used with the RBF kernel, whereas in case of the linear kernel, PSVM outperforms SVDD, i.e., the sum of ranks for PSVM is less than the sum of ranks for SVDD, meaning that PSVMs performance was generally better. A discrepancy is observed in the testing time, where the linear kernel PSVM outperforms SVDD, while for the RBF kernel SVDD outperforms PSVM. Consequently, hereafter only the results for PSVM with the linear kernel and SVDD with the RBF kernel from Table 7.1 are considered in this study. Note that by 1SVM we refer to these two methods with the above combination/setup.

### 7.4.3 Influence of Dimensionality Reduction

While the main intention behind applying a dimensionality reduction method to an SVM is to alleviate the computational complexity, it is also interesting to investigate its effect on accuracy. Table 7.1 compares the differences in the performance of the hybrid and stand-alone methods on the experimental framework detailed above. As expected, the results obtained using the 1SVM approaches exhibit poor performance for high-dimensional
datasets. In contrast, the results from the hybrid 1SVMs suggest that a feature extraction
method such as PCA or DBN are able to enhance the accuracy of the baseline techniques.
Between the two, DBN achieved the best accuracy. On average the PCA-based 1SVMs
exhibit an increase of about 1% in their AUC, while SVDD and PSVM experience a 10%
and 16% increase in AUC, respectively, when combined with a DBN. In this way, the
DBN-1SVM can achieve a comparable AUC to AE2. This improvement is due to the fact
that DBNs are better in characterising highly non-linear functions with many variations
[16]. In contrast, the objective of PCA is to learn a linear manifold that is closest to the
distribution of training samples, i.e., characterising a lower-dimensional region in the in-
put space near where the training records have a higher density. However, for complex
real world domains, data manifolds are likely to be strongly non-linear.

In hybrid models, a similar improvement can also be observed for the testing time,
but not in the training time. Compared with the standalone methods, the results show
a substantial reduction in the testing time of the hybrid models for all the datasets, i.e.,
the average testing time of SVDD and PSVM are reduced by a factor of approximately 2
when aggregated with PCA, and by 150 and 5000 when aggregated with DBN. A trade-
off for the hybrid models is their training time, which shows an increase against SVDD
and PSVM, i.e., the training time of the hybrid SVDD methods are increased by half, and
for the hybrid PSVM methods are tripled.

In order to statistically compare the performance of the studied anomaly detection
methods and find the performance relationship among them, a multiple comparison test
is conducted on the reported results of Table 7.1. For ease of interpreting these results Fig-
ure 7.2 graphically demonstrates the average rankings of the three metrics, AUC, training
and testing time, obtained using the Friedman test [68].

Under all the measures, a statistical study conducting the Iman-Davenport test de-
tects significant differences between the algorithms and rejects the null hypothesis of
equivalence between the methods. Since the returned p-values — i.e., $0.30e^{-18}$ for AUC,
$0.92e^{-13}$ for training time, and $0.64e^{-15}$ for testing time — are significantly lower than
the significance level $\alpha$-value of 0.1. The identified differences are then analysed with a
Shaffer post-hoc test, shown in Tables 7.3 to 7.5. In these tables, “+” symbol indicates that
High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning

Note: The bars represent average rankings based on the Friedman test, and the number on the top of the bars indicates the ranking of the algorithm, from the best (1) to worst (7) for each given measure, and if a tie occurs then the best mean result is taken. The ranking is determined for all datasets and finally an average is calculated as the mean of all rankings.

Figure 7.2: Comparison of rankings of anomaly detection methods for 3 metrics.

the method in the row is statistically better than the one in the column, i.e., in the case of AUC it implies a higher accuracy and in the case of the time it implies a lower running time, whereas “−” implies the contrary; “=” indicates that the two compared methods have no significant differences.

Observing the results from Tables 7.3 and 7.5, it can be noted that the DBN-based methods (AE, DSVDD and DPSVM) clearly outperform their other counterparts in terms of AUC and testing time. Although DBN-based methods are not the leading approaches in terms of training time, as shown in Table 7.4, it can be concluded that DBN is a powerful tool as a dimensionality reduction method, when combined with 1SVMs. Moreover, it overcomes the underlying scalability issues of 1SVMs, and improves their performance.

In addition, it can be seen from the table that the performance of the DSVDD and DPSVM approaches are not significantly different from AE. To assess the significance of the results, the Wilcoxon test is applied to conduct a multiple pairwise comparison. Table 7.6 shows the p-values obtained from this test, which indicate that the AUC values of the three methods are statistically similar. However, both hybrid approaches (D2PSVM and D2SVDD), outperform AE2 in terms of training and testing time, at the significance level of $\alpha = 0.1$, with D2PSVM being in the lead.

From the reported results in Tables 7.1, 7.3 and 7.5, the positive synergy between the DBN and the 1SVM methods can be observed, offering an outstanding anomaly detection...
7.4 Evaluation and Discussion

Table 7.3: Shaffer test to compare AUC values.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSVM</th>
<th>SVDD</th>
<th>AE2</th>
<th>PCAPSVM</th>
<th>PCASVDD</th>
<th>D2PSVM</th>
<th>D2SVDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSVM</td>
<td>x</td>
<td>−(0.06)</td>
<td>−(0.84e⁻⁵)</td>
<td>= (0.56)</td>
<td>= (0.02)</td>
<td>−(0.11e⁻⁵)</td>
<td>−(0.11e⁻⁵)</td>
</tr>
<tr>
<td>SVDD</td>
<td>+ (0.06)</td>
<td>x</td>
<td>−(0.92e⁻²)</td>
<td>= (0.18)</td>
<td>= (0.64)</td>
<td>−(0.01)</td>
<td>−(0.01)</td>
</tr>
<tr>
<td>AE2</td>
<td>+ (0.84e⁻⁵)</td>
<td>+ (0.92e⁻²)</td>
<td>x</td>
<td>+ (0.13e⁻⁴)</td>
<td>+ (0.04)</td>
<td>= (1.0)</td>
<td>= (1.0)</td>
</tr>
<tr>
<td>PCAPSVM</td>
<td>= (0.56)</td>
<td>= (0.18)</td>
<td>−(0.13e⁻⁴)</td>
<td>x</td>
<td>−(0.07)</td>
<td>−(0.13e⁻⁴)</td>
<td>−(0.13e⁻⁴)</td>
</tr>
<tr>
<td>PCASVDD</td>
<td>+ (0.02)</td>
<td>= (0.64)</td>
<td>−(0.04)</td>
<td>+ (0.07)</td>
<td>x</td>
<td>−(0.04)</td>
<td>−(0.04)</td>
</tr>
<tr>
<td>D2PSVM</td>
<td>+ (0.11e⁻⁵)</td>
<td>+ (0.01)</td>
<td>= (1.0)</td>
<td>+ (0.13e⁻⁴)</td>
<td>+ (0.04)</td>
<td>x</td>
<td>= (1.0)</td>
</tr>
<tr>
<td>D2SVDD</td>
<td>+ (0.11e⁻⁵)</td>
<td>+ (0.01)</td>
<td>= (1.0)</td>
<td>+ (0.13e⁻⁴)</td>
<td>+ (0.04)</td>
<td>= (1.0)</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 7.4: Shaffer test to compare training times.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSVM</th>
<th>SVDD</th>
<th>AE2</th>
<th>PCAPSVM</th>
<th>PCASVDD</th>
<th>D2PSVM</th>
<th>D2SVDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSVM</td>
<td>x</td>
<td>= (0.64)</td>
<td>+ (0.19e⁻⁶)</td>
<td>+ (0.54e⁻²)</td>
<td>+ (0.26e⁻²)</td>
<td>+ (0.05)</td>
<td>+ (0.33e⁻³)</td>
</tr>
<tr>
<td>SVDD</td>
<td>= (0.64)</td>
<td>x</td>
<td>+ (0.20e⁻⁵)</td>
<td>+ (0.02)</td>
<td>+ (0.01)</td>
<td>= (0.13)</td>
<td>+ (0.18e⁻²)</td>
</tr>
<tr>
<td>AE2</td>
<td>− (0.19e⁻⁶)</td>
<td>− (0.20e⁻⁵)</td>
<td>x</td>
<td>− (0.02)</td>
<td>− (0.03)</td>
<td>− (0.12e⁻²)</td>
<td>= (0.11)</td>
</tr>
<tr>
<td>PCAPSVM</td>
<td>− (0.54e⁻²)</td>
<td>− (0.02)</td>
<td>+ (0.02)</td>
<td>x</td>
<td>= (0.82)</td>
<td>= (0.42)</td>
<td>= (0.42)</td>
</tr>
<tr>
<td>PCASVDD</td>
<td>− (0.26e⁻²)</td>
<td>− (0.01)</td>
<td>+ (0.03)</td>
<td>= (0.82)</td>
<td>x</td>
<td>= (0.29)</td>
<td>= (0.56)</td>
</tr>
<tr>
<td>D2PSVM</td>
<td>− (0.05)</td>
<td>= (0.13)</td>
<td>+ (0.12e⁻²)</td>
<td>= (0.42)</td>
<td>= (0.29)</td>
<td>x</td>
<td>= (0.11)</td>
</tr>
<tr>
<td>D2SVDD</td>
<td>− (0.33e⁻³)</td>
<td>− (0.18e⁻²)</td>
<td>= (0.11)</td>
<td>= (0.42)</td>
<td>= (0.56)</td>
<td>= (0.11)</td>
<td>x</td>
</tr>
</tbody>
</table>

Increasing the number of hidden layers (i.e., the depth of the representation) has shown promising results in tackling data complexity and the curse of dimensionality issues [16, 18]. However, that is not the case for linear algorithms such as PCA. Stacking up PCAs with the intention of forming a deeper architecture and obtaining more abstract representations is ineffective, since the composition of linear operations yields another linear operation [18]. Thus, only the impact of network depth on DBN-based methods is studied. Table 7.7 summarises the performance results of AE and DBN-1SVMs with one hidden layer, i.e., AE1 and D1x. Analogous to the earlier results, on average the hybrid methods present a higher AUC value and lower training/testing time.

Comparing with DBN2-based results in Table 7.1, the experiments indicate that increasing the number of hidden layers (deeper networks) enhances anomaly detection, in both the baseline AE and the hybrid methods. However, appropriate statistical analysis is
High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning

Table 7.5: Shaffer test to compare test times.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSVM</th>
<th>SVDD</th>
<th>AE2</th>
<th>PCAPSVM</th>
<th>PCASVDD</th>
<th>D2PSVM</th>
<th>D2SVDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSVM</td>
<td>x</td>
<td>− (0.06)</td>
<td>= (0.82)</td>
<td>= (0.25)</td>
<td>= (0.49)</td>
<td>− (0.52e^{-3})</td>
<td>− (0.01)</td>
</tr>
<tr>
<td>SVDD</td>
<td>+ (0.06)</td>
<td>x</td>
<td>= (0.26e^{-2})</td>
<td>= (0.25)</td>
<td>− (0.10e^{-6})</td>
<td>− (0.11e^{-4})</td>
<td></td>
</tr>
<tr>
<td>AE2</td>
<td>= (0.82)</td>
<td>+ (0.04)</td>
<td>x</td>
<td>= (0.35)</td>
<td>= (0.35)</td>
<td>− (0.12e^{-2})</td>
<td>− (0.02)</td>
</tr>
<tr>
<td>PCAPSVM</td>
<td>= (0.25)</td>
<td>+ (0.26e^{-2})</td>
<td>= (0.35)</td>
<td>x</td>
<td>− (0.06)</td>
<td>− (0.02)</td>
<td>= (0.16)</td>
</tr>
<tr>
<td>PCASVDD</td>
<td>= (0.49)</td>
<td>= (0.25)</td>
<td>= (0.35)</td>
<td>+ (0.06)</td>
<td>x</td>
<td>− (0.31e^{-4})</td>
<td>− (0.12e^{-2})</td>
</tr>
<tr>
<td>D2PSVM</td>
<td>+ (0.52e^{-3})</td>
<td>+ (0.10e^{-6})</td>
<td>+ (0.12e^{-2})</td>
<td>+ (0.02)</td>
<td>+ (0.31e^{-4})</td>
<td>x</td>
<td>= (0.35)</td>
</tr>
<tr>
<td>D2SVDD</td>
<td>+ (0.01)</td>
<td>+ (0.11e^{-4})</td>
<td>+ (0.02)</td>
<td>= (0.16)</td>
<td>+ (0.12e^{-2})</td>
<td>= (0.35)</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 7.6: Wilcoxon test to compare the performance of DBN-based methods regarding p-values of Wilcoxon test.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Train Time</th>
<th>Test Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R⁺</td>
<td>R⁻</td>
<td>p-value</td>
</tr>
<tr>
<td>D2PSVM vs. AE2</td>
<td>8.5</td>
<td>1.5</td>
<td>0.375</td>
</tr>
<tr>
<td>D2SVDD vs. AE2</td>
<td>8.5</td>
<td>1.5</td>
<td>0.375</td>
</tr>
<tr>
<td>D2PSVM vs. D2SVDD</td>
<td>1.5</td>
<td>1.5</td>
<td>1</td>
</tr>
</tbody>
</table>

Note: R⁺ corresponds to the sum of the ranks for the method on the left and R⁻ for the right.

required to support this claim. For this purpose, the Wilcoxon signed-rank test is applied to detect differences among each pair of deep and shallow networks. Table 7.8 shows the returned p-values. In concordance with previous studies [21], the experimental results suggest that the deeper the network, the greater the accuracy and training/testing time, except for D2PSVM. The Wilcoxon test shows that the AUC of all DBN1-based networks are outperformed by the DBN2-based networks with a level of significance of α = 0.1. This statement also holds for the training/testing measures of AE and DSVDD, while the p-values for the training/testing times of the DPSVMs are not significantly different at the level of α = 0.1.

Based on these results, it can be concluded that deeper networks are able to infer more abstract features. Accordingly, more abstract features are generally more invariant to underlying variations in the training set. Combining DBN2 and 1SVMs, therefore results in a more robust anomaly detection technique, in which accuracy is not adversely affected by small deviations from the trained model. Recall that PSVM obtains comparable AUC as SVDD, when combined with DBN2. Hence, the greater computational efficiency of D2PSVM makes it a more practical anomaly detection technique.
### Table 7.7: Performance results of DBN-based methods with one hidden layer.

<table>
<thead>
<tr>
<th>Method</th>
<th>Metric</th>
<th>Dataset</th>
<th>AE1</th>
<th>D1PSVM</th>
<th>D1SVDD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>Forest</td>
<td>0.97</td>
<td>0.98</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Banana</td>
<td>0.93</td>
<td>0.96</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adult</td>
<td>0.98</td>
<td>0.94</td>
<td>0.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GAS</td>
<td>0.96</td>
<td>0.352</td>
<td>0.0823</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OAR</td>
<td>0.94</td>
<td>0.433</td>
<td>0.1502</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSA</td>
<td>0.97</td>
<td>0.383</td>
<td>0.2542</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Smiley</td>
<td>0.93</td>
<td>0.389</td>
<td>0.3811</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HAR</td>
<td>0.98</td>
<td>0.0039</td>
<td>0.9675</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avg.</td>
<td>0.9575</td>
<td>0.0047</td>
<td>0.0036</td>
</tr>
<tr>
<td>AE1</td>
<td>Train</td>
<td>0.104</td>
<td>0.267</td>
<td>0.003</td>
<td>0.0087</td>
</tr>
<tr>
<td></td>
<td>Test</td>
<td>0.002</td>
<td>0.352</td>
<td>0.0031</td>
<td>0.0093</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1PSVM</td>
<td>0.0304</td>
<td>0.096</td>
<td>0.0087</td>
</tr>
<tr>
<td></td>
<td>Test*</td>
<td>0.01</td>
<td>0.003</td>
<td>0.0093</td>
<td>0.0087</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1SVDD</td>
<td>0.0827</td>
<td>0.0992</td>
<td>0.0093</td>
</tr>
<tr>
<td></td>
<td>Test*</td>
<td>0.1011</td>
<td>0.14</td>
<td>0.0149</td>
<td>0.1066</td>
</tr>
</tbody>
</table>

Note: Training and testing times are in seconds, except for testing time rows marked with a *, which are in milliseconds.

### Table 7.8: Wilcoxon tests to determine the influence of the number of hidden layers on the performance of DBN-based methods.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy</th>
<th>Train Time</th>
<th>Test Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R+</td>
<td>R-</td>
<td>p-value</td>
</tr>
<tr>
<td>AE2 vs. AE1</td>
<td>36</td>
<td>0</td>
<td>0.0078</td>
</tr>
<tr>
<td>D2SVDD vs. D1SVDD</td>
<td>15</td>
<td>0</td>
<td>0.0625</td>
</tr>
<tr>
<td>D2PSVM vs. D1PSVM</td>
<td>28</td>
<td>0</td>
<td>0.0156</td>
</tr>
</tbody>
</table>

Note: R+ corresponds to the sum of the ranks for the method on the left and R- for the right.

The reason behind this phenomena can be explored in Figure[7.3] which demonstrates the benefits of using DBN features for anomaly detection on two datasets, Banana and HAR. The interpretation of the subfigures, from the left, is as follows: the iVAT image of the raw datasets, the iVAT image of the reduced datasets with one hidden layer, and with two hidden layers. As can be seen in the image of the raw datasets, normal records (first 760 records) appear in dense blocks, while the anomalous records (last 40 records) are shown in gray shadow (since they are distributed across the dataset). When the data are projected to a lower dimensional space with DBN, a clearer separation appears between the normal records and anomalies. The impact of the depth of the network is reflected in the improved clarity and contrast in the block structure. Projecting data with deeper networks (e.g., DBN2) provides sufficient separability among the normal and anomalous records that to enable the use of basic kernels such as linear, rather than more complex ones such as RBF.
7.4.5 Influence of Number of Training Records

Unlike deep networks, SVMs are known to be less effective for large-scale training on high-dimensional data \[90\], i.e., their performance decreases as the number of records in the dataset increases. To study this concern and also examine the performance of our hybrid methods, PSVM, SVDD, AE2, D2PSVM and D2SVDD are trained on the four datasets with the highest dimensionality (HAR, Smiley, DSA and OAR) while varying the number of records (from 200 to 5000). As shown in Figure 7.4 with 1000 or more training records, DBN-based methods outperform PSVM and SVDD. When the number of records is particularly small (less than 500), SVDD can perform slightly better than DBN-based methods. More specifically, the accuracy of the DBN-based methods generally increases as the number of records grows, whereas the 1SVM methods show the reverse behaviour. This suggests that the 1SVMs tend to learn models with higher variance for large datasets, i.e., they overfit the data.
Figure 7.4: Comparing the accuracy of anomaly detection methods as the number of training records is varied.

Table 7.9: Comparing time- and memory-complexities.

<table>
<thead>
<tr>
<th>Technique</th>
<th>PSVM</th>
<th>SVDD</th>
<th>DBN</th>
<th>DSVDD</th>
<th>DPSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>$nm$</td>
<td>$O(nm^2)$</td>
<td>$O(nmd)$</td>
<td>$O(nmd + dm^2)$</td>
<td>$O(nmd + dm)$</td>
</tr>
<tr>
<td>Testing</td>
<td>$n + m$</td>
<td>$O(nm)$</td>
<td>$O(nmd)$</td>
<td>$O(nmd + dm^2)$</td>
<td>$O(nmd + d + m)$</td>
</tr>
<tr>
<td>Memory</td>
<td>$nm$</td>
<td>$O(nm^2)$</td>
<td>$O(nq)$</td>
<td>$O(nq + dm^2)$</td>
<td>$O(nq + dm)$</td>
</tr>
</tbody>
</table>

7.4.6 Complexity

When it comes to large-scale training, in addition to robustness, it is advantageous for an anomaly detection algorithm to be efficient in terms of time- and memory-complexity. Table 8.2 compares the complexity of PSVM, SVDD, DBN, DPSVM and DSVDD. As can be seen from the table, SVDD and DSVDD with the RBF kernel \[170\] have much higher (quadratic) training- and memory-complexity compared to the other techniques. Though the complexity of DPSVM may not be as efficient as PSVM, still it grows linearly.

In summary, from the aforementioned analysis we can conclude that the proposed hybrid methods are more suitable and robust than the standalone techniques for large-scale and high-dimensional anomaly detection. Recall that the efficiency and accuracy
High-Dimensional and Large-Scale Anomaly Detection using a Linear One-Class SVM with Deep Learning

Figure 7.5: Comparing the performance results of the proposed and studied anomaly detection techniques.

of D2PSVM make it the most effective approach, even when compared with the state-of-the-art AE. Figure 7.5 demonstrates that the proposed hybrid methods has the best performance, the highest AUC and lowest training/testing time, among all the studied methods, with D2PSVM taking over D2SVDD in computational time.

7.5 Conclusion

In this chapter, we proposed an unsupervised anomaly detection technique for high-dimensional complex datasets. The technique is a combination of a DBN and one-class SVM. The DBN is trained as a dimensionality reduction algorithm, generating a non-linear manifold and transforming the data into a lower dimensional set of features. The
derived features from the training samples are taken as input to train the one-class SVM. Subsequently, the generated hybrid model from the two algorithms is used for testing.

Coupling a DBN with 1SVM is advantageous since it tackles the complexity and scalability issues of the 1SVM, especially when training with large-scale datasets. DBNs are appropriate feature detectors for anomaly detection, taking only raw (unlabelled) data to capture higher-order correlation among features, generating accurate models, and imposing minimal computational and memory complexity. Reducing the number of irrelevant and redundant features boosts the scalability of a 1SVM for use with large training datasets containing high-dimensional records. In addition, by using a deep architecture, a DBN-1SVM can achieve better generalisation. Therefore, in the hybrid model more basic kernels, such as the linear kernel, can be substituted for more complicated and computationally expensive ones, such as the RBF kernel, with no loss of accuracy. Hence, our model offers an efficient, accurate and scalable anomaly detection approach that can be applied for anomaly detection in large-scale and high-dimensional domains such as PSNs.

The experimental results on a wide range of benchmark datasets demonstrate that the AUC of DBN-1SVM outperforms standalone 1SVMs, by a margin of around 20% in some datasets. Comparing with an autoencoder, the difference is negligible, while the hybrid DBN-1SVM executes 3 times faster in training and 1000 faster in testing. In the next chapter we take an alternative approach to address the problem of unsupervised anomaly detection on high-dimensional data with 1SVMs. Specifically, we avoid a non-linear kernel by applying a linear kernel on randomly projected data.
Chapter 8

R1SVM: a Randomised Nonlinear Approach to Large-Scale Anomaly Detection

One-class support vector machines have demonstrated their effectiveness as an anomaly detection technique. However, their ability to model large datasets is limited due to their memory and time complexity for training. In the previous chapter we proposed DBN-1SVM, a hybrid technique — a combination of deep learning and linear SVM — for anomaly detection in high-dimensional datasets. Alternatively, to address this issue for supervised learning of kernel machines, there has been growing interest in random projection methods as an alternative to the computationally expensive problems of kernel matrix construction and support vector optimisation. In this chapter we leverage the theory of nonlinear random projections and propose the Randomised One-class SVM (R1SVM), which is an efficient and scalable anomaly detection technique that can be trained on large-scale datasets. Our empirical analysis on several real-life and synthetic datasets shows that our randomised 1SVM algorithm achieves comparable or better accuracy to deep autoencoder and traditional kernelised approaches for anomaly detection, while being approximately 100 times faster in training and testing.

The publication arising from the work in this chapter is paper P3.

8.1 Introduction

Unsupervised anomaly detection (also known as outlier detection) plays a significant role in a variety of applications, such as fraud detection, network intrusion detection and fault diagnosis. 1SVMs [149][165] have proven to be a very effective unsupervised learning method to construct highly accurate classifiers for anomaly detection. However, 1SVMs are often impractical for use on very large datasets due to the computational and
memory complexity of their underlying optimisation problem during training \[21]\[169]\[172]\]. Recently, there has been growing interest in randomised approaches to improve the efficiency of kernel methods for supervised learning of SVMs \[137]\[138]\]. In this chapter, we build on the theory of nonlinear random projections in order to accelerate the training of 1SVMs, and propose a new form of anomaly detector called a Randomised One-class SVM (R1SVM). We show that our R1SVM can achieve comparable or better accuracy than an existing 1SVM method, while reducing training and testing time by up to two orders of magnitude.

A key challenge in anomaly detection is how to characterise the distribution of “normal” (i.e., non-anomalous) data. This is particularly challenging when the process that generated the normal data has an unknown, potentially complex underlying distribution. 1SVM approaches have had considerable success in addressing this modelling task by using a kernel function to implicitly map the data from the input space to a higher dimensional feature space, in which a relatively simple model such as a hyperplane \[149]\, hypersphere \[165]\ or hyperellipsoid \[139]\[173]\ can be used to characterise normal observations.

A practical limitation of 1SVM approaches is their computational and memory complexity for training. In a dataset with \(n\) records, each with \(d\) dimensions, training using a nonlinear kernel requires \(O(dn^2)\) computational complexity, as well as \(O(dn^2)\) memory complexity for the kernel matrix \[165]\. This limits the utility of 1SVM in applications involving large datasets. While training can be performed on a smaller sample of the training data, this can reduce the accuracy of the 1SVM due to the sparse sampling of the underlying distribution, particularly in applications that involve high dimensional input spaces.

Recently, there has been significant progress in using randomised features in conjunction with linear algorithms to reveal nonlinear patterns in data. In particular, a nonlinear, randomised variant of component analysis methods such as Randomised Principal Component Analysis (RPCA) and Randomised Canonical Correlation Analysis (RCCA) has been proposed \[118]\. These randomised variants have been applied to the tasks of regression and classification of large datasets, and exhibited significant savings in com-
putation time while incurring little or no loss in accuracy.

In this chapter, we propose a novel application of randomised methods by deriving a highly scalable algorithm for anomaly detection based on training a linear one-class SVM using randomised, nonlinear features. By using randomised features rather than finding a set of optimised support vectors, we can substantially reduce the cost of training our one-class SVM. We provide extensive empirical testing to show that our randomised 1SVM method achieves substantial improvement in both computational complexity and accuracy over exact kernel methods.

To the best of our knowledge this is the first attempt to exploit nonlinear random features in kernel-based methods for anomaly detection. In addition to significantly reducing computational complexity, we show that our randomised 1SVM algorithm achieves comparable or better accuracy compared to autoencoder and kernelised approaches to anomaly detection. We postulate that this improvement in accuracy is due to the implicit regularisation induced by randomness, as well as an improvement in the separation between normal and anomalous data points when compared in the nonlinear feature space. By improving the efficiency of training 1SVMSs in this way, we believe it will be possible to apply anomaly detection to data-intensive applications in resource constrained environments, such as wireless sensor networks.

The remainder of the chapter is organised as follows. Section 8.2 discusses related work. Section 8.3 presents our proposed unsupervised anomaly detection approach R1SVM. Section 8.4 describes the empirical analysis on the performance of R1SVM on various real-world and synthetic datasets. It demonstrates the advantages of random projections on the separability of data which affects anomaly detection. It also compares the accuracy and efficiency of R1SVM with SVDD and AE. Section 8.5 summarises the chapter.

8.2 Related Work and Background

While numerous 1SVM formulations using nonlinear kernel have been proposed in the literature [27, 149, 165], a common feature of many formulations is the solution of a
Quadratic Programming (QP) problem. In particular, these kernel-based methods rely on the calculation of a kernel matrix over all pairs of data points, which limits the scalability of training 1SVMs on large datasets. This can also limit the effectiveness of 1SVMs on high dimensional input spaces, given the need to have a sufficiently large training dataset that spans the variation in the high dimensional space.

Existing approaches to address the scalability problems of SVMs can be classified into two general categories. One category comprises hybrid and complimentary methods for SVMs, which are used to preprocess the data prior to processing by the SVM. For example, clustering [161], dimension reduction techniques such as PCA or KPCA [36, 160], and deep belief networks [21] are some of the most well-known approaches. Although these approaches play an important role in building the model, they do not directly address the scalability of the SVM itself. The second category includes methods that aim to alleviate the QP problem of kernel machines. A more heuristic approach is to reduce the size of the QP problem by breaking it into smaller pieces, for example by using chunking [158, 169], decomposition [95, 172], or Sequential Minimal Optimisation (SMO) [135]. An alternative to enhance the computational efficiency of SVMs is to instead use an approximation of QP [70]. A more radical approach is to totally avoid the QP problem, and obtain the solution through a fast iterative scheme [71, 185].

To reduce the memory and computational complexity, a popular approach is to obtain a low-rank approximation of the kernel matrix. Selective sampling (or active learning) methods iterate through the training data and sample a small subset of the records that are near the boundary in the feature space with higher probabilities, e.g., [167], or see [151] for a survey. To avoid the computational cost of processing the whole dataset, Lee and Mangasarian [107] propose the use of random sampling to obtain a result that is close to the original SVM.

A more recent trend explores the use of randomisation, such as linear random projection [25] as a substitute for the computationally expensive cost of kernel matrix construction. An early example is the work of Achlioptas et al. [2], which replaces the kernel function by a randomised kernel to speedup KPCA. The work of Rahimi and Recht [137, 138] made a breakthrough in this approach. They replicated an RBF ker-
8.3 Proposed Approach — R1SVM

In this section we present our Randomised 1SVM (R1SVM) model for anomaly detection. We begin by recalling a few key aspects of one-class SVMs, and then introduce the use of nonlinear random projections for detecting anomalies in large-scale data. Random projections have been utilised mainly in distance-based classification or data reconstruction schemes to speedup the search, as it approximately preserves $L_2$ distances among a set of points. Thus instead of performing the search in a high-dimensional space, the search is conducted in a space of reduced dimension but on a larger neighbourhood. Note that in our context, ultimately, our goal is anomaly detection. Therefore, we are not necessarily interested in deriving a representation that allows for the best classification or reconstruction of the data, but we rather seek to find a model of the underlying distribution of the data which can then be used to detect anomalies.

8.3.1 One-class SVM (1SVM)

Let $X = [x_1, \cdots, x_n]$ be the $d \times n$ matrix containing $n$ training points $x_i \in \mathbb{R}^d$ of one specific class, and let $\phi$ be a feature map $X \rightarrow \mathcal{H}$ such that the dot product in $\mathcal{H}$ can be computed using some kernel

$$k(x, x') = <\phi(x), \phi(x')>_{\mathcal{H}}.$$ 

A One-class Support Vector Machine (1SVM) \cite{122, 149} finds anomalies by first pro-
jecting the data to the feature space $\mathcal{H}$, and then finding a hyperplane that best separates the data from the origin. In other words, the decision function in the 1SVM returns $+1$ in a region where most of the data points occur (i.e., where the probability density is high), and returns $-1$ elsewhere.

Defining a family of sets $C_{s,\rho} = \{ x | f_{s,\rho}(x) > 0 \}$, the 1SVM estimates a function

$$f_{s,\rho}(x) = \text{sgn}(s \cdot \phi(x) - \rho)$$

that maximises the distance of all the data points (in the feature space $\mathcal{F}$) from the hyperplane to the origin, parameterised by a weight vector $s$ and an offset $\rho$.

Thus, the resulting binary function $f_{s,\rho}(x)$ can be estimated by minimising the regularised risk:

$$R_{\text{reg}}[f_{s,\rho}(x)] = R_{\text{emp}}[f_{s,\rho}(x)] + \frac{1}{2} \| f_{s,\rho}(x) \|^2_{\mathcal{H}}$$

(8.1)

where $R_{\text{emp}}(\cdot)$ is the empirical risk and $\frac{1}{2} \| f_{s,\rho}(x) \|^2_{\mathcal{H}}$ is the regulariser. The empirical risk is the average loss and can be written as

$$R_{\text{emp}}[f_{s,\rho}(x)] \equiv \frac{1}{n} \sum_{i=1}^{n} c(f_{s,\rho}(x_i), y_i)$$

(8.2)

where $c(f_{s,\rho}(x_i), y_i)$ is the loss function that penalises the deviation between the prediction $f_{s,\rho}(\cdot)$ and the label $y$, i.e., this captures the cost of the errors caused when $f_{s,\rho}(\cdot)$ is negative on training vectors.

Replacing $\| f_{s,\rho}(\cdot) \|^2_{\mathcal{H}}$ with a maximum margin regulariser $\| s \|^2$ to penalise complex regions, we can setup the following quadratic program for 1SVM:

$$\min_{s, \xi, \rho} \frac{1}{2} \| s \|^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i - \rho$$

$$s.t. \quad (s \cdot \phi(x_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0$$

(8.3)

where $\nu \in (0, 1]$ is a regularisation parameter that controls the fraction of anomalies and the fraction of support vectors, and $\xi_i$ are the slack variables that allow some of
the data vectors to lie on the wrong side of the hyperplane. By introducing the Lagrange multipliers, we arrive at the following quadratic program, which is the dual of the primal program in Eq. (8.4):

$$\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) \\
\text{s.t.} & \quad 0 \leq \alpha_i \leq \frac{1}{\nu n}, \\
& \quad \sum_i \alpha_i = 1,
\end{align*}$$

(8.4)

where \(\alpha_i\) are the Lagrange multipliers. Further, \(s = \sum \alpha_i \phi(x_i)\). Using the Karush-Kuhn-Tucker (KKT) optimality conditions the data vectors can be characterised in terms of whether they fall below, above, or on the hyperplane boundary in the feature space depending on the corresponding \(\alpha_i\) values. Data vectors with positive \(\alpha_i\) values are the support vectors. Further, for \(0 < \alpha_i < 1/\nu n\), the data vectors fall on the hyperplane and hence \(\rho\) can be recovered using these vectors, vis-a-vis \(\rho = \langle s, \phi(x_i) \rangle = \sum j \alpha_j k(x_j, x_i)\).

Therefore, the decision function can now be written as

$$f_{s,\rho}(x) = \text{sgn}(s, \phi(x) - \rho) = \text{sgn}(\sum \alpha_i k(x_i, x) - \rho)$$

(8.5)

The solution to the quadratic program in Eq. (8.5) is characterised by the parameter \(\nu \in (0, 1]\), which sets an upper bound on the fraction of anomalies (training examples regarded as out-of-class) and a lower bound on the number of training examples used as support vectors.

The computational complexity of 1SVM using an SMO solver is approximately \(O(dn^2)\) for the RBF kernel [170], and \(O(dn)\) for a linear kernel with \(n\) being the number of samples and \(d\) the number of dimensions in feature space. However, it has been noted that when 1SVM is used with a linear kernel, it introduces a bias to the origin. This problem can be removed by using an RBF kernel, which has a higher computational complexity associated with the higher dimensional kernels, thus making it cumbersome for processing with large scale data.

In order to overcome this limitation, in the next section, we propose to exploit nonlin-
ear random projections inside a linear 1SVM, which serves as a good approximation of a nonlinear 1SVM.

8.3.2 Randomised 1SVM

We propose R1SVM, a nonlinear randomised variant of 1SVM, which applies the original linear 1SVM method on a randomised nonlinear projection of the data. We first discuss how to generate the nonlinear random features from the original data, and then we show how to employ these features to detect anomalies using a linear 1SVM. This approach eliminates the need to deal with large kernel matrices for large datasets, consequently reducing the computational complexity while achieving comparable or better anomaly detection performance than a traditional 1SVM (as shown in the Evaluation section).

Generating Nonlinear Random Features

Consider the problem of fitting a function $f$ (note that the subscripts $s$ and $\rho$ in $f_{s,\rho}$ are omitted for brevity) to the data set $\{x_i, y_i\}$, where $y_i$ values are always set to 1 for the one-class problem. This fitting problem consists of finding $f$ that minimises the empirical risk in equation Eq. (8.2). For the 1SVM problem, the loss function $c(y, y')$ is of the form $c(y, y') = \max(0, 1 - yy')$. Using the kernel function, the function $f(x) = \text{sgn}(s.\phi(x) - \rho)$ becomes $f(x) = \sum a_i k(x, x_i)$. Therefore, in the general form, the function $f(x)$ can be written as $f(x) = \sum_{i=1}^{n} a_i \phi(x; s_i)$, where $\phi$ are parameterised by vector $s$ and weighted by $a_i$. By jointly optimising over $s$ and $a_i$, in a greedy manner, the solution can be found [137]. However, this is computationally intensive. Rahimi and Recht [138] have proved that this nonlinear optimisation problem over $(a, s_1, \ldots, s_n)$ in $f$, can be solved by randomly sampling the $s_i \in \mathbb{R}^d$ from a data-independent distribution $p(s)$ and creating $k$-dimensional random features $z(X) = [z_1 \cdots z_k]$, where $z_i = [\cos(s_i^T x_1 + b_i), \cdots, \cos(s_i^T x_n + b_i)]$ are Fourier based random features. For more details refer to [137]. Thus, we arrive at the following simplified optimisation problem:

$$\min_{a \in \mathbb{R}^k} \frac{1}{n} \sum_i c(a^T z_i, y_i)$$
8.4 Evaluation and Discussion

where $B$ is a regularisation constant. Furthermore, it is shown by Rahimi and Recht \cite{138} that using randomly selected features in nonlinear spaces causes only bounded error compared to using optimised features:

**Theorem 8.1.** Let $\mathcal{F} = \{ f(x) = \int_\Omega \alpha(s)\phi(x;s)ds \mid |\alpha(s)| \leq B p(s) \}$, where $p$ is a distribution on $\Omega$ and $|\phi(x;s)| \leq 1$. Draw $s_1, \ldots, s_k$ iid from $p$. Further let $\lambda > 0$, and $c$ be some L-Lipschitz loss function, then the function $f_k(x) = \sum_{i=1}^k \alpha_i \phi(x;s_i)$ minimises the empirical risk $c(f_k(x), y)$ has a distance from the $c$-optimal estimator in $F$ bounded by

$$E_p[c(f_k(x), y)] - \min_{f \in \mathcal{F}} E_p[c(f(x), y)] \leq O \left( \frac{LB}{\sqrt{n}} + \frac{1}{\sqrt{k}}LB\sqrt{\log \frac{1}{\delta}} \right)$$

(8.7)

with a probability of at least $1 - 2\delta$.

The convergence rate of our randomised R1SVM to its original kernel 1SVM version can be expressed by the following theorem \cite{118}:

**Theorem 8.2.** Given the data $X \in \mathbb{R}^{n \times d}$, a shift invariant kernel $k$, a kernel matrix $K_{ij} = k(x_i, x_j)$ and its approximation $\hat{K}$ using $k$ random features, it can be proven that

$$E\|\hat{K} - K\| \leq \sqrt{\frac{3n^2 \log n}{k}} + \frac{2n \log n}{k}. \tag{8.8}$$

The proof to this theorem can be found in \cite{118}.

8.4 Evaluation and Discussion

In this section, we evaluate the effectiveness of our R1SVM method for anomaly detection by conducting the following two experiments. First, we empirically explore the impact of random projections on the separability of normal data records from anomalous records. Then we compare the performance of R1SVM in terms of accuracy, training and
testing time, with SVDD and a deep autoencoder (AE), following the experimental setup described in Section 7.4 and any changes in the setup are as follows.

**Experimental setup:** For visualisation purposes, in the first experiment, we used \textit{iVAT} \cite{174}, which helps visualise the possible number of clusters in, or the cluster tendency of, a set of objects. In the second experiment we used the \textit{svdd} implementation from Dd-tools \cite{166} as the one-class SVM method. Note that the hypersphere-based SVDD model using an RBF kernel is equivalent to a hyperplane-based 1SVM model. In the case of the autoencoder, we implemented a basic autoencoder including five-layers with tied weights and a sigmoid activation function for both the encoder and decoder. The training is conducted in mini-batches of \( q = 100 \) records. Initially the autoencoder was trained based on greedy layer-wise pre-training (i.e., training one layer at a time) to extract features, and then using these features to train the next layer.

**Datasets:** The experiments are conducted on six real-life datasets from the UCI Machine Learning Repository: (i) Forest, (ii) Adult, (iii) Gas, (iv) OAR, (v) DSA, and (vi) HAR, with dimensionalities of 54, 123, 128, 242, 315 and 561 features, respectively. We also use two synthetic datasets, the “Smiley” dataset containing 20 dimensions and the “Banana” containing 100 dimensions. All the records in each dataset are normalised between [0,1].

Although R1SVM is designed to overcome the challenges that arise for anomaly detection in large datasets, we conducted our experiments on datasets with varying numbers of dimensions and records, from 200 to 40,000 records, to assess the effect of data size on its performance. In each experiment records are divided based on the instruction in Section 5.4 for training and testing.

### 8.4.1 Experiment 1: Interpretation of Basis for Anomaly Detection with R1SVM

Figure 8.1 demonstrates the benefits of using random nonlinear features for anomaly detection on two datasets, Banana and HAR. The interpretation of the subfigures, from the top is as follows: \textit{iVAT} image of the raw datasets, the \textit{iVAT} image of the projected datasets, and the scatter plot of the projected datasets (projected to \( \mathbb{R}^3 \)). As can be seen in the image of the raw datasets, normal records (first 1000 records) appear in dense
Figure 8.1: Demonstration of the effect of nonlinear projection on normal and anomalous records.
blocks, while the anomalous records (last 50 records) are shown in gray shadow (since they are distributed across the dataset). When the data are projected to a lower space, a clearer separation appears between the normal records and anomalies, as reflected in the improved clarity and contrast in the block structure. This is also reflected in the corresponding scatter plots, which show the effect of projection on normal records (shown in blue) and anomalies (shown in black). We postulate that the explanation of this effect is the concentration of the data around its mean as a result of the random projection [49,190].

8.4.2 Experiment 2: Empirical Performance of R1SVM

Table 8.1 compares the AUC results, training and testing time of R1SVM with SVDD and AE for several medium size (2000 records) datasets — a limit of 2000 records was chosen because the performance of SVDD significantly degrades on larger datasets. As shown in the table, our proposed approach delivers a comparable AUC to the state-of-the-art AE. However, the AUC of SVDD is significantly lower — e.g., up to 14% for DSA and 13% for Smiley datasets. A more significant advantage of R1SVM is its reduction in training/testing time. R1SVM reduces these measures by a factor of approximately 100 and 1000 times compared to AE and SVDD, respectively.

When selecting an anomaly detection technique, the size of the training dataset is an immediate concern. Some techniques, such as kernel machines, can perform best with small datasets, i.e., their performance decays as the number of records grows. In con-
Figure 8.2: Comparison of accuracy of anomaly detection methods as the number of training records is varied.
Methods like AE can be inaccurate if trained with small numbers of records [21]. Additionally, the training time for some techniques is prohibitive for large numbers of records, e.g., the time complexity of kernel-based methods can grow at least quadratically in the number of data records. Figure 8.2 and Figure 8.3 show how the AUC and training/testing time vary with the number of training records.

Figure 8.2 shows the impact of larger-scale training and high-dimensional data on the accuracy of the studied approaches. We observe that the accuracy of SVDD decreases as the number of training records increases. In some cases, e.g., Smiley, OAR and DSA datasets, SVDD experiences a substantial decrease in AUC when the number of training records reaches 5000. In contrast, the accuracy of AE can initially be low but increases as the data size grows. Overall, only R1SVM delivers more consistent results across various ranges of data sizes.

An attractive property of autoencoders is their efficiency in training/testing time, which scales linearly with the number of records. Therefore, we compared the training and testing times of AE and R1SVM for large datasets. The results in Figure 8.3 suggest that although both measures grow linearly for the two techniques, the training/testing time of AE grow at a much faster rate. The AUC values for this experiment are not included, since they were more or less consistent. Table 8.2 summarises the computational and memory complexity of these techniques, where $m$ and $q$ are the size of the bottleneck layer and batch in the AE.
Table 8.2: Comparing computational and memory complexity of SVDD, AE and R1SVM

<table>
<thead>
<tr>
<th>Technique</th>
<th>SVDD</th>
<th>AE</th>
<th>R1SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>(O(dn^2))</td>
<td>(O(dmn))</td>
<td>(O(kn))</td>
</tr>
<tr>
<td>Testing</td>
<td>(O(dn))</td>
<td>(O(dmn))</td>
<td>(O(k + d))</td>
</tr>
<tr>
<td>Memory</td>
<td>(O(dn^2))</td>
<td>(O(dq))</td>
<td>(O(kn))</td>
</tr>
</tbody>
</table>

In summary, the above experiments demonstrate that R1SVM is as accurate as a deep autoencoder, but approximately 100 times faster in terms of training/testing time. It is also important to note that the training algorithm of R1SVM requires only one parameter, \(k\), to be set. Since we are interested in anomaly detection, modelling the underlying distribution is more important than preserving distances among classes, therefore the data can be projected to very low dimensions. In our empirical analysis we have also tested the sensitivity of R1SVM to the choice of the parameter \(k\). Overall, the value of \(k\) has little effect on accuracy, while larger values of the \(k\) increase the training time. Hence, smaller values of \(k\) are most effective.

8.5 Conclusion and Research Directions

We presented R1SVM, an unsupervised anomaly detection technique that approximates a nonlinear 1SVM by applying the original linear 1SVM method on a randomised nonlinear projection of the data. Using a simple but effective random projection overcomes the scalability issues of 1SVM methods while enhancing the accuracy of anomaly detection. Our empirical analysis on several benchmark datasets shows that R1SVM not only delivers significant improvements over a conventional nonlinear 1SVM, but it matches the performance of a state-of-the-art deep autoencoder — while reducing its training and testing time by up to two orders of magnitude. These savings in time and space enable R1SVM to execute anomaly detection on large datasets more efficiently, in real-time applications or memory constrained devices, such as smart phones and wireless sensor networks. In addition to large-scale training, the other major advantage of R1SVM is that
it is also shown to be effective at maintaining its accuracy on small datasets when training data is limited.

In future work, we will also explore the changes in the eigen-spectrum of the kernel matrix and generalisation error bound when using the Nyström method. Unlike random Fourier features, the basis functions in the Nyström method are data dependent and randomly sampled from the training data.
Chapter 9
Conclusion and Future Work

9.1 Summary of Contributions

The increasing number of sensor-rich devices connected to the Internet and the ease of data collection have motivated a novel paradigm known as Participatory Sensing Networks (PSNs). The vision of PSNs is to network smart devices to realise innovative, data-intensive applications through collaborative data gathering, processing and sharing. Using their smart phones, ordinary citizens can monitor their daily activity or surrounding environment in fine detail. Given the ubiquity of mobile phones and communication infrastructure (e.g., cellular and WiFi) in metropolitan areas, participatory applications can provide an unprecedented level of coverage, capturing unpredictable events and revealing new emerging patterns. Anomaly detection aims to detect unusual behaviour or events of interest in the monitored environment, and thus is of great importance in participatory applications. Consequently, the focus of this thesis has been on new collaborative anomaly detection methods, tailored to cope with the requirements of participatory networks.

In Chapter 2, we explored the common types of challenges arising in PSNs, and surveyed the relevant approaches in the literature. Based on this survey, we have focused on two open research problems associated with anomaly detection in PSNs, namely, (i) how to protect the privacy of participants’ data, and (ii) how to maintain the accuracy of anomaly detection models generated from large-scale high-dimensional data that is contributed in such networks.

In collaborative learning, two or more participants contribute their local datasets to a
server for analysis. The common trend in the PPDM literature is to rely on trusted parties to maintain the privacy of participants. However, our privacy-preserving schemes are designed to be independent of these entities to better fit real-world scenarios. While traditional approaches can achieve good performance and efficiency, trusted parties can be susceptible to various types of attacks. Instead, in our work, participants themselves maintain their privacy, perturbing their data prior to sharing them with the data mining server. Given that perturbation is applied on the participants’ side, our privacy schemes are designed so that they are computationally efficient and scale to large numbers of participants.

We mainly focused on two types of queries: aggregation queries and anomaly detection queries. In aggregation queries, the challenge is that the server should be able to calculate an accurate sum of the contributed records, but cannot reveal the participants’ true values, whereas in anomaly detection queries, the server generates a global model from all participants’ data.

Our privacy-preserving data aggregation scheme is based on data slicing and mutual protection through data sharing. Sampled measurements are split into random “slices” and transmitted to the server via the neighbouring participants in the network. To provide an additional layer of privacy, participants mask the data slices using homomorphic encryption to hide their values. While enhancing data privacy, the homomorphic encryption enables the neighbouring participants and the server to aggregate all the received data slices, without decrypting them. Moreover, in this scheme participants provide the server with a HMAC of their record to facilitate integrity checking of the aggregated data. We demonstrated that if a data record is distributed among sufficient number of neighbours, the participant’s privacy is preserved with high probability. In addition, the integrity verification guarantees the legitimacy of the aggregated values.

To address privacy concerns in anomaly detection queries we took a different approach than relying on mutual protection. We proposed a data randomisation scheme, called RMP, which enables participants to perturb their data independently. Unlike the common trend in the literature, RMP enables private perturbation, using a combination of nonlinear and participant-specific linear perturbation. The linear transformation is
personalised by imposing random noise on a publicly available perturbation matrix. The properties of this data randomisation were altered to thwart several attacks, namely, ICA, MAP, collusion, and distance-inference attacks.

Later in our work, to enhance the level of privacy for normal and anomalous records, we redesigned the nonlinear transformation and called the scheme as ERMP. The latter scheme inherits all the advantages of RMP, as well as delivering a higher level of privacy for normal and anomalous records.

Although the perturbation matrices in RMP and ERMP vary from one participant to another, they are derived from a shared matrix. To improve the privacy even further we proposed IRMP. Following the two-stages of a nonlinear and a linear perturbation, IRMP allows participants to independently perturb their sampled records. Applying unique individual transformations has the advantage that it provides the system with stronger security against malicious collusion. To compensate for the randomness introduced to the transformation (i.e., noise imposed on the transformed data) and maintain the accuracy of the anomaly detection model, we used a deep learning architecture for our anomaly detector based on a contractive autoencoder (CAE). The CAE’s regulariser encourages the model to be invariant to small perturbations (noise) occurring in the input data, hence resulting in more robust results. Moreover, we showed experimentally that our proposed transformation assists anomaly detection, since it acts as a regulariser and improves the separability of normal and anomalous records. Further, our empirical analysis on synthetic and real-life datasets show that IRMP achieves higher overall recovery resistance than existing schemes, and comparable accuracy to non-privacy-preserving anomaly detection methods.

In all the above works we used deep autoencoders as our anomaly detector. In comparison, shallow architectures, such as 1SVMs, face two major shortcomings when it comes to large-scale high-dimensional data analysis: their computational complexity grows quadratically with the number of records, and they can fail to generate a robust model, due in the introduction of noise in the form of irrelevant features in the data. A common approach to building a robust anomaly detection model for use in high-dimensional spaces is to apply a feature extractor (i.e., preprocess the data) and then run
an anomaly detector on the reduced dataset. While a variety of feature reduction methods have been considered for SVMs, we presented a hybrid model using a DBN, called DBN1SVM. DBNs are a promising technique for learning robust features, which can be applied to large datasets and require no labelled samples, where an unsupervised DBN is trained to extract generic underlying features, and a 1SVM is trained from the features learned by the DBN. Our empirical results demonstrate that when the data are projected to a lower dimensional space using a DBN, a clearer separation appears between the normal records and anomalies thus enabling the use of linear kernels instead of more complex kernels, such as RBF kernels. As a result, DBN1SVM significantly reduces the training and testing time of the standalone techniques, while achieving a comparable accuracy to a deep autoencoder.

Another way to overcome the shortcomings of 1SVMs in processing high-dimensional datasets is to calculate a lower rank approximation of the underlying optimisation function. Random projections have been utilised mainly in distance-based classification or data reconstruction schemes to speedup search, as it approximately preserves $L_2$ distances among a set of points. Thus, instead of performing the search in a high-dimensional space, the search is conducted in a space of reduced dimension, but on a larger neighbourhood. We leveraged the theory of nonlinear random projections and propose the R1SVM, which is an efficient and scalable anomaly detection technique that can be trained on large-scale datasets. The vision of the R1SVM is to replace a nonlinear machine with randomised features and a linear machine. Our empirical analysis shows that our randomised 1SVM algorithm achieves comparable or better accuracy to deep autoencoders and traditional kernelised approaches for anomaly detection, while being approximately 100 times faster in training and testing.

In summary, we have proposed several anomaly detection techniques to overcome privacy and data-dimensionality related issues in participatory sensing networks. The privacy frameworks are highly efficient, scalable and deliver a comparable accuracy to non-privacy preserving techniques. Our high-dimensional anomaly detection techniques are highly efficient and achieve higher or comparable accuracy in detecting anomalies compared to the state-of-art techniques.
9.2 Future Research

Participatory sensing is an emerging area that has not been fully explored. PSNs impose special challenges, such as mobility in sensors, the condition and dynamics of the environment, limited computational resources and energy constraints, which affect the use of anomaly detection in participatory applications. In addition to the future work mentioned in some of the chapters, herein we present some possible directions for future research in the field of anomaly detection in PSNs.

In participatory sensing applications, data is collected over time through sensors or by people [44]. In other words, participatory data mainly consists of attributes that change over time, that is, time series. Time series database systems have been on the rise, and existing techniques for relational data are not directly applicable to time series data. Hence, PPDM schemes for time series data are increasingly important. A direction for future work is to adapt the privacy-preserving approaches such as IRMP to online learning and study the impact of various attacks over the time. For example, it is interesting to explore whether changes in the underlying data pattern gives clues to the attacker in revealing a participants’ data, and how the data perturbation should be modified to resist such attacks.

Correspondingly, our high-dimensional anomaly detection techniques can also be adapted to online learning, by incorporating incremental learning for 1SVMs. For example, currently in our DBN-1SVM, the DBN is trained with mini-batches, which is suited to real-time training. However, 1SVMs are trained with a full-batch of records. Further, it will be interesting to apply DBN-1SVM to time series, where the challenge would be how to maintain the accuracy of the model if normal behaviour evolves significantly over time.

Another research direction is to incorporate the relationship between features and present data as a tensor. Most existing data mining algorithms, including our techniques, are vector-based methods, ignoring the relationship between dimensions. Tensors enable one to include heterogeneous data from various sources and maintain the correlations across the data features. Recently there has been a lot of interest in tensor based approaches to data analysis in high dimensional spaces [35] [119]. Some data mining al-
algorithms have been generalised to handle three or higher-order tensors. Support Tensor Machine (STM) [31, 84, 164] are an extension of SVMs to more complex data, for supervised classification. Similar to SVMs, an STM requires the solution of a quadratic optimisation problem, and similarly as the data size (volume and dimensionality) grows so does the complexity, which poses serious challenges for data mining. Random projection has shown promising results as an alternative to the computationally expensive problems of kernel matrix construction and SVMs. It is interesting to investigate how to replace kernels in an STM with random projections, and also extend the general approach to a one-class STM.
Bibliography


Author/s: MONAZAM ERFANI, SARAH

Title: Anomaly detection in participatory sensing networks

Date: 2015

Persistent Link: http://hdl.handle.net/11343/58256

File Description: Anomaly Detection in Participatory Sensing Networks