CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	On the detection of privacy and security anomalies
Authors	Khan, Muhammad Imran
Publication date	2020-03
Original Citation	Khan, M. I. 2020. On the detection of privacy and security anomalies. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2020, Muhammad Imran Khan. - https://creativecommons.org/licenses/by-nc-nd/4.0/
Download date	2026-11-07 05:04:07
Item downloaded from	https://hdl.handle.net/10468/10521



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

On the Detection of Privacy and Security Anomalies

Muhammad Imran Khan

M.Sc. (Computer Science)

M.Sc. (Electrical & Electronic Engineering),

B.Sc. (Computer Engineering)



NATIONAL UNIVERSITY OF IRELAND, CORK

Faculty of Science

Department of Computer Science

Thesis submitted for the degree of
Doctor of Philosophy

March 2020

Head of Department: Prof. Cormac Sreenan

Supervisors: Prof. Dr. Barry O'Sullivan
Prof. Dr. Simon N. Foley

Research supported by Insight Centre for Data Analytics

Contents

List of Figures vi
List of Tables ix
List of Publications xii
Abstract xv
Acknowledgements xviii

1 Introduction **1**
1.1 Introduction 1
1.1.1 Insider Attacks and Anomalous Access to Databases 2
1.1.2 Detecting Privacy-Anomaly 3
1.2 Research Hypothesis 5
1.3 Key Contributions 7
1.4 Thesis Structure 10

2 Malicious DBMS Accesses - State of the Art **12**
2.1 Introduction 12
2.1.1 Threats to Contemporary Organizations 13
2.1.2 Defining Insiders 13
2.1.2.1 Masqueraders and Masquerade Attacks 15
2.1.3 The Impact of an Insider Attack 16
2.2 Anomaly Detection in Systems 17
2.3 A Taxonomy for DBMS Anomaly Detection 19
2.3.1 Prevalent Architecture of Anomaly-based Database Intrusion Detection Systems 21
2.3.2 Feature Classification 22
2.3.3 SQL Query Abstraction 23
2.3.4 Syntax-centric Features-based Techniques 24
2.3.5 Data (Result)-centric Features-based Techniques 26
2.3.6 Context-centric Features-based Techniques 27
2.3.7 Hybrid Techniques 28
2.4 Conclusions 32

3 Definitions of Privacy **33**
3.1 Introduction 33
3.2 Countless Shades of Privacy 34
3.3 Formal Privacy Definitions 36

3.3.1 Key Relational Database Terms . . . 37
3.3.2 k-anonymity based and Extensions . . . 38
3.3.2.1 k-anonymity . . . 38
3.3.2.2 l-diversity . . . 41
3.3.2.3 t-closeness . . . 43
3.3.2.4 (α, k)-anonymity . . . 43
3.3.2.5 m-invariance . . . 44
3.3.2.6 $(k, \boldsymbol{e})$-anonymity . . . 44
3.3.2.7 (ϵ, m)-anonymity . . . 45
3.3.2.8 Multi-relational k-anonymity . . . 45
3.3.2.9 δ-disclosure privacy . . . 46
3.3.3 Differential privacy (ϵ-differential Privacy) . . . 46
3.4 Summary . . . 47
3.5 Conclusions . . . 48

4 Anomalous DBMS Access Detection Using N-Grams 49
4.1 Introduction . . . 49
4.2 Modelling Normative Query Behaviour using N-Gram . . . 50
4.3 Training Phase . . . 52
4.3.1 Availability of an Anomaly-free Audit Log . . . 52
4.3.2 Chosen SQL Query Abstraction . . . 53
4.3.3 Building a Normative Model of Behaviour . . . 54
4.4 Detection Phase . . . 55
4.5 Evaluation of the N-Gram based Approach . . . 56
4.5.1 A Synthetic Data Generator for a Banking-style Application . . . 56
4.5.2 Selecting Suitable *'n'* . . . 57
4.5.3 Attack Scenarios . . . 58
4.5.4 Computational Complexity . . . 62
4.5.5 Anomaly Response . . . 62
4.6 Mimicry Attacks . . . 63
4.7 Resisting Mimicry Attacks . . . 65
4.7.1 Query Analytics-based Model of Normative Behaviour . . . 65
4.7.2 Evaluation . . . 66
4.7.2.1 Experimental Setup . . . 67
4.7.2.2 Mimicry Attack Scenarios . . . 67
4.8 Conclusions . . . 72

5 A Semantic Approach to Frequency-based Anomaly Detection of Insider

Attacks **74**
5.1 Introduction . . . 74
5.2 Record-oriented Model of Normative Behaviour . . . 76
5.2.1 Statistical Process Control and Control Charts . . . 77
5.2.2 Outlier-free Scenario . . . 79
5.2.3 Handling Outliers . . . 80
5.2.4 Oversight-anomalies . . . 81
5.2.5 Redefining Limits at Run-time . . . 81
5.3 Translating the record-oriented Model into a Role-oriented Model . . . 82
5.4 Evaluation . . . 83
5.4.1 Record-oriented: Outlier-free Scenario . . . 84
5.4.2 Record-oriented: With-outlier Scenario . . . 85
5.4.3 Role-oriented: Outlier-free And With-outlier Scenario . . . 86
5.4.4 Observations and Limitations . . . 87
5.5 Conclusions . . . 89

6 Detecting Malicious Access to DBMS using Item-set Mining **91**
6.1 Introduction . . . 92
6.2 Item-set Mining . . . 92
6.3 Querying Behaviour Modelled via Item-set Mining . . . 93
6.3.1 Query-set Mining-based Malicious Query Detection System . . . 98
6.3.1.1 Constructing Behavioural Profiles using Item-set Mining . . . 98
6.3.1.2 Comparing Frequent Query Profiles . . . 100
6.3.1.3 Mining Rare Query-sets . . . 100
6.3.1.4 Selecting a Threshold Value For Support . . . 101
6.4 Evaluation . . . 101
6.4.1 Mimicry Attacks: Frequent Queries as Malicious Behaviour . . . 103
6.4.2 Frequent, Rare, and In-between . . . 105
6.4.3 Comparison with N-gram Approach . . . 105
6.4.4 Complexity of Item-set Mining Approach . . . 106
6.4.5 Potential Application . . . 106
6.4.6 Sequential Pattern Mining . . . 107
6.5 Conclusions . . . 108

7 Privacy Interpretation of Behavioural-based Approaches **109**
7.1 Introduction . . . 109
7.2 Privacy-Anomaly Detection (PAD) System . . . 111

7.2.1 A *k*-Anonymity based Privacy-profile . . . 112
7.2.1.1 Mining *k*-anonymity based Profiles for PAD . . . 113
7.2.1.2 Detecting Privacy-anomalies . . . 114
7.2.2 Computational Complexity . . . 115
7.3 Security-anomaly Detection System Detecting Privacy-anomalies . . 116
7.3.1 Detected Privacy-anomalies . . . 118
7.3.2 Undetected Privacy-anomalies . . . 119
7.3.3 Identifying Appropriate Privacy Limits . . . 120
7.4 *k*-anonymity and Discrimination based Privacy-anomaly Detection System . . . 120
7.4.1 Discrimination Rate (DR) . . . 121
7.4.2 *DRSQL*: Computing Identification Capability of SQL Queries 123
7.4.3 Application of *DRSQL*: Privacy Comparison v/s Simple Comparison . . . 125
7.4.3.1 Privacy Ordering Relations . . . 126
7.4.4 A *k*-Anonymity and Discrimination Rate based Privacy-profile 128
7.4.4.1 Composing Privacy Criteria . . . 129
7.4.4.2 Example Run of (*k*-anonymity, *DR*)-PAD . . . 130
7.5 Privacy Attacks (Inferences) . . . 132
7.6 Applying Security-anomaly Detection to Detect Unknown Privacy Attacks . . . 134
7.6.1 Detecting Inferences as Anomalies . . . 135
7.7 Conclusions . . . 138

8 PriDe: A Quantitative Measure of Privacy 140
8.1 Introduction . . . 140
8.2 PriDe - The Model . . . 141
8.2.1 Difference between Naïve Calculation and PriDe Model . . . 141
8.2.2 The Design . . . 142
8.2.3 Constructing Profiles of Querying Behaviour . . . 144
8.2.3.1 Comparison of Profiles . . . 145
8.2.3.2 Distance Between n-grams . . . 146
8.2.3.3 Privacy Equivalence . . . 147
8.2.3.4 Computing The Score . . . 148
8.2.4 The Scenario of Cold Start . . . 150
8.3 Cumulative Score . . . 151
8.3.1 Max (worst-case) Score . . . 152

8.3.2 Properties of PriDe 153
8.4 Evaluation 154
8.4.1 Computational Complexity of PriDe 157
8.4.2 Use-case: Global Consistency 159
8.5 Conclusions 160

9 Conclusions and Future Work **161**
9.1 Conclusions 161
9.2 Future Work 165
9.2.1 Similarity Index for SQL Statements 165
9.2.2 Scarcity of Real-world or Benchmark Datasets 166
9.2.3 Handling Concept Drift in Behaviours 166
9.2.4 Defence in Depth: Privacy Perspective 167
9.2.5 Translation onto Other Data Models 167
9.2.6 Explaining Anomalies 168
9.2.7 Instantiation of PAD with Multiple Privacy Models 168

List of Figures

2.1 Figure depicting insiders, outsiders, internal and external masqueraders in organizational settings. 16
2.2 Taxonomy of anomalous DBMS-access detection systems. 20
2.3 Training phase of an anomaly detection system. 21
2.4 Detection phase of an anomaly detection system. 21
2.5 List of features considered. Figure cropped from *Securing Data Warehouses from Web-Based Intrusions* by Santos et al. [1] 30

3.1 The state of Data protection and privacy legislation worldwide [2]. . . 35

4.1 Training phase of the proposed n-gram based model. The first step is of data collection in the form of audit logs. Abstraction is chosen in the second step, followed by the construction of a normative profile. . 51
4.2 Detection phase of the proposed n-gram based model. The data collection, SQL query abstraction and profile construction steps are same as of training phase. In the detection phase, a *run-time profile* is constructed from run-time logs. The run-time profile is compared with the normative profile. Mismatches are an indication of an anomaly. 52
4.3 Discovered number of n-grams when comparing profiles. The zeros on the x-axis show the comparison of the normative profile with itself. 58
4.4 The figure shows the number of mismatched n-grams, for various sizes of n-gram, that indicated the presence of anomalous queries (attacks) in the audit logs. Attack 1, 2, & 3 represents profiles constructed using logs $L_{\mathcal{R}1}$, $L_{\mathcal{R}2}$, and $L_{\mathcal{R}3}$, respectively. 61
4.5 A sample n-gram from the normative profile, depicting a transfer of an amount from one client's bank account to another client's bank account. 63
4.6 This figure depicts queries made by an inside attacker. First, an amount is transferred from the first victim's account to the second victim's account, and then the same amount is transferred back to first victim's from second victim's account. 64
4.7 The figure shows the values of φ_{P_i} for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 2 (attack 2 profile). 68
4.8 The figure shows the values of $\varphi_{P_i}^{SELECT}$ for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 1 (attack 2 profile). 68

4.11 The figure shows the values of $\varphi_{P_i}^{DELETE}$ for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 2 (attack 2 profile). 69
4.9 The figure shows the values of $\varphi_{P_i}^{INSERT}$ for the normative profile as compared to attack 1 profile and attack 2 profile. 69
4.10 The figure shows the values of $\varphi_{P_i}^{UPDATE}$ for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 2 (attack 2 profile). 69
4.12 The figure shows the values of φ_{Γ_l} for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 2 (attack 2 profile). 70

5.1 The adopted variation of control chart. 78
5.2 A fragment of sample relation $\mathcal{T}$ from the Patient database. 83
5.3 The figure shows the control chart developed during the detection phase, while the training dataset was outlier-free. The solid coloured circles (•) represents record access frequencies. The anomalies are indicated in red circles. 85
5.4 The figure shows the record access frequencies plotted for 10 days over the control chart while the training dataset is with-outlier. 86
5.5 The figure shows the control chart for the role of Consultant in the outlier-free scenario. The control chart presents the frequency of record accesses by the role Consultant for the duration of 10 days. . . 87
5.6 The figure shows the control chart for the role of Nurse in the with-outlier scenario. 87

6.1 Detection Phase: the first step is data collection; the second step is of query abstraction, followed by run-time profile construction. The run-time profile is compared against the normative profile (baseline profile), and infrequent query-sets are mined. 99

7.1 The figure shows the number of mismatches between $ngram(Abs(L_{test1}^{hosp}), n)$ and $ngram(Abs(L_{test2}^{hosp}), n)$ for different values of n. 117
7.2 A fragment of privacy-aware attribute relationship diagram. 127

8.1 The run-time profile is compared with the baseline profile resulting in a score. Each n-gram from the set of mismatched n-grams is compared with each n-gram in the baseline profile. The minimum value of the Jaccard distance is taken from one iteration of comparison and, subsequently, all the minimum values are added together to get the score. 150

8.2 Variations of score computations: This figure shows a variety of ways in which individual scores and cumulative scores can be computed. For example, $\Delta[<\beta_0>,<\beta_1,\beta_2>]$ represents the cumulative score for day 1 and day 2. $P_{\langle\beta_0,\beta_3\rangle}$ represents the individual score for day 3. . . . 152

8.3 This graph shows the results for the scenario of the banking settings. The red bar in the figure shows the actual score while the green bar in the figure indicates the maximum possible score (worst-case score – $\mathcal{M}_{\beta_X}$) for each day. The blue line shows the cumulative score. 155

8.4 This graph shows the results of the cold start scenario. The red bar shows the actual score, and the blue line shows the cumulative score over the time horizon. 155

List of Tables

2.1 Identifiers example for the query abstraction approach proposed in [3]. 25
2.2 An overview of the characteristics discussed and well-known approaches proposed in the literature. ⊛, ⊙, and ⌔ represents syntax, data(result), and context-centric features respectively. 31

3.1 An example relation $\mathcal{T}$. 38
3.2 A relation with `Name` as an identifier, `Age` & `Zipcode` are quasi-identifiers and `Salary` as a sensitive attribute. 40
3.3 A 3-anonymized version of Table 3.2. 40
3.4 A l-diverse (3-diverse) version of Table 3.2. 42

4.1 Examples of deployed SQL abstractions. 54
4.2 Attack logs and scenarios. 59
4.3 The table shows the values of φ_{P_i}, averaged over the time frame of τ, made part of the profile of normative behaviour. 71
4.4 The table shows the values of φ_{Γ_l}, averaged over the duration of 30 days, made part of the normative profile. 72

6.1 An example of a query-set table where a query-set QS_i is analogous to an item-set, while each query is analogous to an item in item-set. . . . 94
6.2 The table shows frequent query-sets mined from the query-set table shown in Table 6.1. The minimum support for the frequent query-set is set to 3 (or 37.5%). 95
6.3 Rare query-sets for the query-set table shown in Table 6.1. The query-sets in this table have the support of less than 3 (or 37.5%). 96
6.4 The table shows Perfectly rare + Minimal rare query-sets mined from the query-set table shown in Table 6.1. The query-sets in this table have the support of less than 3 (or 37.5%). 98
6.5 The table showing the make up of attacks and attack logs in the case of mining rare query-sets. 102
6.6 The number of mined rare query-sets that are an indication of anomalies. 103
6.7 The table shows the length of the attack logs, the length of the query sequences and the number of times the rare query sequence was inserted in the attack logs. 104
6.8 The number of mismatches when the normative profile is compared with attack profiles. 104

7.1 A fragment of relation `temp_table`. 112
7.2 A relation $\mathcal{T}_{R1}$ resulting from the query `SELECT age, zipcode FROM temp_table WHERE gender = 'Male';`. 113
7.3 A relation $\mathcal{T}_{R2}$ resulting from the query `SELECT age, zipcode, county FROM temp_table WHERE gender = 'male';`. 114
7.4 A relation $\mathcal{T}_{R3}$ resulting from the query `SELECT age, zipcode FROM temp_table WHERE gender = 'female';`. 115
7.5 A fragment of hospital dataset.The strike-through attribute values represents a deleted row. 118
7.6 Description of Privacy-anomalies injected. 118
7.7 Response to a undetected privacy-anomalous query. 119
7.8 An example relation `companyTable`. 122
7.9 Discrimination rate values for attribute shown in Table 7.8. 123
7.10 Computed combined discrimination rate (CDR) values for attributes in the relation shown in Table 7.8. 123
7.11 Deployed specialization of SQL query abstraction. The elements of $Abs(Q_i)$ are the attributes in the SQL query. 124
7.12 Sample SQL queries used to query the `companyTable` relation shown in Table 7.8. 125
7.13 A sample relation `table_smp` of records for the running example. . . 131
7.14 Sample SQL queries executed over the relation `table_smp` shown in Table 7.13 131
7.15 Records returned in the response to query Q_1. 132
7.16 Records returned in the response to query Q_2. 132
7.17 Relation `updated_table_smp`: updated version of the `table_smp` relation shown in Table 7.13. 133
7.18 Sequence of queries executed over the relation `updated_table_smp` shown in Table 7.17. 133
7.19 Records returned in the response to query Q_2. 134
7.20 Records returned in response of query Q_3. 134
7.21 Inserted unique records in the database to enable privacy attacks. . . . 137
7.22 Length of the query sequences to reveal salaries. 137
7.23 Detection of privacy attacks (inferences) as anomalies: the table shows the number of mismatches that resulted from each of the 5 privacy attacks with the n-gram of size 4. 138

8.1 An example table having records of five individuals. 143

8.2 Deployed SQL query abstraction. 144
8.3 Sample SQL queries. 147

List of Publications

Parts of this work have appeared in the following peer-review international publications.

1. M. I. Khan, S. N. Foley, and B. O'Sullivan, "Quantitatively Measuring Privacy in Interactive Query Settings within RDBMS," *Frontiers in Big Data*, (Accepted), March 2020.

2. M. I. Khan, S. N. Foley, and B. O'Sullivan, "PriDe: A Quantitative Measure of Privacy-Loss in Interactive Querying Settings," in *New Technologies, Mobility and Security: Proc. of the 10th IFIP International Conference, NTMS 2019, Canary Island, Spain, June 24-26, 2019,* IEEE, 2019. pp. 1 - 5.

3. M. I. Khan, S. N. Foley, and B. O'Sullivan, "Computing the Identification Capability of SQL Queries for Privacy Comparison," in *Security and Privacy Analytics: Proc. of the 5th ACM International Workshop, IWPSA 2019, co-located with the 9th ACM Conference on Data and Application Security and Privacy (CODASPY 2019), Dallas, TX, USA, March 25-27, 2019,* ACM, 2018. pp. 47 - 52.

4. M. I. Khan, B. O'Sullivan, and S. N. Foley, "Towards modeling Insiders Behaviour as Rare Behaviour to Detect Malicious RDBMS Access," in *Big Data: Proc. of the IEEE International Conference, IEEE Big Data 2018, Seattle, WA, USA, December 10 - 13, 2018,* IEEE, 2018. pp. 3094 - 3099.

5. M. I. Khan, S.N. Foley, & B. O'Sullivan, "DBMS Log Analytics for Detecting Insider Threats in Contemporary Organizations," in *Security Frameworks in Contemporary Electronic Government*, R. Abassi, & A. Ben Chehida Douss, Eds. 2018. pp. 207 - 234.

6. M. I. Khan, S. N. Foley, and B. O'Sullivan, "On database intrusion detection: A query analytics-based model of normative behavior to detect insider attacks," in *Communication and Network Security: Proc. of the 7th International Conference, ICCNS 2017, Tokyo, Japan, November 24 - 26, 2017,* ACM, 2017. pp. 12 - 17.

7. M. I. Khan, B. O'Sullivan, and S. N. Foley, "A semantic approach to frequency based anomaly detection of insider access in database management systems," in *Risks and Security of Internet and Systems: Proc. of the 12th International Conference, CRiSIS 2017, Dinard, France, September 19 - 21, 2017,* N. Cuppens, F.

Cuppens, J.-L. Lanet, A. Legay, J. Garcia-Alfaro, Eds. Berlin: Springer, 2017. pp. 18 - 28.

8. M. I. Khan and S. N. Foley, "Detecting anomalous behavior in dbms logs,"in *Risks and Security of Internet and Systems, Proc. of the 11th International Conference, CRiSIS 2016, Roscoff, France, September 5 - 7, 2016,* F. Cuppens, N. Cuppens, J.-L. Lanet, A. Legay, Eds. Berlin: Springer, 2017. pp. 147 - 152.

Other publications/presentations related to the dissertation:

1. M. I. Khan, S. N. Foley, & B. O'Sullivan, "On Privacy Comparison of SQL Queries," 6*th* *Insight Student Conference (INSIGHT SC 2020)*, National University of Ireland, 12*th* February 2020, Galway, Ireland.
2. M. I. Khan, B. O'Sullivan, & S. N. Foley, "Computing a quantitative score for privacy," 5*th* *Insight Student Conference (INSIGHT SC 2018)*, University College Dublin (UCD), 7*th* September 2018, Dublin, Ireland.
3. M. I. Khan, B. O'Sullivan, & S. N. Foley, "Detecting anomalous insider accesses to relational databases - A semantic approach," 4*th* *Insight Student Conference, (INSIGHT SC 2017)*, University College Cork (UCC), 8*th* September 2017, Cork, Ireland.
4. M. I. Khan, B. O'Sullivan, & S. N. Foley, "On Detection of Anomalous Query Sequences," 3*rd* *Interdisciplinary Cyber Research workshop (ICR 2017),* 8*th* July, 2017, Tallinn, Estonia.
5. M. I. Khan, B. O'Sullivan, & S. N. Foley, "DBMS Log Analytics For Insider Attack Detection," *Data Summit 2017*, 15*th* June 2017, Dublin's Convention Centre, Dublin, Ireland.
6. M. I. Khan, & S. N. Foley, "Towards Insider Attacks Detection in Database Management Systems," 3*rd* *Insight Student Conference 2016, (INSIGHT SC 2016)* University College Dublin (UCD), 14*th* September 2016, Dublin, Ireland.
7. M. I. Khan, S. N. Foley, and B. O'Sullivan, "Privacy Interpretation of Behavioural-based Anomaly Detection Approaches," (Under Review).
8. M. I. Khan, S. N. Foley, and B. O'Sullivan, "Privacy-anomalies: Detecting Unknown Privacy Attacks as Anomalies in Interactive Query Setting," (Under Review).
9. M. I. Khan, S. N. Foley, and B. O'Sullivan, "Privacy-anomaly Detection: Discovering Correlation between Security and Privacy-anomalies," (Under Review).

I, Muhammad Imran Khan, certify that this thesis is my own work and I have not obtained a degree in this university or elsewhere on the basis of the work submitted in this thesis.

Muhammad Imran Khan

Abstract

Data analytics over generated personal data has the potential to derive meaningful insights to enable clarity of trends and predictions, for instance, disease outbreak prediction as well as it allows for data-driven decision making for contemporary organisations. Predominantly, the collected personal data is managed, stored, and accessed using a Database Management System (DBMS) by insiders as employees of an organisation.

One of the data security and privacy concerns is of insider threats, where legitimate users of the system abuse the access privileges they hold. Insider threats come in two flavours; one is an insider threat to data security (security attacks), and the other is an insider threat to data privacy (privacy attacks). The insider threat to data security means that an insider steals or leaks sensitive personal information. The insider threat to data privacy is when the insider maliciously access information resulting in the violation of an individual's privacy, for instance, browsing through customers bank account balances or attempting to narrow down to re-identify an individual who has the highest salary. Much past work has been done on detecting security attacks by insiders using behavioural-based anomaly detection approaches. This dissertation looks at to what extent these kinds of techniques can be used to detect privacy attacks by insiders.

The dissertation proposes approaches for modelling insider querying behaviour by considering sequence and frequency-based correlations in order to identify anomalous correlations between SQL queries in the querying behaviour of a malicious insider. A behavioural-based anomaly detection using an n-gram based approach is proposed that considers sequences of SQL queries to model querying behaviour. The results demonstrate the effectiveness of detecting malicious insiders accesses to the DBMS as anomalies, based on query correlations. This dissertation looks at the modelling of

normative behaviour from a DBMS perspective and proposes a record/DBMS-oriented approach by considering frequency-based correlations to detect potentially malicious insiders accesses as anomalies. Additionally, the dissertation investigates modelling of malicious insider SQL querying behaviour as rare behaviour by considering sequence and frequency-based correlations using (frequent and rare) item-sets mining.

This dissertation proposes the notion of '*Privacy-Anomaly Detection*' and considers the question whether behavioural-based anomaly detection approaches can have a privacy semantic interpretation and whether the detected anomalies can be related to the conventional (formal) definitions of privacy semantics such as *k*-anonymity and the discrimination rate privacy metric. The dissertation considers privacy attacks (violations of formal privacy definition) based on a sequence of SQL queries (query correlations). It is shown that interactive querying settings are vulnerable to privacy attacks based on query correlation. Whether these types of privacy attacks can potentially manifest themselves as anomalies, specifically as privacy-anomalies, is investigated. One result is that privacy attacks (violation of formal privacy definition) can be detected as privacy-anomalies by applying behavioural-based anomaly detection using n-gram over the logs of interactive querying mechanisms.

Again,
To Humanity.

Acknowledgements

First of all, I would also like to express my sincere gratitude to my supervisor (Prof. Dr. Simon N. Foley), for his guidance, nurturing, support and generous encouragement. He has been supportive as well as critical at every stage of my research work. He has been a mentor to me; without his guidance, this research work would not have yielded such positive results. I've learnt a lot from his broad knowledge, insight, kindness and patience. It is due to Simon's invaluable feedback that had shaped and matured my research work.

I owe a debt of gratitude to my supervisor (Prof. Dr. Barry O'Sullivan), for his continuous help and guidance throughout my research work. My PhD journey would not have been possible without his continuous support, encouragements and supervision. Barry's insightful feedback on my research work has helped me to see the bigger picture. This work would not have been realized without Barry's persistent support.

Further, I would also like to acknowledge the administrative support of the Caitrŕiona Walsh, Eleanor O'Riordan, Peter McHale, Chrys Ngwa and Linda O'Sullivan during my PhD research.

I also would like to extend my gratitude to my colleagues at Insight Center for Data Analytics and University College Cork for making my time enjoyable along with their encouragement. I wish to personally thank my friends and colleagues that I met during this journey Dr. Begum Genc, Dr. Mojtaba Montazery, Dr. Milan De Cauwer, Dr. Anne-Marie George, Hong Huang, Ali Naeem, Dr. Ahmed Khalid, Qiao Cheng, Federico Toffano, Andrea Visentin, Diego Carraro, and Dr. Vincent Armant, and Dr. Kabir Ali for their emotional support, camaraderie and with whom I've spent most of my leisure time at during my stay at Insight Centre.

I would like to thank my external and internal examiners for taking out the time to read

my dissertation.

I would also like to acknowledge all the authors of open source codes, as well as the authors of the research work with which the presented work is compared. I also would like to acknowledge Dr. Gokhan Kul, Assistant Professor at Delaware State University, USA, Dr. Asmaa Sallam at Purdue University, USA, and Dr. Louis at Orange and IMT Atlantique, France for constructive collaborations during the course of this work.

I would like to thank my Father, my Mother, my Sister and my Brother for their support, care, continuous guidance and advice.

Last but not least, I would like to acknowledge my better half Dr. Fareeha, who bore me in my most difficult times. Her indirect contribution made this work possible.

My work would not have been possible without the support of Science Foundation Ireland Grant No. 12/RC/2289, which is co-funded under the European Regional Development Fund.

Chapter 1

Introduction

"Every new beginning comes from some other beginning's end."

Seneca (5 BC - 65 AD)

1.1 Introduction

The recent past has witnessed an exponential increase in the volume of data being collected by organizations. This has been enabled by the aggressive development of computing technologies. Data is fuelling most of the revolutionary technologies. Technological advances and widely available data have nurtured the area of data analytics. Data analytics aims to discover meaningful insights from the data that may lead to improved decision-making. Data analytics offers a broad spectrum of benefits, for example, it enables a contemporary organization to anticipate business opportunities as well as enable the delivery of relevant products to its customers. In a nutshell, analytics over a large volume of data has the potential to impact businesses and our society.

Data comes from multiple sources and may include sensitive personal data. On the one

hand, one cannot deny the importance and value of data, while on the other hand, the usage, storage, and access to this data can raise security and privacy concerns.

1.1.1 Insider Attacks and Anomalous Access to Databases

Contemporary organizations systems rely heavily on Database Management Systems (DBMS) to store and manage access to their application data. Organizations need to take extra care in the management and storage of sensitive application data. Misuse or leakage of such data can lead to an organization suffering from damages in terms of reputation and financial loss. The harm caused by data breaches has routinely been reported in the popular press.

Threats to an organization's data come from external attackers - *outsider attacks* or internal attackers - *insider attacks*. Traditional security controls such as authentication, role-based access control, data encryption and physical-security can help to control access to this data. However, there is a persistent concern of insider attacks whereby legitimate users of the system abuse the access privileges they hold. A recent survey reported that malicious insiders are the cause of the costliest cybercrimes [4], and in one study 89% of respondent organizations reported themselves vulnerable be to insider attacks [5]. A further study reports that a significant level (43%) of data exfiltration was caused by insiders and half of which was intentional [6]. A challenge in insider attack detection is that they often go unnoticed for months and years [7]. Therefore, effective security controls to mitigate insider attacks are desirable.

An Intrusion Detection Systems (IDS), in particular, behavioural-based intrusion detection systems (also referred to as behavioural-based anomaly detection systems) [8, 3, 9], can play a role in detecting insider attacks. Behavioural-based anomaly detection systems model normative behaviour of the system user and look for deviations in the run-time behaviour of the users. The primary challenge in designing an anomaly-based system is how to model normative behaviour.

This dissertation explores how to model this normative behaviour of users of DBMS, in order to detect malicious accesses by an insider as anomalies specifically as security-anomalies. Much of existing literature considers database queries in isolation to model the querying behaviour of a user [3, 10, 11, 12]. Modelling approaches that consider queries in isolation can detect a single malicious query, however, are unable to detect a sequence of queries where every single query in a sequence is not malicious, but the entire query sequence results in a malicious event. An approach for capturing short-term query correlations to model querying behaviours, based on n-grams, from the audit logs of DBMS is proposed in this dissertation. The approach considers sequences of queries rather than a query in isolation. This leads to a novel *white-box* [13] behavioural-based approach for the detection of anomalous DBMS accesses by an insider as anomalies.

Additionally, the dissertation proposes a novel notion of modelling user behaviour from a DBMS perspective. Referred to as a *semantic approach*, it utilizes control charts [14] from the statistical process control domain and detects anomalous accesses to the databases. In this dissertation, the application of machine learning algorithms, specifically Frequent and Rare Item-sets Mining algorithms, to model user's repeated and infrequent (rare) querying behaviour is explored in order to detect insiders malicious behaviour as a rare behaviour.

1.1.2 Detecting Privacy-Anomaly

The recently enacted EU's General Data Protection Regulation (GDPR) [15] makes it more challenging to use personal data for analytics. However, once the data is anonymized, it is considered to no longer be personal data [15, 16]. Achieving anonymization is non-trivial while preserving the utility of data. Increasing the level of anonymization protects data but reduces utility. Thus organizations must trade-off the need for more in-depth analytics against the privacy of individual's data. In essence, it

is a long-standing open problem to get high-quality analytics by querying the databases consisting of information about individuals while preserving the privacy of those individuals.

Numerous incidents have been reported where privacy was compromised due to poor anonymization of released data, for example, the famous case of Netflix [17], AOL [18], de-anonymization of NYC taxi data [19], and the famous case of the Massachusetts Governor [20]. In [20], it was shown that by linking on shared attributes (zipcode, birth date, and gender) in two datasets, Massachusetts Group Insurance Commission's released data (considered anonymous) and voter rolls, records belonging to the Massachusetts Governor were identified. Researchers have devised formal privacy definitions[1] [21, 22, 23, 24, 25, 26] when these definitions are followed then the anonymized data manifests some formal guarantees. There are several privacy definitions to anonymize data, including, k-anonymity, l-diversity, t-closeness, and differential privacy.

The majority of the syntactic privacy definitions were designed for a one-time release of data. In contrast to these definitions, differential privacy is for interactively querying a database. However, differential privacy has practical limitations as well; for instance, differential privacy allows only a limited number of queries to be answered. Allowing an unlimited number of queries results in higher noise; thus, the ability to observe correlations between attributes are lost, which is not desirable for richer analytics [27]. Approaches to detect privacy violations, while allowing an unlimited number of queries while having richer analytical utility are desirable.

This dissertation discovers that the malicious accesses to the DBMS detected as anomalies by the traditional (security-)anomaly detection approaches (the proposed n-gram based anomaly detection approach in this dissertation) on closer analysis reveals that those anomalies, in a sense, are privacy violations. In this dissertation, we

[1] Privacy definitions, in literature, are otherwise known as privacy models, privacy criteria, privacy metrics, privacy constraints as well as privacy principles.

carried out a study to investigate whether these privacy violations can be called privacy violations from a formal privacy definition perspective, that is, violation of a privacy definition. Additionally, in this dissertation, a novel privacy-oriented interpretation of behavioural-based detection approach is proposed. It is shown that a behavioural-based detection approach when applied to the logs of a k-anonymity based interactive query system, have the potential to detect violation of privacy definition and instances of privacy attacks (inferences) as anomalies specifically privacy-anomalies.

1.2 Research Hypothesis

The hypothesis in this dissertation is that *"behavioural-based anomaly detection techniques for detecting malicious DBMS access can be used to detect violations of privacy definition and instance of privacy attacks (inferences) as anomalies (referred to as privacy-anomalies)"*.

In order to explore this hypothesis, a number of research questions have been identified.

RQ1: Is it possible to build behavioural-based anomaly detection systems to detect malicious accesses, manifested in sequences of queries rather than a query in isolation, to DBMS and whether these malicious accesses encompasses privacy violations?

It is known that behavioural-based approaches are able to detect intrusions in networks and computer systems [28, 29, 30, 31, 32, 33]. There are techniques in literature that detect malicious queries made by insiders to DBMS [3, 11, 34, 12, 35, 36]. While such approaches can certainly be used to detect individual queries that are entirely different from normal (similar queries), they cannot identify an insider that mimics valid queries. In this case, it is insufficient to identify the anomaly by the query alone, we must also correlate it with its surrounding (sequence of) queries. The question that arises is whether one can

build behavioural-based anomaly detection systems for detection of malicious accesses to DBMS by considering sequences of queries. Chapter 4, addresses this question, where a behavioural-based approach is built that captures querying patterns using n-grams in an audit log of SQL queries generated by an application system. The normative behaviour is modelled using n-grams by considering sequences of queries.

A related question that arises by looking at the detected malicious access is what these malicious accesses represent. Do these malicious accesses encompasses privacy violations? A closer look at the detected malicious accesses, it was observed that these malicious accesses do encompass privacy violations of individuals privacy as well.

RQ2: Whether one can consider non-sequence based correlations manifested in features such as frequencies to model normative behaviour in the context of detecting malicious DBMS access by insiders? Is it possible to capture distant correlations to model querying behaviour in contrast to short-term correlations that were captured by n-grams?

Chapters 5 and 6 explore ways of modelling querying behaviour. A DBMS-oriented perspective of modelling access to DBMS is presented in Chapter 5 by considering record access frequencies. The machine learning domain was examined, specifically Item-set Mining to model querying patterns that captured distant correlations, in Chapter 6.

RQ3: A key question is whether we can provide a privacy semantics for these behavioural-based approaches or relate the notion of privacy-anomaly detection to the conventional definitions of privacy semantics?

Following on from the discovery that the detected malicious accesses to DBMS encompass privacy violations of individuals privacy, another question that emerges is whether we can have an interpretation of detected privacy viola-

tion in-terms of formal definitions of privacy. Chapter 7 addresses this question and explores from various perspectives the privacy semantic interpretation of behavioural-based approaches. A novel notion of *privacy-anomaly detection systems* based on formal definitions of privacy (*k*-anonymity and *Discrimination rate*) is presented. It is examined whether there is an overlap between the violations of privacy definitions detected by the privacy-anomaly detection system and the malicious accesses detected by the security-anomaly detection system. Another interpretation of behavioural-based anomaly detection (the n-gram based security-anomaly detection system) is presented in Chapter 7 which shows that the behavioural-based anomaly detection system, when applied to the logs of a *k*-anonymity-based privacy-anomaly detection system in interactive query settings, has the potential to detect instances of privacy attacks (inferences) as anomalies precisely as privacy-anomalies.

1.3 Key Contributions

The key contributions within this dissertation are summarised as follows.

- **N-gram based approach to detect malicious DBMS access:** a *white-box* anomaly detection system based on n-grams of SQL queries is proposed. It is quite possible that a single query may not be malicious, but a sequence of queries can result in an undesirable event. Therefore, the n-gram based approach considers sequences of queries to model normative behaviour. The model of normative behaviour captures the users querying behaviour instead of considering an SQL query in isolation. The dissertation describes experiments to judge the effectiveness of using an n-gram based scheme to detect query anomalies in DBMS logs and thereby identify insider attacks. The model demonstrates that it is possible to build a useful query abstractions and that n-grams of these query abstractions

do capture the short-term correlations inherent in the application. The approach detects malicious accesses to databases by insiders leading to privacy violations within the relational database management framework.

- **DBMS-oriented behavioural modelling to detect malicious DBMS access:** a different perspective on the construction of the normative behaviour is introduced, that is the notion of *DBMS/record-oriented* behavioural modelling. Traditional approaches can be referred to as role-oriented behavioural modelling where normative behaviour of users are modelled according to the user roles in contrast to DBMS-oriented modelling. This perspective gives rise to a semantic approach to frequency-based anomaly detection of malicious insider accesses to databases. The DBMS/record-oriented model of normative behaviour is constructed and utilizes techniques from statistical process control. Two scenarios were considered, in the first scenario, the training data contains outliers, and in the second scenario, the training data is free from outliers. The experiments demonstrate the effectiveness of the proposed approach in the detection of frequent observation attacks as well as anomalies introduced due to human negligence or human errors. An example of human negligence is an instance where the doctor or the nurse (caregiver) missed the daily check-up of a patient and thus forgot to update the patients' record in the database. To the best of our knowledge, the proposed semantic approach labels these unique type of anomalies as it not only identifies unseen behaviours but also identifies behaviours that should have been present in the current behaviours. It is also demonstrated that the proposed DBMS-oriented behaviour modelling can be transformed into a role-oriented behaviour modelling.

- **Modelling rare query behaviour:** an alternative anomaly detection system is presented for the detection of insider attacks as security-anomalies. The approach considers sets of queries to model behaviour using item-set mining al-

gorithms. Based on the notion that behaviours that are rare (infrequent) represent possible anomalies, while frequent behaviours can be considered safe, rare (Apriori-Inverse [37] and Apriori-Rare [38]) and frequent (PrePost$^+$ [39]) item-set mining algorithms are adopted for modelling querying behaviour. Unlike n-grams, it is shown that the item-sets mining based anomaly detection approach has the potential to capture distant/long-term correlations inherent in the application.

- **Privacy semantic interpretation of behavioural-based approach:** a novel notion of *privacy-anomaly detection* is proposed. The idea is to learn user's past querying behaviour in terms of privacy and then identifying deviations from past behaviour in order to detect privacy violations – *privacy-anomalies*. Two instantiations of privacy-anomaly detection systems are proposed. The first instantiation can be considered a naïve interpretation of k-anonymity [24]. The second instantiation combines k-anonymity and *Discrimination Rate Privacy Metric* [40]. It is demonstrated that the privacy-anomalies that were being detected by the privacy-anomaly detection system can also be detected by the n-gram based approach (security-anomaly detection system).

  Privacy definitions when used in interactive query settings are susceptible to privacy attacks based on query correlations (inferences) leading to privacy violations. Much attention has been paid on mitigating inference attacks by suggesting stronger privacy definitions, for instance, using differential privacy, but this has its practical limitation. It is demonstrated that the application of an n-gram based approach can potentially detect privacy attacks as privacy-anomalies.

- **Identification Capability of SQL queries:** an interpretation of a recently published privacy metric, known as Discrimination Rate Privacy Metric, for SQL query is presented. The approach is based on information theory and enables one to measure the identification capability of an SQL query.

- **PriDe:** a model that quantifies the privacy-loss between querying behaviours is proposed. The proposed approach serves as a good tool to provide a global consistency, i.e. if firm A's privacy score is much higher as compared to the rest of the firms, then this alerts the organization to take imperative measures before a breach is materialized. The score can augment a privacy dashboard that provides the health of the system in terms of privacy.

1.4 Thesis Structure

The thesis is organized in the following manner:

Chapter 2 presents a literature review of anomaly-based detection of insider attacks. The chapter presents the detailed background and preliminaries of the research in focus, which paves the path to understand the presented work. A taxonomy of anomaly detection methods that detect anomalous access to a DBMS is also introduced in this chapter.

Chapter 3 focusses on the literature review that covers the privacy aspects of the dissertation. The chapter introduces the notion of privacy and reviews a number of privacy models (definitions) that fall within the scope of the dissertation.

Chapter 4 presents an anomaly-based detection approach to detect insider attacks. The proposed approach uses n-grams to capture querying patterns in an audit log of SQL queries generated by an application system by considering sequences of queries.

Chapter 5 introduces a novel notion of DBMS/record-oriented modelling that leads to the development of a semantic approach to frequency-based anomaly detection of malicious insider accesses to databases.

Chapter 6 investigates alternatives approaches to model querying behaviour. An alternative approach for detection of insiders malicious access to DBMS is proposed that

deploys frequent and rare item-sets mining (PrePost$^+$, Apriori-Inverse and Apriori-Rare algorithms) to model querying behaviour of an insider.

In Chapter 7, a privacy semantic interpretation of a behavioural-based anomaly detection approaches is presented. A notion of *privacy-anomaly detection* is also introduced. Two instantiations of privacy-anomaly detection systems that are, a naïve instantiation based on k-anonymity and an intricate instantiation by combining k-anonymity and *Discrimination Rate* are demonstrated. In the chapter, it is investigated whether privacy violations can be detected as anomalies by a security-anomaly detection system, in particular, n-gram based approach. This chapter also puts forward a way to compute the identification capability of SQL queries. An application of n-gram based approach to detect privacy attacks based on query correlation as privacy-anomalies is also shown in this chapter.

Chapter 8 presents *PriDe* a model that quantifies and measures the privacy-loss between querying behaviour interactive query session.

Lastly, Chapter 9 concludes the dissertation and outlines the direction for future work.

Chapter 2

Malicious DBMS Accesses - State of the Art

"A little learning is a dangerous thing; Drink deep, or taste not the Pierian spring".

Alexander Pope (1688-1744)

2.1 Introduction

Contemporary organization systems rely heavily on Database Management Systems (DBMS) to store and manage access to their application data. This data can be sensitive in nature, thus giving rise to security and privacy concerns. This chapter presents background on the Database Intrusion Detection relevant to the thesis. Section 2.1.1 considers the threats to contemporary organizations, while Section 2.1.2 and 2.1.3 defines and discusses the impact of an *insider threat*. Section 2.2 explores the detection methods for insider attacks. Section 2.3 presents a taxonomy of anomaly-based

detection methods and reviews the well-known techniques reported in the literature.

2.1.1 Threats to Contemporary Organizations

Threats to an organization can be classified as external (outsider attack) or internal (insider attack). External threats come from attackers outside of the organization who discover network and/or system vulnerabilities and use this information to penetrate the organization. Outside attackers may, for example, utilize social engineering techniques to accomplish a malicious goal, such as stealing confidential information or making some resources unavailable using a Denial-of-Service attack. Much research exists on dealing with external threats, and many security defences have been proposed, including host-based access controls, Intrusion Detection Systems (IDSs), and access control mechanisms [41, 42, 43]. On the other hand, an insider is a person who belongs to an organization and is authorized to access a range of its data and services. We are particularly interested in insider attacks as the nature of these attacks make them challenging to detect. The next section reviews how the understanding and definition of insider has evolved in the literature and provides the definition used in this work.

2.1.2 Defining Insiders

Several definitions can be found in the literature for an insider [44, 45, 46]; however, there is no consensus for a single definition [47]. In the 2008 paper, "Defining the Insider Threat", Bishop and Gates considered three definitions of insiders [44]. The first definition was from a RAND report [48] that defines an insider to be *"an already trusted person with access to sensitive information and information systems"*. The second definition was also from the same RAND report, which defines an insider to be *"someone with access, privilege, or knowledge of information systems and services"*.

The third definition, originating from [49], defines an insider to be *"anyone operating inside the security perimeter"*. In the first definition, a person needs to be trusted in order to be called an insider, however, in the second definition, a person having knowledge of the system and services is also considered as an insider while the third definition considers everyone within the security perimeter to be an insider.

The first three definitions are regarded as binary definitions because if the person satisfies one of these definitions, then that person is called an insider, otherwise, the person is not an insider. Bishop and Gates [44] also provided a fourth, non-binary, definition for an insider. The non-binary notion of an insider is based upon a measure of the damage that an organization would suffer if entities such as resources, important documents, e-mails, source code, etc. are compromised or leaked. Each entity is assigned an impact value that specifies this measure of damage. Entities with the same impact value are grouped in protection domain groups. These protection domain groups are then paired with groups of users having access to entities in protection domain groups. Users having access to protection domain groups with the highest impact-value pose the highest risk of insider threat. This model provides a spectrum on which the degree to which an insider poses a threat can be identified. We believe that such a model is useful in developing more fine-grained security mechanisms by taking into account the threat-level that an insider poses. This proposed model for insider threat that Bishop and Gates presented is useful in understanding how threats may be traced and aggregated through a system. However, the model does not define what is meant by an insider.

In a 2008 cross-disciplinary workshop on *"Countering Insider Threats"* [45], a more specific definition was proposed. In this workshop, the insider was defined as *"a person that has been legitimately empowered with the right to access, represent, or decide about one or more assets of the organization's structure"*. This dissertation adopts this definition, following its growing support.

A famously reported case of an insider attack was of an employee at an office that issues driving licences. The employee exploited access privileges to issue fraudulent licenses [46]. It has been reported that insider attacks can be unintentional as well. For instance, the breach resulted from the carelessness of an employee [50].

Once we've defined who an insider, we look at the definition of an insider threat is defined. The definition for insider threat that has consensus on it is put forward by Predd et. al. in [51] that defines insider threat as *"[...] an insider's action that puts an organization or its resources at risk"* [51].

2.1.2.1 Masqueraders and Masquerade Attacks

It is worth mentioning that this of an insider does not cover the *masqueraders*. A masquerader is defined as an attacker who has gained (steals) the credentials of an employee (insider) of a contemporary organization and subsequently, uses those credentials to maliciously access organization resources (includes databases) by impersonating as a legitimate employee of that organization. A masquerade attack can take one of the two following forms (i) - a masquerader is an insider who gains control of credentials of another employee of the same organization having different a privilege level than that being held by the masquerader, (ii) - the masquerader is an outside attacker who somehow gains control of legitimate employee's credentials. A distinction, in-terms knowledge base, can be drawn between an *internal masquerader* and *external masquerader*. Intuitively, an insider masquerader knows more about the organization as compared to an external masquerader though this is not necessarily always be the case. Additionally, an internal masquerader can mimic behaviour similar to the behaviour of other employees of the organization, whereas an external masquerader's behaviour is likely to manifest differently because of the lack of knowledge about employee behaviours. In this dissertation, internal and external masqueraders are treated as one. Figure 2.1 a depiction of insiders, outsiders, and internal and external masquer-

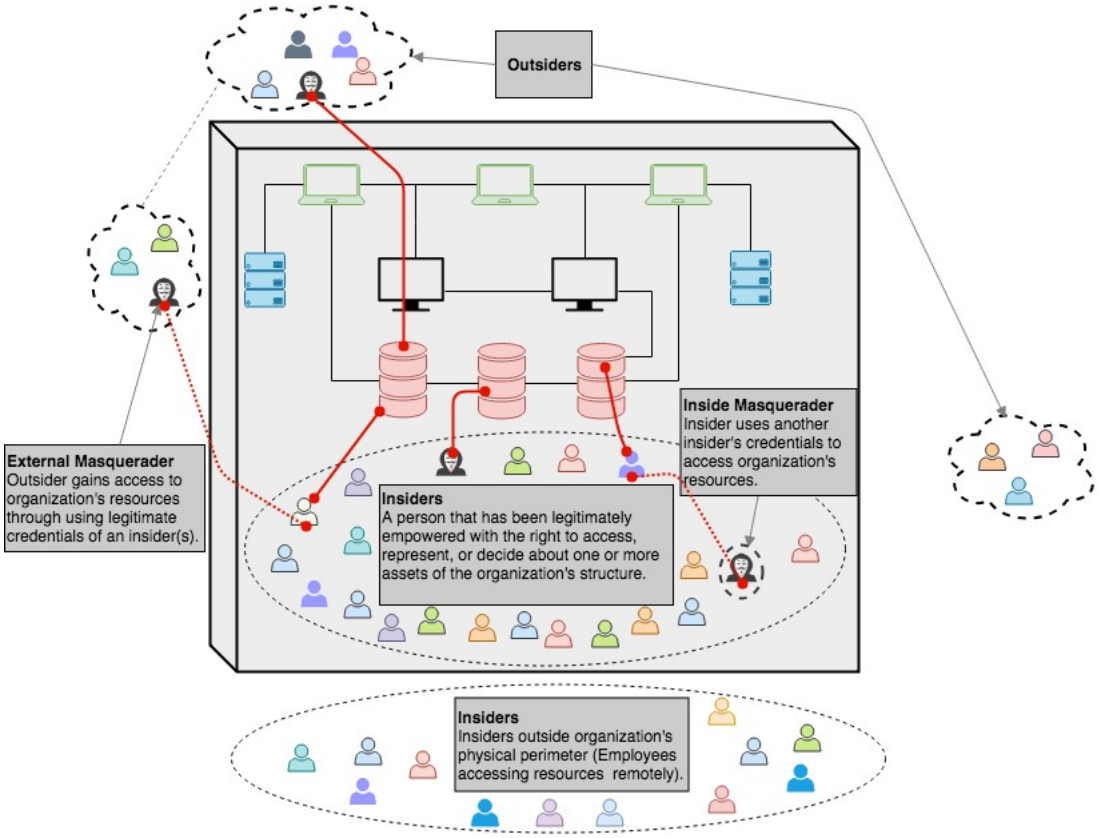


Figure 2.1: Figure depicting insiders, outsiders, internal and external masqueraders in organizational settings.

aders in an organizational setting.

2.1.3 The Impact of an Insider Attack

A 2015 report titled *Insider Threat Report* by Vormetric [5] reported that globally 89% of the respondent organizations are at risk of insider attack and that among these respondent organizations 34% of them felt extremely vulnerable to this kind of attack. Of the respondent organizations, 56% plan to increase their spending on tackling the challenge of insider threat. The 2015 *Cost of Cyber Crime: Global Report* from the Ponemon Institute [52] reported that insiders cause the costliest cyber-crimes. Information Systems Audit and Control Association (ISACA), reported that, globally, insider threat was among the top three threats for 2016 [53]. In the context of healthcare,

insiders have reported as the cause of most health-care data breaches [54]. For example, a health-care data security report from IBM Managed Security Services reported that insiders were responsible for 68% of all network attacks targeting health-care data in 2016 [55]. Insider attacks are on the rise, as reported in a 2015 survey report [6] that internal factors caused 43% of data breaches as compared to 2004 where it was reported in [56] that insiders caused 29% of the crimes.

Nonetheless, the reporting rate of insider attacks remains very low for a variety of reasons compared to the actual number of instances, including inconsequential impact or lack of evidence [56]. Another reason is that organizations are sometimes hesitant to report insider attacks due to loss of reputation and liability; however, legislations are now in place directing organizations to report data breaches such as the recently passed Privacy Amendment (Notifiable Data Breaches) Act 2016 in Australia [57] and the EU's General Data Protection Regulation that came into effect from 2018 [15].

2.2 Anomaly Detection in Systems

Originally, IDSs were designed to detect network intrusions and can be classified into misuse detection systems and anomaly detection systems [58]. Misuse detection systems look for existing misuse patterns and are limited to detect previously known attacks [58, 59, 60]. In general, misuse detection systems have also been referred to as *signature-based systems* or *knowledge-based intrusion detection systems* in the literature [61, 62, 63]. The majority of commercial intrusion detection systems are misuse detection systems [64, 65, 66]. Misuse detection systems are circumvented by sophisticated attackers targeting an organization as these systems only detect known attacks [67]. A misuse detection system detects attacks by comparing the audit trails with the existing attack signatures. It provides a guarantee that the known attack is detected, but it cannot detect an unknown attack. It is difficult to determine attack

signatures for all the variants of a particular attack as different ways exist to exploit vulnerability or weakness.

In contrast to misuse detection systems, anomaly detection (also known as *behavioural-based*) systems look for a deviation from normative behaviour. In principle, anomaly detection systems have the potential to detect zero-day attacks – attacks for which there is not a known predefined pattern. However, in practice, it is a challenge to model normative behaviour accurately. Existing work has generally focused on identifying anomalous system operations [68], malicious network events [69] or malicious application system events [70]. The anomaly detection systems can be distinguished on the basis of the way in which normative behaviour is modelled, that is, either the system learns the normative behaviour by automatically mining the past behaviours (learning-based anomaly detection systems) or the normative behaviour (specification-based anomaly detection systems) is specified manually [71, 72].

It has been reported that the attacker can evade the anomaly detection system by carefully mimicking normative behaviour while exploiting a vulnerability. Such attacks are known as *mimicry attacks* [73, 74, 75, 76, 77].

The effectiveness of IDSs is evident in domains like computer networking, operating systems, and industrial control systems thus making them a favourable choice to be deployed to protect databases against intrusions [28, 30, 29, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87]. However, IDSs deployed to protect systems in the above-mentioned domains are not adequate for databases; therefore, IDSs tailored to databases are desirable. Such a tailored IDS for DBMS is known as *Database Intrusion Detection System (DIDS)*. This dissertation focuses on the detection of malicious queries that are made by an insider to a DBMS. The literature has shown that intrusion detection systems tailored to databases are effective in the detection of these malicious queries made by an insider [3, 11, 10, 12]. The following sections of this chapter review DIDS research and proposes a taxonomy of IDS in the context of DBMS.

An anomaly-based detection system is further classified into learning-based or specification-based. While remaining within the scope of detecting malicious access to DBMS, the literature lacks any specification-based detection system. A naïve way to design a specification-based detection system would be to list all the legitimate SQL queries. However, it is impractical to a priori specify every potentially legitimate query. The development of a complete specification in the case of DBMS is unattainable, essentially for the inherent flexibility of SQL, that is, a SQL statement can be written in different ways to query the information.In our opinion, the notion of the specification-based detection system is immaterial in the context of databases intrusion detection because of its impracticality. In the literature in general, within the context of DIDS research, the anomaly-based detection systems imply that it is a learning-based system. In literature and commercially, the DIDS solutions are routinely referred to as Data Loss Prevention (DLP) solutions.

2.3 A Taxonomy for DBMS Anomaly Detection

This section introduces a taxonomy of methods that detect anomalous access to a DBMS. Anomaly-based DIDS research has remained a centre of focus of the research community, while less attention is being paid on misuse (or signature)-based DIDSs. The proposed taxonomy is shown in Figure 2.2 that broadly categorizes IDSs to detect anomalous access in a DBMS. An aspect to keep in consideration while performing classification of anomaly-based DIDS is the set of features used to model normative behaviour of a user, for example, time of access, attributes in projection clause, relations/tables queried etc. Section 2.3.2 and its following sections discuss these classifications. To address these classifications, we first present the prevalent architecture of anomaly-based database intrusion detection systems in the following section.

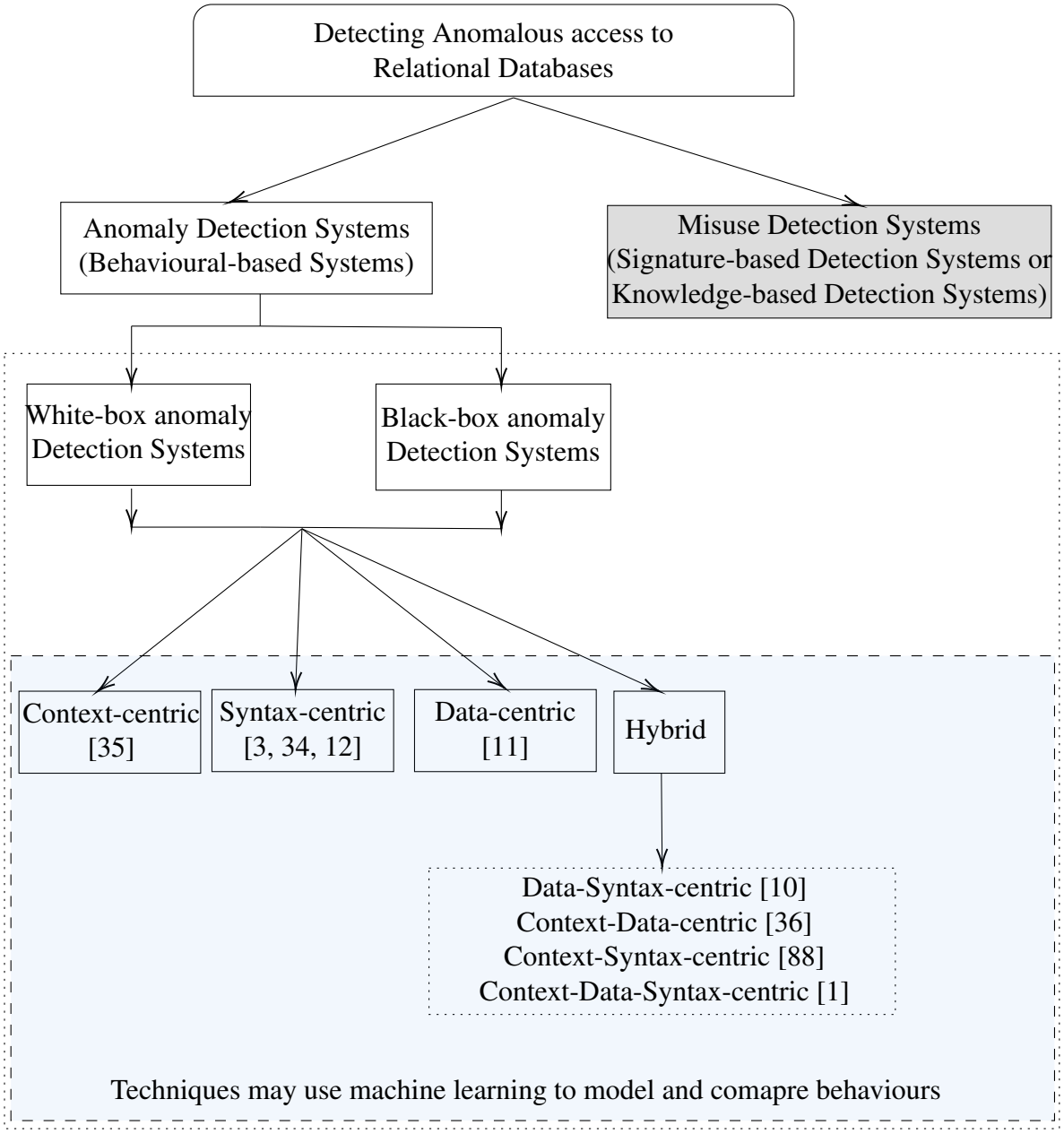


Figure 2.2: Taxonomy of anomalous DBMS-access detection systems.

2.3.1 Prevalent Architecture of Anomaly-based Database Intrusion Detection Systems

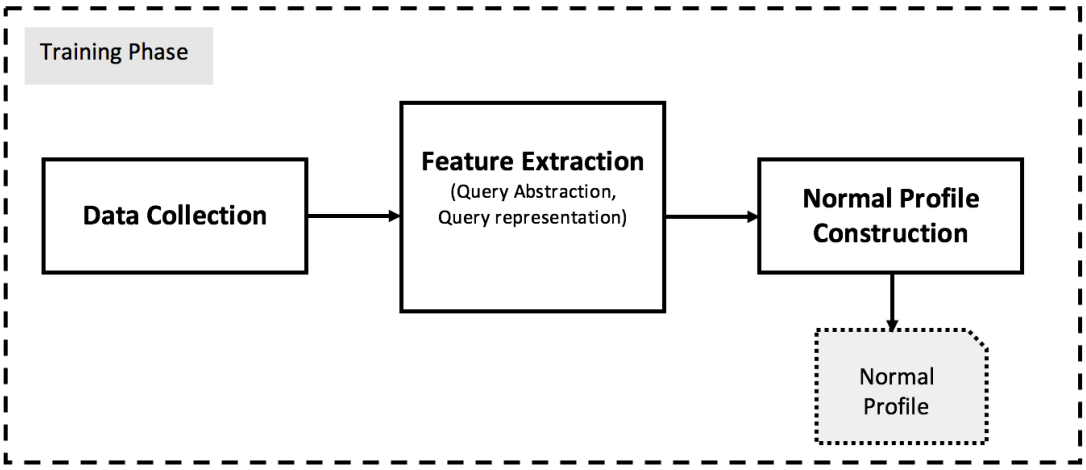


Figure 2.3: Training phase of an anomaly detection system.

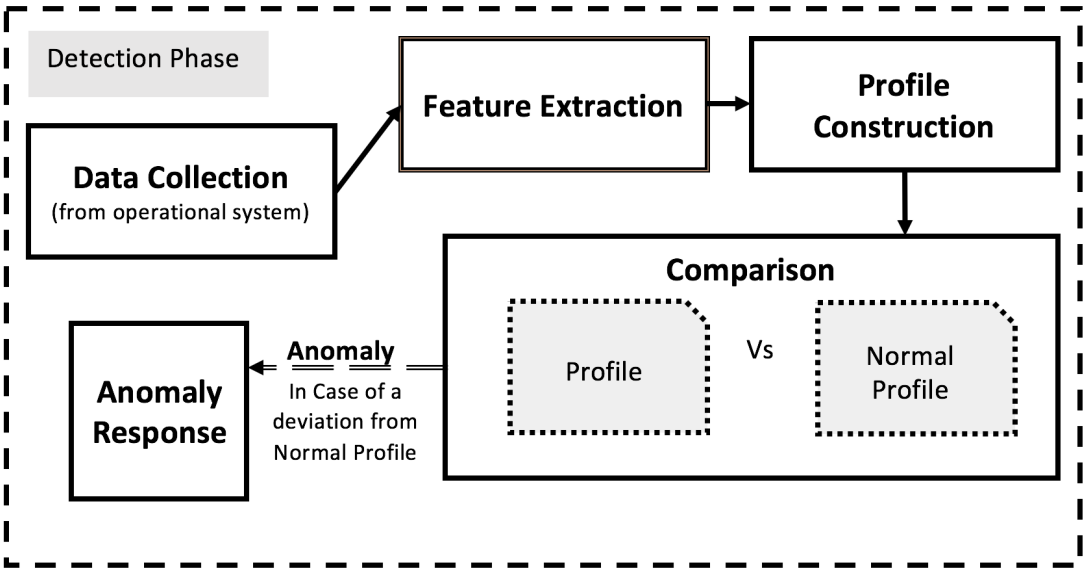


Figure 2.4: Detection phase of an anomaly detection system.

Figures 2.3 and 2.4 depict the prevalent architecture of an anomaly-based database intrusion detection system. The architecture involves two phases: a training phase and a detection phase. A profile of normative behaviour is constructed during the training phase and is compared against the run-time profile that is constructed in the detection phase. The future deviations of the run-time profile from normative profiles

are labelled as anomalies and necessitate attention. The architecture consists of two fundamental components: feature extraction (abstraction) and profile constructor. In feature extraction, the features required to construct the profile are extracted. The profile construction component is the technique to construct the profile using selected features.

Profiling "normative behaviour" is non-trivial as many features about queries can be taken into consideration while constructing profiles, and identifying significant determinant features is a challenge. The classification based on features and the techniques to construct profiles is discussed in the following next sections.

2.3.2 Feature Classification

Anomaly-based intrusion detection techniques for detecting malicious access to databases can be further distinguished based on what features they extract from a DBMS audit log of SQL queries to model behaviours. The modelled behaviour is represented as a behavioural profile [8, 3]. These features can be syntax-centric, context-centric, and data-centric, which is sometimes is referred to as result-centric in the literature [3, 10, 11].

Techniques using syntax-centric features construct behavioural profiles by using syntax features of the SQL query included but are not limited to the attributes in a projection clause, the relations queried, the attributes in selection clause, and/or the type of SQL command [3]. Techniques using data-centric or result-centric features construct behavioural profiles using data returned in response to an SQL query or any other statistical measurement on returned data, for instance, the minimum and maximum value in case of numeric data. For example, one could use the amount of information (the percentage of data returned) returned in response to a query or the returned values of attributes to model behaviour or a user [11]. The context-centric techniques construct behavioural profiles use contextual features. Contextual features are associated with

the context of the query, for example, the time at which the query was made, the user ID of the person making the query, or the number of queries made in a specified time period, etc [35]. A combination of the context, syntax and data-centric features, can be used while modelling normative behaviours. One such anomaly detection technique that uses syntax and data-centric features is proposed in [10].

An anomaly-based detection system building behavioural profiles that are not understandable by humans in a meaningful way is known as a *Black-Box* anomaly detection system. On the other hand, an anomaly-based detection system building behavioural profiles that are understandable by humans are known as *White-Box* anomaly detection system. Understandability implies that the actual root cause of the anomaly can be identified by the administrators (security officer, database administrator, etc.) when they inspect the anomaly. Intuitively, white-box approaches have the potential to help explain anomalies.

2.3.3 SQL Query Abstraction

Syntax-centric implies using only factors associated with the syntax of an SQL statement. A question that arises is how much of the SQL statement can be considered while constructing a behavioural model, that is, should one consider the entire statement or some parts of the statement – the challenge of selecting appropriate SQL query abstraction. Abstraction is a tuple representation of an SQL statement and consists of query features like relation name, attribute names, the amount of returned data or any statistics on the returned data. A number of techniques [3, 9, 10, 89, 90, 91, 92] have been proposed that transform the syntax of an SQL statement into a more abstract fingerprint that can be used for comparing queries. SQL query abstractions are also referred to as SQL query fingerprints [89, 90], SQL query signatures [3] or SQL query skeletons in the literature [9]. In this dissertation, various specializations of SQL query abstraction is used and described in their respective chapters. The query abstractions

used by existing anomaly-based detection approaches are also described in this chapter.

Other than DIDS research, the use of abstraction in practice has also been studied by audit log summarization research as typically audit logs encompass a large number of queries and methods to summarize logs in a meaningful way are sought [93]. For instance, it has been reported in a recent study that within the time period of 19 hours, approximately 17 million SQL queries were made in a major US bank [9].

2.3.4 Syntax-centric Features-based Techniques

Several anomaly detection approaches to detect malicious accesses to DBMS use syntax-centric features of the SQL queries to construct behavioural profiles [3, 34, 12]. For instance, the approach *DetAnom* [3] detects malicious DBMS-accesses by application programs. SQL queries executed by an application program are represented in the form of SQL query abstractions to generate a normative profile of an application. In [3], a SQL query abstraction consists of the following elements, (c, t, r, q, n), where c is SQL command type, for instance, `SELECT`. Attribute t is the list of attribute identifiers projected in the query and are relative to the relation to which they belong. Attribute r is the list of relation identifiers. Attribute q is the list of attribute identifiers in the `WHERE` clause and n is the number of predicates in the `WHERE` clause. For example, consider the following SQL query:

```
SELECT bankdb.acc_number, bankdbacc.current_amount,
FROM bankdbacc, bankdb,
WHERE bankdb.acc_number=7594
AND bankdb.acc_number = bankdbacc.account;
```

The corresponding identifiers for relations and the attributes are shown in Table 2.1. This query is then represented as follows:
{<1> <100:10, 200:20> <100, 200> <100:10, 200:30> <2> }.

Table 2.1: Identifiers example for the query abstraction approach proposed in [3].

Command Type	**Representation**
SELECT	1
UPDATE	2
INSERT	3
DELETE	4

Relation Name	**Relation Identifier**
bankdb	100
bankdbacc	200

Attribute Name	**Attribute Identifier**
acc_number	10
current_amount	20
account	30

The generation of a query abstraction is followed by the exploration of all execution paths of the application program accomplished by Concolic execution (Concolic testing) which is a technique for program analysis [94, 95, 96]. Once all the paths are explored, the branching condition is then paired with the SQL query that falls under that branching condition that was discovered in the process of concolic testing. This *(branch condition, SQL query)* pair is referred to as a query record.

In the detection phase of DetAnom, a query is intercepted, and its branch condition is matched with the branch condition in the corresponding a query record. If the branch condition is satisfied, then matching of query abstraction is carried out. In case of a mismatch, the query is said to be malicious. DetAnom works only for application programs where the SQL queries are embedded in program code, and it is not extendible to capture users behaviours manifested in a variety of SQL queries made to the database.

Another early approach based on syntax-centric features is presented in [12]. The approach uses three query abstractions each having different level of granularities, coarse triplet, medium-grain triplet, fine triplet.

A Naïve Bayes classifier was used to predict a role for the SQL query. If the predicted role is not the same as from the one SQL query was originated then an alarm is raised.

This approach has several limitations including, considering each query in isolation and it does not consider the sequence of queries, Second, the approach constructed profiles of roles while ignoring the case where one user can belong to multiple roles.

Syntax-centric approaches, in general, are useful in detection masquerader attacks as well as SQL injection attacks both these attacks lead to structural changes in SQL statement.

2.3.5 Data (Result)-centric Features-based Techniques

Little research has been reported on using data-centric features as the basis for anomaly detection in the context of DIDS. Data-centric features include the amount of data returned in response to a query or returned values of attributes or any other statics performed on the returned set of attribute values.

In [11], it is argued that syntax-centric features of a query alone are a poor discriminator of intent. Syntactically different queries can potentially give the same result while syntactically similar queries can potentially yield different results. Therefore, a user can craft a legitimate SQL query to retrieve results from the database which the user is authorized to retrieve, yet a purely syntax-centric anomaly-based detection system might label this as anomalous behaviour.

In the approach proposed in [11], user profiles are clusters that are specified in terms of an *S-Vector* that provides a statistical summary of the results (tuples/rows). In the detection phase, clustering algorithms were adopted, that are, as supervised learning methods, Euclidean k-means clustering, Support Vector Machines (SVM), Decision Tree Classifier, and Naïve Bayes, and as unsupervised methods, Cluster-Based Outlier Detection (based on Euclidean distance clustering) and Attrib-Deviation, using L_∞-norm. If a query belongs to the cluster, then it is considered normal else it is regarded as anomalous. The presented approach is suitable for the detection of a query in isolation

and does not take a sequence of queries into account.

Data-centric approaches are capable of detecting sophisticated attacks as well. For instance, data harvesting attacks involving retrieval of a large amount of data, therefore, exceeding what is retrieved by a legitimate user.

2.3.6 Context-centric Features-based Techniques

Few purely context-centric approaches are reported in the literature. One such context centric approach is presented in [35] in which contextual features are considered in modelling user behaviours. The approach took the deployment of anomaly-based IDS in the medical sector as its use-case and studied the Break-The-Glass (BTG) procedure which is a procedure that breaks the traditional access control mechanism and enable access of patients data in case of emergency to employees of different departments. In this approach in [35], users who supposed to behave similarly are divided into groups and profiles are constructed for groups. The feature space comprised of contextual features like access type, time, division, date. Profiles were represented as the sequence of histograms and were constructed using the concept of *Bins*. Bins represent the frequency of features. In the detection phase, the distance between the histogram of a user and the existing profile is measured, and a larger distance is an indication as an anomaly. The approach represented user and group profile in terms of a sequence of histograms of features that can easily be interpreted by a concerned individual like a security officer. Therefore, the approach in [35] can be classified as a white-box anomaly detection approach.

Context-centric approaches typically increase the detection effectiveness of an ID approach when combined with syntax or data-centric approach.

2.3.7 Hybrid Techniques

An example of an approach using data and syntax-centric features to construct behavioural profiles is presented in [10]. Machine learning techniques, in particular, Naïve Bayes classifiers and multi-labelling classifiers, were also deployed in the profile generation process. User profiles are built in the training phase from logs containing user activities.

The approach transforms an SQL query into an SQL query abstraction called a quadruplet. A quadruplet $QT(c, P_R, P_A, S_R)$ is composed of data-centric and syntax-centric features including the command type c; the list of relations accessed by the query P_R; the list of attributes accessed by query relative to the relation P_A; and the amount of selected information from the relation S_R. This hybrid approach is demonstrated in two settings that is role-based anomaly detection and unsupervised anomaly detection. In the detection phase the role of a querier was predicted using a Naïve Bayes classifier. Multi-labelling classification was used in case of an overlap of roles that results in more than one role. If the predicted role is different from the actual role then the query is labelled as anomalous. In the unsupervised settings, the COBWEB [97, 98] clustering algorithm was selected. The query is treated as anomalous if a query made by a user falls into a cluster that does not contain any query made by this user. This approach is promising for the detection of a single malicious query in isolation; however, it ignores sequences of queries while modelling behaviour.

The approach presented in [36] uses context-centric and data-centric features. Normative profiles are constructed by discovering association rules between context-centric features and data-centric features using frequent item-set mining [99]. The basic idea of the approach is to tie the results retrieved by the SQL query with the context in which they were retrieved. For instance, a transaction made in London in the morning typically retrieves records for employees of the human resource department. Therefore, human resource department employees records are tied with the context that they

are normally retrieved in the morning from London. In the detection phase, for any incoming query, context-centric features were extracted, and rules conforming to these features are matched, and then the result of the query is matched with the results associated with the retrieved rules. A drawback of this approach is that large databases result in large profiles. Additionally, this approach is too restrictive and less likely to be scalable. Another drawback of this approach is in general the drawback of context-centric approaches that is the context can be easily mimicked.

The approach in [88], also employed context and syntax-centric features to model behaviours and forms some assumptions for instance that every department in an organization has a unique IP space, employees work in shifts (there are three shifts in a day). The features collected for modelling include employee ID, role ID, time, IP address, access type (direct or through an application). The approach also records the SQL query associated with the contextual features. The profile consists of the probability of each feature's occurrence observed for every user. In the detection phase, a new transaction is compared against the constructed profiles to check the closest issuer (user) of that transaction. The transaction is labelled as an anomaly in case the issuer of the transaction is different from the one computed. This approach also considered role hierarchy, meaning, if role $\ddot{r}_1$ is above role $\ddot{r}_2$ in the hierarchy, then the access privileges of role $\ddot{r}_2$ is a subset of role $\ddot{r}_1$. For instance, if a query made by $\ddot{r}_1$ is labelled as malicious, but the same query is legitimate for $\ddot{r}_2$ then this is not considered as a malicious query. The focus of this approach is to augment Role-Based Access Control (RBAC) to ensure that the query is made only by the authorized users. Similar to approaches discussed above, this approach also detects only single malicious DBMS transaction where later in the dissertation it is argued that a single query may be legitimate, however, a group of them made together might result in malicious or illegitimate action.

In [1] contextual, data and syntax-centric features are considered in modelling. The

Features per User/IPAddress		
F#	**FeatureName**	**Description**
F_1	#ConsFailedLoginAttempts	The number of consecutive failed database login attempts by a UserID or from an IPAddress (accumulated or in a given timespan)
F_2	#SimultSQLSessions	The number of active simultaneous database connections
F_3	#UnauthorAccessAttempts	The number of consecutive user requests to execute an unauthorized actions (*e.g.* request to modify data when the database is read-only, or requesting to query data to which does not have access privileges)
Features per User/IPAddress per Command		
F_4	CPUTime	CPU time spent by the DBMS to process the command
F_5	ResponseSize	Size (in bytes) of the result of the command's execution
F_6, F_7	#ResponseLines, #ResponseColumns	Nr. of lines and columns in the result of the command's execution
F_8, F_9	#ProcessedRows, #ProcessedColumns	Nr. of accessed rows and columns for processing the command
F_{10}	CommandLength	Number of characters
F_{11}	#GroupBy	Number of GROUP BY columns
F_{12}	#Union	Number of UNION clauses
F_{13}...F_{17}	#Sum, #Max, #Min, #Avg, #Count	Nr. of SUM, MAX, MIN, AVG and COUNT functions
F_{18}, F_{19}	#And, #Or	Nr. of AND and OR operators in the command's WHERE clause(s)
F_{20}	#LiteralValues	Nr. of literal values in the command's WHERE clause(s)
Features per User/IPAddress per Session		
F_{21}	#GroupBy	Number of GROUPBY columns in all SELECT statements, p/ session
F_{22}	#Union	Number of UNION clauses in all SELECT statements, per session
F_{23}...F_{27}	#Sum, #Max, #Min, #Avg, #Count	Nr. of appearances of SUM, MAX, MIN, AVG and COUNT functions in all commands, per session
F_{28}	TimeBetwCommands	Time period (in seconds) between exec. of commands, per session
F_{29}	#SimultaneousCommands	Number of commands simultaneously executing, per session

Figure 2.5: List of features considered. Figure cropped from *Securing Data Warehouses from Web-Based Intrusions* by Santos et al. [1]

database intrusion detection approach in [1] is tailored for Data Warehouses in which applications are enabled to access Data Warehouses via the web. The profiles are constructed by considering various features, as shown in Figure 2.5 and are represented in terms of the probabilistic distribution of each feature for each user and for the entire population. In the detection phase for this approach, testing is done to match features distribution with the distributions obtained in the training phase using statistical hypothesis tests like Chi-square, Shapiro-Wilk, and Kolmogorov-Smirnov tests [1]. In case of a non-conformity, the activity involving that feature is labelled as an anomaly. The approach is focused on web-based malicious access to Data Warehouses though insiders remain unaddressed. Table 2.2 shows a consolidated presentation of discussed and well-known approaches proposed in the literature.

Table 2.2: An overview of the characteristics discussed and well-known approaches proposed in the literature. ⊛, ⊙, and ⏢ represents syntax, data(result), and context-centric features respectively.

Approach	Features	White-box / Black-box	Single Query detection	Sequence of Queries detection	Profiles	Detection Style
Hussain et al. [3]	⊛	■	✔	✖	Tuples of SQL-branch condition	Tuple matching
Mathew et al. [11]	⊙	■	✔	✖	Clusters / Classes of S-Vector	**Supervised:** Euclidean *k*-mean SVM, Decision Tree Classifier Naïve Bayes **Unsupervised:** Cluster-Based Outlier Detection Attrib-Deviation
Alizadeh et al. [35]	⏢	□	✔	✖	Histograms	Distance b/w histograms
Sallam et al. [10]	⊛ ⊙	■	✔	✖	Classes / Clusters of Quadruplet	**Supervised:** Naïve Bayes. Multi-labelling classifier **Unsupervised:** COBWEB
Gafny et al. [36]	⊙ ⏢	■	✔	✖	Association Rules	FIM, rule matching
Kamra et al. [12]	⊛	■	✔	✖	quiplets,	Naïve Bayes
Wu et al. [88]	⊛ ⏢	■	✔	✖	Probabilities of Feature's Observed Values	Naïve Bayes
Santos et al. [1]	⊛ ⊙ ⏢	■	✔	✖	Probability Distribution	Distribution Matching (statistical hypothesis tests i.e. Chi-square, Shapiro-Wilk, Kolmogorov-Smirnov tests)

2.4 Conclusions

Database intrusion detection, specifically behavioural-based database intrusion detection approaches have seen to be effective in detecting insider attacks. Two significant aspects in the design of behavioural-based approaches are what level of features (and in the case of SQL statement then what level of abstraction) are selected for modelling behaviours and the technique (algorithm) selected for constructing profiles. The literature contains approaches using syntax-centric, data-centric, or context-centric features or a combination of these features. To construct profiles, machine learning approach like classification, clustering, as well as distance functions, rule matching algorithms are adopted. In the existing literature, pure data-centric, and context-centric approaches are not prevalent. Some of the approaches are tailored for specific applications, i.e., data warehouses; second, these approaches do not allow for regularly updating normative profiles.

The majority of these approaches reported in the literature are focused on the detection of a single malicious SQL query in different settings by considering only single SQL query in modelling behaviours [3, 11, 35, 10, 36, 1, 88]. The drawback of such approaches is that they are effective in detecting only those malicious accesses by insiders that are based on an SQL query in isolation. Little attention has been paid on the detection of malicious queries sequences. As a single query might not be malicious, but a sequence of SQL queries might be an indication of malicious activity. For instance, consider a banking scenario where a query made to check customer's account balance is normal and is usually followed by queries pertaining to other banking-related actions such as withdraws or transfers. In the case, where an insider browses through the bank's database by querying each customer's account balance then each query in isolation is normal; however, the sequence of queries made by the insider results in malicious activity. Therefore, such approaches that can detect the malicious sequence of SQL queries are desirable.

Chapter 3

Definitions of Privacy

"Right to privacy is really important. You pull that brick out and another and pretty soon the house falls."

Tim Cook (1960 - Present)

3.1 Introduction

This dissertation involves the domains of detecting anomalous access to DBMS and formal definitions of privacy. The state of the art research on detecting anomalous access to databases was examined in the previous Chapter 2. This Chapter reviews the formal privacy definitions from the literature. Section 3.2 reviews various aspects of privacy; for instance, how privacy is understood. Section 3.3 examines well-known formal privacy definitions. Section 3.4 summarizes the Chapter and presents the key observations gathered from the literature. Conclusions are drawn in Section 3.5.

3.2 Countless Shades of Privacy

Privacy is a challenging concept to define because of its culturally subjective basis. A classic paper [100] considers the philosophical dimensions of privacy in three perspectives: *"i. a claim, entitlement or right of an individual to determine what information about himself may be communicated to others [101, 102]. ii. measure of the control an individual has over: (a) information about himself; (b) intimacies of personal identity; or (c) who has sensory access to himself [103, 104, 105, 106]. iii. state or condition of limited access to a person [107, 108]."* This right to privacy is a globally recognised right as stated in Article 12 of the UN's Universal Declaration of Human Rights that states [109],

> "*No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.*"

Taking the privacy concerns of the modern-day digital world, legislators have provided legal cover to the public to ensure their right to privacy. Dominant examples of such legislation include the Health Insurance Portability and Accountability Act (HIPAA, 1996) [110], the EU's General Data Protection Regulation [111], the OECD privacy framework [112]. 58% of total countries in the world have data protection and privacy legislation in place while 10% have draft legislation and 21% of the countries still have no legislation in place (there is no data for the remaining 12%) [113].

Contemporary organizations collect large volumes of data over which analytics are routinely carried out for various purposes, including data-driven marketing and informed decision making. Organizations often delegate this task to third parties specializing in data analytics. The collection, storage, processing, and sharing of personal data raises dimensions privacy concerns. The work of Tore Dalenius in [114], points out the prob-

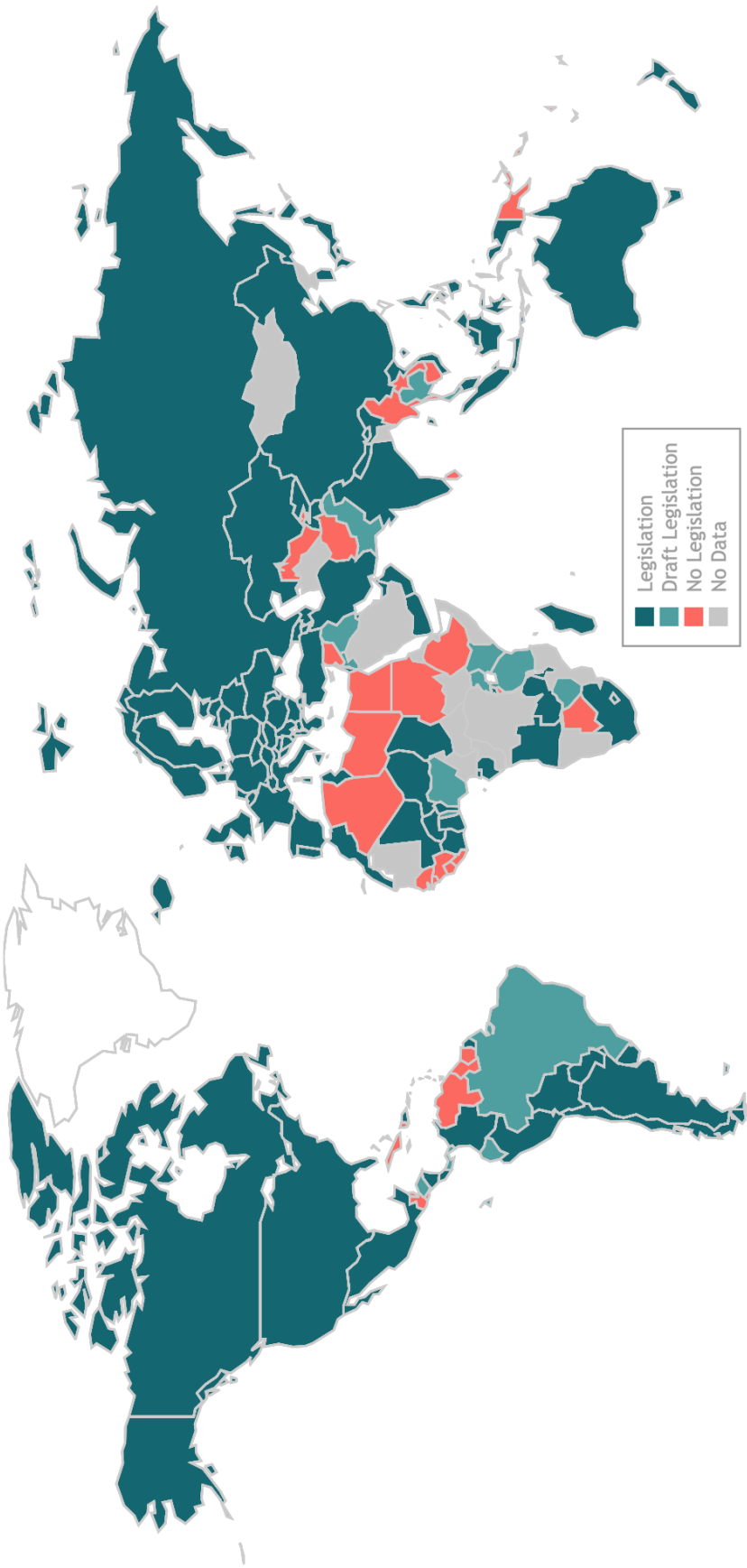


Figure 3.1: The state of Data protection and privacy legislation worldwide [2].

lem of publishing the data of populations without risking individual's privacy. The research in this domain started to gain momentum in the late 1970s and 1980s, especially in the context of publishing census data. Recently privacy research is no longer limited to census data, as emerging new technologies have brought privacy concerns along, for example, the privacy concerns in social networks [115, 116, 117], internet of things [118, 119, 120] and e-commerce [121, 122, 123].

The relational data model is the most widely used data model for storage and processing of data [124, 125]; for this reason, the relational data model in DBMS (Relational DBMS – RDBMS) is considered in this work. To ensure the privacy of individuals, the data is anonymized. The anonymized version of data is typically made available in two settings within the RDBMS framework, that are, non-interactive and interactive query setting. Interactive query setting allows dynamic queries, while in a non-interactive setting an anonymized version of the entire database is made available. In general, data is considered anonymized if it complies with a formal privacy definition. Several of the formal definitions of privacy have been proposed in the literature that are reviewed in the next Section 3.3. When an anonymized version of data is made available, then in that context an adversary aims to strive for identifying individuals and disclosing their sensitive attributes in the database.

The next section looks further into the formal definitions of privacy research proposed in the literature to achieve anonymization in the context of databases.

3.3 Formal Privacy Definitions

Several forms of privacy have been formalized in the literature. The two mainstream definitions of privacy are k-anonymity [3] and differential privacy [126]. k-anonymity serves as the foundation for several privacy definitions including l-diversity [127], t-closeness [25], (α, k)-anonymity [128].

The syntactic definitions of privacy, like k-anonymity, l-diversity, t-closeness, are typically for one-time release of data. The table, specifically microdata table, that is required to be released is transformed such that the transformed table complies with chosen privacy definition. Then the transformed version of the table is later released. On the other hand, differential privacy can be used in interactive settings, where a user sends queries and receives responses to those queries. In general, differential privacy works for statistical databases, and the differential privacy criteria is achieved by adding noise in the query responses. Statistical databases deal with aggregates instead of microdata tables. Richer analytics requires a release of microdata tables in interactive query settings which is a challenge as multiple releases may give rise to inferences being made by an adversary. Therefore, approaches to protect individuals privacy, while allowing the release of microdata tables in interactive query settings, having a foundation in formal privacy models are desirable.

This section reviews the well-known formal privacy definitions. The definitions are described within the framework of relational databases, and therefore, before reviewing the privacy definitions, the next section defines the key relational database terms used in this dissertation.

3.3.1 Key Relational Database Terms

In a relational model a relation instance is denoted as $\mathcal{T} = \{r_1, r_2, \ldots, r_n\}$ where r_i is a tuple of attribute values that represents a record. $\mathcal{T}$ is a subset of some larger population $\mathcal{U}$. Each tuple represents an individual from $\mathcal{U}$. Let the set of attributes be denoted by $Attr = \{attr_1, attr_2, \ldots, attr_n\}$, for example, the table shown in Figure 3.1, $Attr = \{\texttt{Name}, \texttt{City}, \texttt{Country}\}$. The value of attribute $attr_j$ for a tuple r_i is denoted as $r_i[attr_j]$, for example, $r_1[\texttt{Country}]$ = ('Spain'). $r_i[Attr]$ denotes the tuple $(r_i[attr_1], r_i[attr_2], \ldots, r_i[attr_n])$ which is the projection of r onto the attributes in $Attr$ for example $r_1[\texttt{Name}, \texttt{City}, \texttt{Country}]$ = ('Nicolau', 'Barcelona', 'Spain').

Table 3.1: An example relation $\mathcal{T}$.

r_i	Name	City	Country
r_1	Nicolau	Barcelona	Spain
r_2	Jordi	Barcelona	Spain
r_3	John	New York	USA
r_4	Sean	Cork	Ireland
r_5	Patrick	Dublin	Ireland
r_6	Matías	Berlin	Germany

3.3.2 *k*-anonymity based and Extensions

k-anonymity [24] can be considered among the first formal definitions of privacy. The following sections presents *k*-anonymity and its extensions.

3.3.2.1 *k*-anonymity

In the context of *k*-anonymity, attributes are classified in the following non-exclusive categories, *Identifiers*, *Quasi-Identifiers*, and *Sensitive attributes*. The classification is typically performed based on the risk of record re-identification using these attributes and the sensitivity of the information these attributes convey.

- **Identifier:** an identifier is defined as *"an attribute that refers to only a particular individual in the given population* $\mathcal{U}$*"*. Let the set of identifiers be denoted as $\mathcal{I} \in Attr$. An example of an identifier is the Personal Public Service Number (PPS Number) which can uniquely identify individuals in Ireland. Other examples include an individual's passport number, driving license number, and e-mail address.

- **Quasi-Identifier (QI):** quasi-identifiers by themselves do not uniquely identify individuals; however, when correlated with other available external data, an individual (or individuals) can be identified. A quasi-identifier is defined in [24, 20] as a *"set of non-sensitive attributes of a relation if linked with external data to then uniquely identify at least one individual in the population* $\mathcal{U}$*"*. Let the set of

quasi-identifiers be denoted as $QI \in Attr$. An example of quasi-identifier is the set of attributes `Zipcode`, `Date of Birth`, and `Gender`. For instance, the set of attributes `Zipcode`, `Date of Birth`, and `Gender` was used to re-identify governor of Massachusetts in [20]. The re-identification was performed by directly linking shared attributes in two datasets, i.e. voter rolls and insurance company datasets. It was reported that 87% of the US population could be identified by these three attributes [20].

- **Sensitive attribute:** sensitive attributes consist of sensitive person-specific information. This information includes salary, disability status, or disease. The set of sensitive attributes is denoted as $SenAttr \in Attr$. All possible values for the sensitive attribute is denoted as $SAV = \{sv_1, sv_2, \ldots, sv_n\}$.

k-anonymity is defined in [24] as follows, *"a relation $\mathcal{T}$ satisfies k-anonymity if and only if each tuple $r_i[QI] \in \mathcal{T}$ appears with at least k occurrence in $\mathcal{T}$"*.

k-anonymity [24] provides a degree of anonymity if the data for each person cannot be distinguished from k-1 individuals in a released dataset with respect to a set of quasi-identifiers. Given $QI \in Attr$ then two tuples r_i and r_j are quasi-identifier equivalent if $r_i[QI] = r_j[QI]$. The relation $\mathcal{T}$ can be divided into quasi-identifier equivalence classes. Let the set of all the equivalence classes in $\mathcal{T}$ be $\mathcal{E}$ where each equivalence class $e \in \mathcal{E}$ consists of all the rows that have the same values for each quasi-identifier.

Another way to define k-anonymity is that a relation $\mathcal{T}$ satisfies k-anonymity if the minimum equivalence class size is at least k in $\mathcal{T}$. Tables 3.2 and 3.3 show an original and a k-anonymized (3-anonymized) version of the microdata relations, respectively, where `Name` is an identifier, attributes `Age` & `Zipcode` are quasi-identifiers, and the attribute `Salary` is a sensitive attribute. In Table 3.3 the identifier `Name` is suppressed and is shown as * while quasi-identifiers `Age` & `Zipcode` are generalized – replacing with a semantically similar but less specific value.

In Table 3.3, records #1, #2, and #3, form an equivalence class, similarly records #4,

#5, and #6 as well as records #7, #8, and #9 also forms equivalence classes with respect to the quasi-identifiers {`Age` & `Zipcode`}. Originally, *k*-anonymity was proposed for a one-time release of data, meaning that the user is not enabled to query the DBMS interactively.

Though considered to be among the first privacy definitions, *k*-anonymity, has been widely applied in many domains to preserve privacy for examples Location-based services [129, 130, 131, 132, 133], ride-hailing services [134], and webmail auditing [135]. *k*-anonymity has been used along with cryptographic hashing to develop a protocol that provides a degree of anonymity while checking for passwords in a compromised databases [136].

Table 3.2: A relation with `Name` as an identifier, `Age` & `Zipcode` are quasi-identifiers and `Salary` as a sensitive attribute.

#	Name	Age	Zipcode	Salary
1	Kenneth	23	3134	77k
2	Hendry	37	3135	77k
3	John	34	3134	83k
4	Noemi	52	7290	65k
5	James	58	7291	77k
6	Amanda	55	7290	83k
7	Miyu	49	3134	65k
8	Vlad	43	3135	65k
9	Alex	46	3134	83k

Table 3.3: A 3-anonymized version of Table 3.2.

#	Name	Age	Zipcode	Salary
1	*	<45	313*	77k
2	*	<45	313*	77k
3	*	<45	313*	83k
4	*	≥50	729*	65k
5	*	≥50	729*	77k
6	*	≥50	729*	83k
7	*	4*	313*	65k
8	*	4*	313*	65k
9	*	4*	313*	83k

A weakness of *k*-anonymity is its susceptibility to the homogeneity attack and the

background knowledge attack [127]. If all the values for one of the sensitive attributes, within an equivalence class, are same then it results in a homogeneity attack, for instance, if the adversary knows an individual who is 33 years old and lives in the zipcode 3134 and has record is in an equivalence class where all the salaries are 77k then the adversary deduces that the individual's salary is 77k, though the table is k-anonymized. In [127], it was shown that if the adversary knows that Japanese patients are less likely to have heart-related diseases (background knowledge) enabled an adversary to predict the diagnosis for an individual.

3.3.2.2 *l*-diversity

Another well-known definition of privacy is l-diversity which is an extension of k-anonymity. The l-diversity principle in [127] states *"an equivalence class e is l-diverse if it contains at least 'well-represented' values for the sensitive attribute* $\in$ *SenAttr. A relation* $\mathcal{T}$ *is l-diverse if all the equivalence classes (q-block) are l-diverse"*. The term 'well-represented' is instantiated in three different ways as defined by the authors in [127] that are Distinct l-diversity, Entropy l-diversity, and Recursive *(c-l)*-diversity. The elementary form of l-diversity is distinct l-diversity. The remaining two instantiations are stronger instantiations and take the distribution of sensitive attribute values into account.

- Distinct l-diversity definition requires l distinct values in each equivalence class $e \in \mathcal{E}$.
- Entropy l-diversity is defined in [127] as *"A table l-diverse if for every equivalence class* $e \in \mathcal{E}$ the entropy of the distribution of sensitive values in each equivalence class is at least $log(l)$".
- Recursive (c, l)-diversity requires that the least frequent sensitive attribute values that do not appear rarely as well as most frequent sensitive attributes values do not appear too frequently in an equivalence class e.

Table 3.4 shows an *l*-diverse (3-diverse) version of Table 3.2, where each equivalence class has three distinct sensitive attribute values.

Table 3.4: A *l*-diverse (3-diverse) version of Table 3.2.

#	Name	age	Zipcode	Salary
7	*	<50	313*	65k
2	*	<50	313*	77k
3	*	<50	313*	83k
4	*	≥50	729*	65k
5	*	≥50	729*	77k
6	*	≥50	729*	83k
1	*	<50	313*	77k
8	*	<50	313*	65k
9	*	<50	313*	83k

In some scenarios it is difficult to achieve *l*-diversity. Let us say that a relation consisting of $20,000$ records has only one sensitive attribute, `Test Results`, where `Test Results` can take either negative or positive value. If 99.75% of the `Test Results` are positive, while 0.25% are negative, then there can be at most $20,000 * 0.25\% = 50$ equivalence classes.

l-diversity is susceptible to a *skewness attack* and *similarity attack* [25]. Consider a class with 9 positive values and 1 negative value, and another class with 9 negative values and 1 positive value, both classes are 2-diverse but present different privacy risks – skewness attack. *l*-diversity doesn't take the semantic-level closeness of the sensitive attribute values into account and thus leads to similarity attacks. For example, in a given equivalence class with the sensitive attribute `diagnosis`. Consider all the values for `diagnosis` in that given equivalence class are related to heart-related diseases. Therefore, an adversary knows someone in that given equivalence class then the adversary concludes that the individual has heart-related disease.

3.3.2.3 *t*-closeness

The *t*-closeness definition is a refinement of *l*-diversity. The authors in [25] define *t*-closeness as *"An equivalence class e is said to have t-closeness if the distance between the distribution of a sensitive attribute value in this class and the distribution of the sensitive attribute value in the whole relation $\mathcal{T}$ is no more than a threshold t."* As a distance metric, the Earth Mover's Distance can be used to measure the distance between two frequency distributions [25].

l-diversity treats sensitive attributes of all the equivalence classes in a similar manner without taking into account the global distribution of these sensitive attribute values in the relation. However, in the case of real-world datasets, sensitive attribute values might be skewed; therefore, it would be desirable to take into account the global distribution of sensitive attribute values. Limitations of *t*-closeness are discussed in [137], *t*-closeness degrades utility, is challenging to be achieved, and under certain scenarios, it is shown to be NP-hard [138].

3.3.2.4 (α, *k*)-anonymity

The (α, k)-anonymity is an enhanced version of k-anonymity. The (α, k)-anonymity defines that the frequency of sensitive attribute values α remains within the user-defined threshold in equivalence classes. The authors introduced an α-deassociation requirement in [128], if this requirement is satisfied along with k-anonymity, then it is said that (α, k)-anonymity is satisfied. Given a sensitive attribute value sv_i, a relation $\mathcal{T}$ is α-deassociated if the relative frequency of sv_i in every equivalence class of $\mathcal{E}$, that is $|(e, sv_i)|/|e|$, is no more than α. Where (e, sv_i) be the set of tuples in equivalence class e containing sv_i and α. A limitation of (α, k)-anonymity is that it may result in a high level of distortion if the values for sensitive attributes are skewed [139].

3.3.2.5 *m*-invariance

The *m*-invariance definition is also built upon *k*-anonymity. Most of the privacy definitions are suitable for the scenario where the relation is to be released once only. Second or subsequent releases of the relation (with an updated record or insertion of a new record) may lead to inferences.

The *m*-invariance definition [140] states that *"an anonymized relation $\mathcal{T}$ is m-unique if each equivalence class $e \in \mathcal{T}$ contains at least m set of records (or tuples) and all the records in e must have different sensitive attribute values. The sequence of published relations is said to be m-invariant if all the releases are m-unique, along with the condition that the set of sensitive attribute values for every e in $\mathcal{T}$ must remain same for subsequent releases of relation $\mathcal{T}$."*

This privacy definition captures dynamic republication of relations. However, the condition that values of the sensitive attribute should remain the same, as are in previous releases, is strong and limits its applicability.

3.3.2.6 (*k*, *e*)-anonymity

The (*k*, ***e***)-anonymity definition is an alteration of *k*-anonymity tailored for numeric data and can only be applied to sensitive attributes having numeric values. (*k*, ***e***)-anonymity requires that the range of the equivalence class *e* to be larger than a certain threshold [141].

(*k*, ***e***)-anonymity is susceptible to *proximity attack* [142]. A proximity breach occurs when the adversary predicts a short interval for an individual's numeric sensitive attribute's value but not the exact value for the sensitive value itself.

3.3.2.7 (ϵ, m)-anonymity

The (ϵ, m)-anonymity definition is a refinement to (k, e)-anonymity to overcome a proximity breach, and similarly, is only applicable to sensitive attributes having numerical values. The (ϵ, m)-anonymity requires that given an equivalence class e, for every sensitive value $sv_i \in e$, then at most $1/m$ of the tuples in e can have sensitive values 'similar' to sv_i. The factor of ϵ quantifies the similarity. The authors list two ways to quantify the numerical similarity between two numerical values of sensitive attributes [142]: two values are similar if their absolute difference is less than the parameter ϵ, i.e. $|sv_i - sv_j| \leq \epsilon$, and relative similarity where sv_i is similar to sv_j if $|sv_i - sv_j| \leq sv_j \cdot \epsilon$.

3.3.2.8 Multi-relational k-anonymity

Most of the privacy definitions examined above deal with a single relation. Multi-relational k-anonymity defines privacy for cases involving multiple relations by modifying k-anonymity to accommodate multiple relations settings.

The multi-relational k-anonymity privacy assumes that there exists a *Person-specific Relation* $\mathcal{T}_{per}$ with respect to a population $\mathcal{U}$ such that $\mathcal{T}_{per}$ has identifiers (primary key attribute or set of attributes) that uniquely correspond to an individual in population $\mathcal{U}$. Given a set of relations $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$, containing foreign keys along with quasi-identifiers as well as sensitive attributes. The notion of multi-relational k-anonymity is to apply k-anonymity at owner level instead of the record level. As there can be many records, spreading over multiple relations, belonging to a single owner. Multi-relational k-anonymity requires that for each record owner r^{id} in the join of all relations, i.e. $\mathcal{T}_{per} \bowtie \mathcal{T}_1 \bowtie \mathcal{T}_2 \bowtie \ldots \bowtie \mathcal{T}_n$, there are k - 1 other record owners with the same quasi-identifier values [143].

3.3.2.9 δ-disclosure privacy

The *delta*-disclosure privacy definition can be considered a more restrictive version of *t*-closeness. Given a set of quasi-identifiers $QI \in Attr \setminus SenAttr$ then tuples r_i and r_j are quasi-identifier equivalent if $r_i[QI] = r_j[QI]$. The relation $\mathcal{T}$ can be divided into quasi-identifier equivalence classes as done in *k*-anonymity. The delta δ-disclosure privacy defines that *"an equivalence class e is δ-disclosure-private with respect to the sensitive attribute in SAV if, for all $sv_i \in SAV$, $\left|log\frac{p(e,sv_i)}{p(\mathcal{T},sv_i)}\right| < \delta$. Where $p(X, sv_i)$ is the probability that a randomly chosen member of X has sensitive attribute value sv_i."* A relation $\mathcal{T}$ is δ-disclosure private if for every $e \in \mathcal{E}$ is δ-disclosure private [144]. However, the δ-disclosure privacy definition is too strong for practical reasons because an equivalence class may not be able to cover all the sensitive attribute values.

3.3.3 Differential privacy (ϵ-differential Privacy)

Differential privacy requires that any given disclosure is, within a small multiplicative factor that is ϵ, just as likely regardless of whether or not the individual's record in the relation [126]. Intuitively, consider two datasets DS_1 and DS_2, only differing is in one record. Res_1 and Res_2 are the result of query Q to DS_1 and DS_2. Res_1 and Res_2 must be indistinguishable from each other in order to fulfil differential privacy requirement. This is usually achieved by the addition of noise to the query results.

Many differential private algorithms have been proposed [145, 146, 147, 148, 149, 150]. Differentially private algorithms are usually designed for interactive query settings; however, with a limitation that they answer only a limited number of queries – governed by privacy budget. The privacy budget regulates that number of queries after which the answers to the queries no longer considered free from privacy risk. Additionally, in general, differential privacy works for statistical databases. Statistical databases deal with aggregates, in contrast with microdata release. In some scenarios

the amount of noise added to the output degrades the utility of the data [151].

Besides the reviewed definitions the privacy literature encompasses many formal privacy definitions and models including integral privacy [152], (X,Y)-privacy [153], and β-likeness [154].

3.4 Summary

This Chapter introduced some of the definitions of anonymity. Observations from the literature are summarized as follows:

- ***k*-anonymity as a foundation:** k-anonymity being one of the first privacy definitions, provided a foundation to further privacy research. Several privacy definitions are based on k-anonymity.
- **Majority of the definitions are for one-time publication:** the privacy literature encompasses a large number of privacy definition, mainly for one-time publication of a single relation [24, 127, 25, 128, 153, 140, 142, 155, 144].
- **Research lacks privacy definitions supporting interactive query settings for microdata release:** the privacy research lacks privacy definitions and mechanisms that work in interactive setting. While differential privacy supports interactive queries, the mechanisms are constrained in the number of queries permitted, and secondly, it allows aggregate queries instead of microdata (row-level data), in some scenarios the amount of noise added degrades the utility of the data.
- **Susceptibility to attacks:** privacy definitions have a level of susceptibility to attacks. There exist vulnerabilities and scenarios where the privacy restriction laid down by the privacy definition is met, but privacy is compromised. Their weaknesses are widely studied in the literature; for instance, the paper from

Josep Domingo-Ferrer and Vicenç Torra have discussed the shortcomings of well-known syntactic privacy models in [137]. Therefore, novel ways to detect these privacy attacks are always desirable.

- **Building privacy-preserving interactive query mechanism – a challenge:** in the present era of big data and data analytics, approaches supporting unlimited interactive querying along with permitting the queries to return microdata while preserving privacy is desirable. Therefore, weaker notions of privacy for interactive query settings for microdata release are potential starting point. However, development interactive query settings for microdata release is a challenging task as multiple releases may give rise to inferences being made by an adversary.

3.5 Conclusions

The privacy literature encompasses a large number of privacy definitions, mainly for one-time publication of a single relation. Most of these definitions are evolved from k-anonymity. k-anonymity and differential privacy can be considered the two mainstream definition of privacy. The privacy research lack privacy definitions and mechanisms that works for interactive settings for microdata release. Differential privacy supports interactive querying, but it is contained by the limited number of queries and allows aggregate queries. Privacy definitions, in various scenarios, are susceptible to attacks – where privacy definitions are met, but still the privacy is compromised. Therefore, novel ways to detect these privacy attacks are always desirable.

Chapter 4

Anomalous DBMS Access Detection Using N-Grams

"Everything begins with an idea."

Earl Nightingale (1921 – 1989)

4.1 Introduction

Chapter 2 reviewed the well-known techniques in the literature that detect malicious queries made by insiders to DBMS. The reviewed techniques can be used to detect individual malicious queries made by an insider. However, while a query can be safe in isolation, when combined with surrounding (in a sequence of) queries, it might result in an undesirable action. Therefore, it is insufficient to identify the anomaly by the query alone, and one must also correlate a single query with its surrounding queries. The research question that arises is whether one can build behavioural-based anomaly detection systems for the detection of malicious accesses to DBMS by considering

sequences of queries. This chapter answers this question (the first research question stated in Chapter 1), that is, whether one can build behavioural-based anomaly detection systems to detect malicious accesses, manifested in sequences of queries rather than a query in isolation, to a DBMS and whether these malicious accesses might encompass privacy violations? In this chapter, the use of sequences of SQL queries in the modelling of insider behaviour to detect malicious insider access is considered. While the insider may hold the correct access permission to make the query, the particular query may not be considered a 'normal' action by the user. This chapter considers how patterns of normal query behaviour can be learnt from DBMS audit logs using n-grams, and how these patterns can be in turn used to detect malicious queries made by insiders.

Section 4.2 describes an approach to anomaly detection in DBMS logs based on n-grams and describes the steps involved in the construction of a profile that represents normal sequences of user queries. The training and the detection phases are described in Section 4.3 and Section 4.4, respectively. A case-study based on detecting anomalies in a banking application is discussed in Section 4.5. Section 4.6 discusses a mimicry attack. Section 4.7 demonstrates how to achieve precision in modelling querying behaviour by considering additional features in constructing profiles.

4.2 Modelling Normative Query Behaviour using N-Gram

In this chapter, an n-gram based approach is proposed, for detecting malicious accesses to DBMS that considers sequences of SQL queries instead of considering an SQL statement in isolation. The inspiration to adopt n-grams comes from early work in [30, 29] that considered the problem of modelling normative behaviour of an application based on sequences of system calls. In [30, 29] system calls were extracted from the

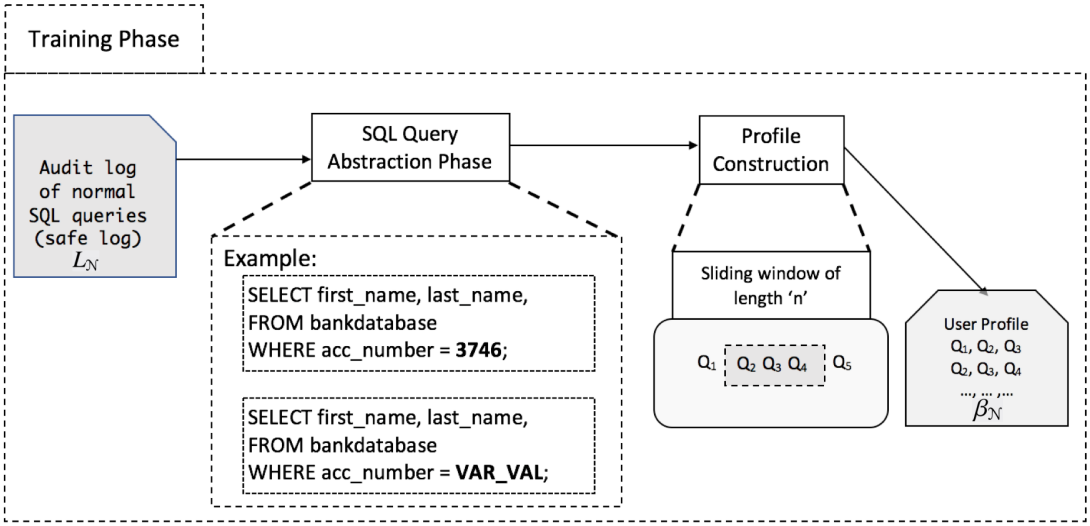


Figure 4.1: Training phase of the proposed n-gram based model. The first step is of data collection in the form of audit logs. Abstraction is chosen in the second step, followed by the construction of a normative profile.

system logs and used to construct sets of n-grams representing 'self'. These sets of n-grams (self) defined the normative behaviour of the application. Past research has shown that n-grams can capture short-range correlation in application programs, for example, in Sendmail [30, 29] along with its effectiveness in intrusion detection in general [156].

The normative profile is a set of n-grams that model the acceptable/normal query sequences. This model is 'mined' from the DBMS log, subsequent query sequences are checked against this profile, and a mismatch is considered to be an anomaly. The training phase of the proposed n-gram approach requires data collection in the form of audit logs for mining of the model, and an SQL query abstraction step creates abstraction of the query that is suitable for mining the model followed by the construction of normative profile. In the detection phase, a run-time profile is constructed and compared against the normative profile to check for deviations from normative querying behaviour. Figures 4.1 and 4.2 depicts the architecture of the proposed n-gram-based anomaly detection approach.

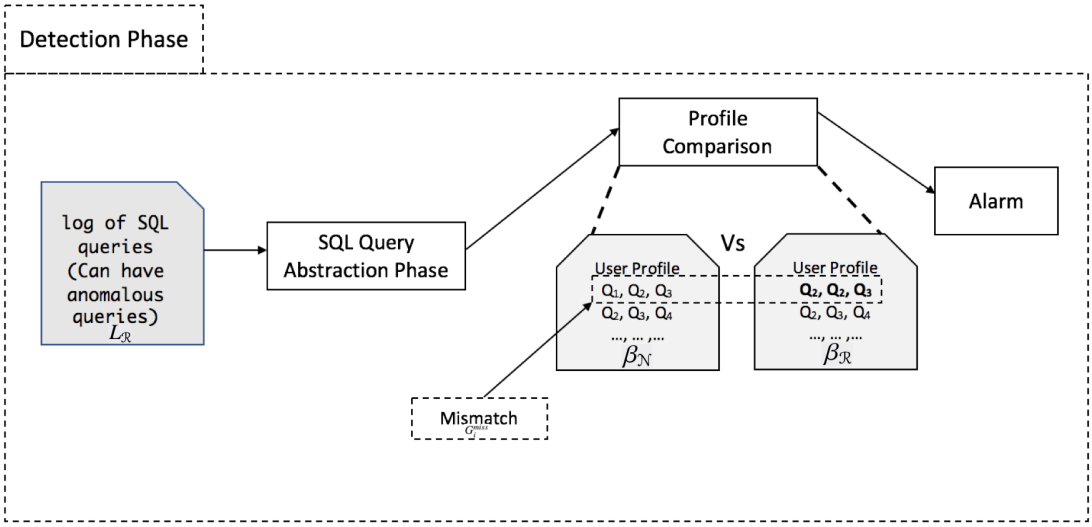


Figure 4.2: Detection phase of the proposed n-gram based model. The data collection, SQL query abstraction and profile construction steps are same as of training phase. In the detection phase, a *run-time profile* is constructed from run-time logs. The run-time profile is compared with the normative profile. Mismatches are an indication of an anomaly.

4.3 Training Phase

The steps involved in the training phase, including the availability (collection) of safe audit logs, generating SQL query abstraction, and constructing a normative profile of behaviour (profile construction phase), are described in the following subsections.

4.3.1 Availability of an Anomaly-free Audit Log

It is assumed that an audit log of SQL statements is available for the training phase. This audit log is assumed free from any malicious query or transaction. An audit log L is a sequence of SQL queries $Q_1, Q_2, Q_3, \ldots, Q_n$, where each Q_i can be any of the SQL query command types, i.e., `SELECT`, `UPDATE`, `INSERT` or `DELETE`.

4.3.2 Chosen SQL Query Abstraction

Section 2.3.3 described the SQL query abstraction that is a tuple representation of an SQL query and consists of query features like relation name, attribute names, the amount of returned data or any statistics on the returned data. In this dissertation, various specializations of query abstractions are used and described in their respective chapters. A naïve specialization consider only the SQL command type (i.e., `INSERT`, `SELECT`, `UPDATE`, `DELETE`) is selected to represent the original SQL query. This naïve specialization is too coarse grained and does not take into account the queried attributes. Therefore, it gives us a less precise representation of user's querying behaviour and significantly increases the number of false negatives. Intuitively, one would be interested in using an entire SQL query. However, using an entire SQL query leads to a high number of false positives in addition to precise but very large profiles. If a profile is to represent normal behaviour then, the objective of abstraction is to have a representation between that lies somewhere a naïve specialization and the entire SQL query.

An abstraction of an SQL query Q_i is denoted as $Abs(Q_i)$. The adopted query abstraction technique in this chapter has also been studied in [9]. This query abstraction technique in [9] provides an abstraction of a SQL query by representing the query that lies somewhere between a coarse grained and a fine grained specialization of query abstraction. The query abstraction approaches are studied in the domain of query log summarization. An in-depth study on SQL query abstraction falls in the domain of query log summarization and beyond the scope of this Chapter. The design of proposed n-gram is modular in nature, that is, one can substitute another SQL abstraction technique from query log summarization domain.

The query abstraction technique replaces the constant values in a query Q_i with placeholders (literal 'VAR_VAL'), and is denoted as $Abs(Q_i)$. $Abs(L)$ is defined as the mapping of $Abs(Q_i)$ over the elements Q_i of L. The reason to chose this query abstraction

Table 4.1: Examples of deployed SQL abstractions.

Q_i	SQL statement	SQL query abstraction $Abs(Q_i)$
Q_1	`SELECT city FROM bankDatabase WHERE id = 2`	`SELECT city FROM bankDatabase WHERE id = VAR_VAL`
Q_2	`SELECT city FROM bankDatabase WHERE id = 9`	`SELECT city FROM bankDatabase WHERE id = VAR_VAL`
Q_3	`SELECT city FROM bankDatabase WHERE id = 3`	`SELECT city FROM bankDatabase WHERE id = VAR_VAL`
Q_4	`SELECT city FROM bankDatabase WHERE id = 3 AND Name = "Alice"`	`SELECT city FROM bankDatabase WHERE id = VAR_VAL AND Name = VAR_VAL`

is that it gives us a reasonable level of precision in capturing the querying behaviour of user. A more fine grained abstraction would require some symbolic evaluation of the queries which was beyond the scope of an n-gram based approach. Table 4.1 shows example mappings of a query to its abstraction.

4.3.3 Building a Normative Model of Behaviour

This chapter describes a model of normative behaviour built from SQL query audit logs. Normative behaviour can be constructed, based on a variety of perspectives, using the SQL log. For example, in case of a user, $user(L, Uid)$ returns the abstract log $Abs(L)$ containing those queries in L executed on behalf of a user with user ID Uid. Similarly, $role(L, R)$ returns the abstract log $Abs(L)$ containing those queries in L executed on behalf of users in the role R. For the sake of clarity and without the loss of generality, we assume that L contains queries made by a single user and $Abs(L)$ represents its abstraction.

While n-grams have their origins in computational linguistics [157, 158] and natural language processing [159], they are well suited for modelling short-range correlations

between events in logs [28]. N-grams are sub-sequences of event generated by sliding a window of size 'n' over a log of events. When $n = 2$ the resulting sub-sequences / n-grams are known as bi-grams while in case of $n = 3$ the sub-sequences are known as tri-grams.

Given a sequence L of SQL queries (abstractions), $ngram(Abs(L), n)$ is the set of all sub-sequences of size 'n' that appear in $Abs(L)$. For example, the bi-gram model for the log abstraction, that is, the sequence of query abstractions ($Abs(Q_1)$, $Abs(Q_2)$, $Abs(Q_3)$,$Abs(Q_4)$), in Table 4.1 is $\{\langle Abs(Q_1), Abs(Q_1)\rangle, \langle Abs(Q_1), Abs(Q_4)\rangle\}$. An n-gram profile is denoted as $\beta = ngram(Abs(L), n)$.

Let $\beta_{\mathcal{N}} = ngram(Abs(L_{\mathcal{N}}), n)$ be the n-gram profile constructed from SQL log $L_{\mathcal{N}}$. This normative profile, in essence, forms a baseline for comparison of later behaviours; therefore, it may also be referred to as a baseline profile.

4.4 Detection Phase

In the detection phase, a run-time profile is constructed in the same manner as the normative profile is constructed but with a run-time log. If $L_{\mathcal{R}}$ is a collected run-time log then The run-time profile is denoted as $\beta_{\mathcal{R}} = ngram(Abs(L_{\mathcal{R}}), n)$.

A comparison between two profiles (sets of n-grams) is defined as, given $Set_i = ngram(Abs(L), n)$, let G_i be one of the sub-sequences from $Set_i = ngram(Abs(L), n)$, for example, $G_1 = \{Abs(Q_1), Abs(Q_2)\}$. Given a normative profile $\beta_{\mathcal{N}}$ and a runtime profile $\beta_{\mathcal{R}}$, a mismatch occurs when these two profiles are compared such that an n-gram G_i^{miss} exists in $\beta_{\mathcal{R}}$ but does not appear in $\beta_{\mathcal{N}}$. This mismatch is labelled as an anomaly. Thus, the comparison of run-time profile $\beta_{\mathcal{R}}$ with normative profile $\beta_{\mathcal{N}}$ is given by $\beta_{\mathcal{R}}$ - $\beta_{\mathcal{N}} = ngram(Abs(L_{\mathcal{R}}), n)$ - $ngram(Abs(L_{\mathcal{N}}), n) = S_{miss}^{\beta_{\mathcal{R}},\beta_{\mathcal{N}}}$, where $S_{miss}^{\beta_{\mathcal{R}},\beta_{\mathcal{N}}}$ is a set of labelled anomalies.

4.5 Evaluation of the N-Gram based Approach

A challenge in this domain of research is the lack of benchmark datasets. Real-world datasets of SQL queries, both anomaly-free and with malicious sequences of queries, may exist in organisations; however, they are not available to researchers. Organisations are hesitant to share their real-world data due to privacy and security concerns. Therefore, in order to evaluate the proposed model, a synthetic dataset generator was used to generate test data and is described in the next section.

4.5.1 A Synthetic Data Generator for a Banking-style Application

The synthetic data generator mimics a banking-style application constructed by considering the queries made by multiple users (insiders – bank employees). Let us denote the set of users as $U = \{Uid_1, Uid_2, Uid_3, \ldots, Uid_n\}$. Each user Uid is selected randomly to appear to make queries in the log. Query templates for mimicking typical banking transactions such as opening an account, closing an account, transferring an amount from one bank account to another account and depositing an amount to a bank account, and withdrawal of an amount from a bank account were specified. Each transaction consisted of multiple SQL queries. Each transaction TX for a given user Uid_i is randomly selected and executed. These transactions mimicked the transaction carried out by insiders (bank employees).

For evaluation of the proposed approach, $10,000$ transactions were executed in the application system that resulted in the generation of an audit log consisting of approximately $28,000$ SQL statements. Two logs were generated: a log of 'normal' behaviour $L_{\mathcal{N}}$, and a test run-time $L_{\mathcal{R}}$ log, initially of normal activity.

It is worth mentioning that in different environments different ways are employed by insiders to query the databases, for instance, there are scenarios where insiders query the databases directly while we also have scenarios where insiders query the databases

via Graphical User Interface (GUI). A well-known scenario where insiders access the database directly is of analytics, where insiders execute queries and received responses at run-time and often is the case where the insiders execute batches of queries together and receive the responses of those batch queries for analytics. In scenarios, where the insiders often are DBMS non-experts, the underlying complexities of the DBMS system are covered by a GUI, and in this case, the insider uses the GUI to query the databases. In general, the contemporary DBMS have features, when enabled, logs the user interaction with the DBMS and provide more information besides the user ID of the insider and the executed SQL queries by that insider. In the scenario of database access via GUI, one can easily extract the executed SQL queries by insiders. Therefore both the scenarios provide the required information in the form of audit logs, from which the data, like user Id's and SQL queries executed by that user, can be extracted using some preprocessing. The generated synthetic logs mimic that shape of the data that is actually required to be passed as an input for the proposed approaches.

4.5.2 Selecting Suitable *'n'*

The n-gram profiles $ngram(Abs(L_{\mathcal{N}}), n)$ and $ngram(Abs(L_{\mathcal{R}}), n)$ (test profile) were built from the logs $L_{\mathcal{N}}$ and $L_{\mathcal{R}}$ while varying the size of n-gram, that is, n = 1, 2, 3, ..., 20. The logs $L_{\mathcal{N}}$ and $L_{\mathcal{R}}$ consisted of transactions having transactions size between 3 SQL queries to 7 SQL queries. Figure 4.3 depicts the number of mismatches arising when comparing the test (but normal) log $ngram(Abs(L_{\mathcal{R}}), n)$ against normative (baseline) behaviour, that is, $ngram(Abs(L_{\mathcal{N}}), n)$, for different values of n. As indicated in Figure 4.3 the number of false positives (mismatches) increases with the size of n, as expected – a low number of false positives for n =2, 3, 4, 5, 6, 7. Therefore, n-grams of sizes 2, 3, and 4 are found to be suitable for constructing a profile.

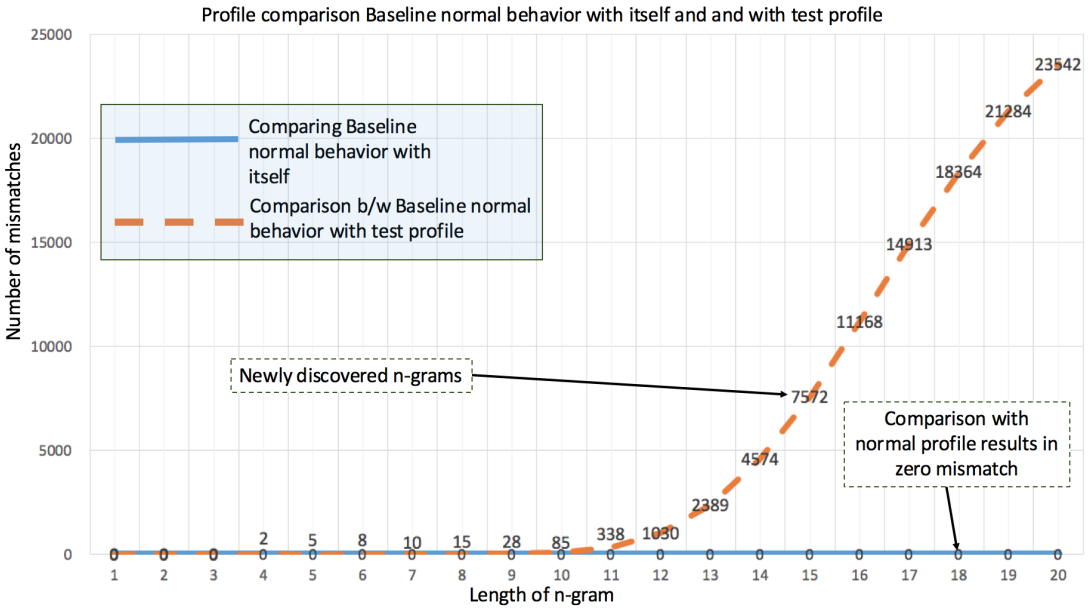


Figure 4.3: Discovered number of n-grams when comparing profiles. The zeros on the x-axis show the comparison of the normative profile with itself.

4.5.3 Attack Scenarios

SQL anomalies can give rise to malicious data observation, malicious data modification, or deletion of data from a database. In this chapter, the focus is on the detection of malicious data observation that arises from anomalous SQL queries. Malicious data observation can be referred to as a privacy violation of an individual in the case where malicious observation of data record in the database pertains individual's personal data. Intuitively, attacks pertaining to malicious data observation is challenging to identify.

Several different attack scenarios were considered. For these attack scenarios, the test run-time log $L_{\mathcal{R}}$ consisted of malicious (adversary) activities simulating attacks for experiments. In the attack scenarios, the anomalous statements had the same query abstraction/skeleton as the statements in the training log; this ensures that a detected mismatch is not based on the query abstraction alone, but on its correlation (or lack thereof) with other events in the log. Any malicious SQL query that has a different SQL query abstraction than that of those in the training log is easily detectable against

a uni-gram. Thus, for the experimentation, the focus is to detect malicious SQL statements mimicking legitimate queries. However, these malicious SQL statements cannot be correlated to a legitimate transaction. In one scenario, 50 SQL statements of malicious observations, matching the structure of the normal SQL statements, were introduced in the log $L_{\mathcal{R}}$. They were inserted into groups of 5 statements at 10 locations selected at random. This version of $L_{\mathcal{R}}$ is referred to as $L_{\mathcal{R}1}$. In the other scenario, the log $L_{\mathcal{R}}$ was perturbed by the insertion of a single malicious SQL query, including scenarios where a single malicious SQL query was inserted between two transactions, during a transaction, at the end and beginning of the log resulting in logs $L_{\mathcal{R}2}$, and $L_{\mathcal{R}3}$, respectively.

Table 4.2: Attack logs and scenarios.

Attack Log	Attack
$L_{\mathcal{R}1}$	Attack 1: 50 malicious queries (5 queries at 10 random locations)
$L_{\mathcal{R}2}$	Attack 2: Malicious SQL query inserted between two transactions
$L_{\mathcal{R}3}$	Attack 3: Malicious SQL query inserted within a transaction

We constructed an attack generator, along the line of a synthetic data generator, that generated various attack sequences in the scenario of the banking environment. The attack generator generated attack sequences for malicious data observation, malicious data modification and malicious data deletion attacks. We focused on the generation of malicious data observations for evaluation. Furthermore, we generated two variants of malicious observation query sequences, where each variant was based on the intent of the adversary. The first variant was when the adversary does not target specific data subjects in the database; rather, the adversaries aim was to reveal sensitive information of a high number of data subjects in the database. The second variant was when the adversary targets specific data subjects. The attack generator had the ability to generated malicious query sequences where each generated SQL query in the malicious query sequences was not malicious in isolation as well as to generate query sequences where

each query in the query sequence can be malicious in isolation. However, in our experimentation, the main focus was on the generation of malicious query sequences, where each query in the sequence was a safe query in isolation. As any malicious SQL query that is different from those in the training log for normative profile is easily detectable against a uni-gram. Therefore, we were only interested detect the ones that have same query abstraction as of the queries in training log but sequences of those queries were resulting in a malicious event.

The considered attacks (Attack 1, Attack 2 and Attack 3) were malicious observation attacks. The Attack 1 consisted of 50 SQL queries, where the aim of a malicious insider (adversary) was to disclose information about account balances of high number of bank's customers and not targeting specific customers (victims). Therefore the 50 SQL queries in the attack represented the browsing behaviour of an insider. For Attack 2 and Attack 3, the malicious insider carried out targeted observations where the aim of the adversary was to disclose targeted customer's details by articulating a SQL query that narrowed down to a single data subject.

Figure 4.4 shows the number of mismatches, for the various sizes of n-grams, that lead to the detection of malicious queries, as mentioned above in the attack scenarios. For instance, malicious query sequences in $L_{\mathcal{R}2}$ resulted in 5 mismatched n-grams when the size of n-gram was 3. N-grams of size 2, 3 or 4 were used, and detected the anomalous queries inserted in logs $L_{\mathcal{R}1}$, $L_{\mathcal{R}2}$, and $L_{\mathcal{R}3}$.

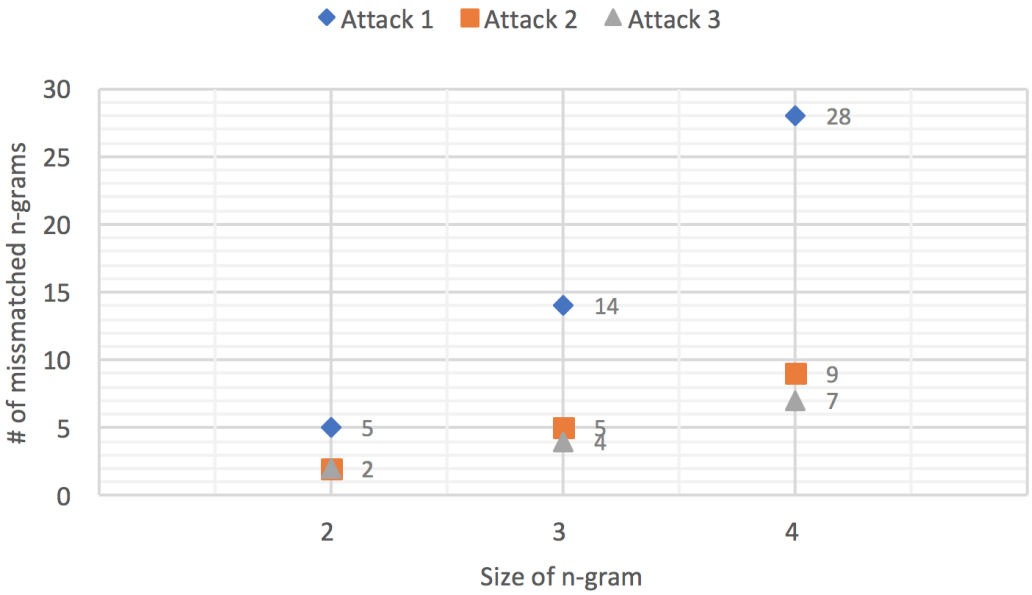


Figure 4.4: The figure shows the number of mismatched n-grams, for various sizes of n-gram, that indicated the presence of anomalous queries (attacks) in the audit logs. Attack 1, 2, & 3 represents profiles constructed using logs $L_{\mathcal{R}1}$, $L_{\mathcal{R}2}$, and $L_{\mathcal{R}3}$, respectively.

The considered attack scenarios in this Section were malicious observation attacks by insiders. The other types of attacks by an insider in the context of DBMS include malicious data modification and deletion attacks. The malicious observation attacks are more challenging to detect compared to malicious data modification and deletion attacks. The is because in malicious data modification and deletion attacks, besides that malicious querying behaviour is inconsistent with the normative querying behaviour, the underlying data also changes due to the carried out modification or deletion, and this can be considered as one of the indications of an anomaly. Deviation from normative querying behaviour is a common indication for these attacks. The proposed n-gram approach, in principle, detects the deviations from normative querying behaviour which implies its effectiveness for detecting malicious data modification and deletion attacks as well.

4.5.4 Computational Complexity

N-grams has been widely used in various domain. The computational complexity of the algorithm for the generating normative profile in the training phase of the n-gram approach is linear, that is, O(n) since it requires generating n-grams of query abstractions. In our case this computational complexity is same as the computational complexity of generating n-grams in any other application [160]. The comparison of profiles in the detection phase consists of the algorithm for the generation of a run-time profile and the algorithm for the profile comparisons (run-time profile is compared against normative profile). The complexity of run-time profile generation algorithm is the same as of the algorithm that generates a normative profile in the training phase, that is, O(n). The algorithm for profile comparison also has the linear complexity O(n). Therefore, we can say that the n-gram approach has an overall linear complexity.

4.5.5 Anomaly Response

The mismatches are an indication of an anomaly and trigger an alert signal. Response to an alert signal can range, depending on the level of sensitivity of the environment, from dropping the response of anomalous SQL queries, revoking the user access privileges or disconnecting the session. For instance, in a sensitive environment like banking or health database systems, an immediate response is to suppress the response to an anomalous query and in the case of a more less critical, a query can be labelled as anomalous and logged for later manual inspection by an information security officer or database administrator. Responses to anomalies (response models) is a separate line of research. A taxonomy of intrusion response systems is presented in [161]. A refined policy framework that can govern the responses to anomalous SQL queries is still desirable.

```
⟨ SELECT first_name, last_name, status, current_balance
FROM bankdb
WHERE account_number = VAR_VAL,
UPDATE bandb SET current_balance = VAR_VAL
WHERE account_number = VAR_VAL,
SELECT first_name, last_name, status, current_balance
FROM bankdb
WHERE account_number = VAR_VAL,
UPDATE bandb SET current_balance = VAR_VAL
WHERE account_number = VAR_VAL ⟩
```

Figure 4.5: A sample n-gram from the normative profile, depicting a transfer of an amount from one client's bank account to another client's bank account.

4.6 Mimicry Attacks

The proposed n-gram based approach uses sequences of queries to model an insider's normative behaviour by using n-grams. However, an inside attacker, who is familiar with the working of the detection system, can mimic normative access patterns to evade the detection system. Mimicking normative behaviour to carry out malicious tasks by deceiving the detector into believing the anomalous behaviour is normative is known as a *mimicry attack*. Behavioural-based approaches are typically susceptible to mimicry attacks [162, 163, 164, 165]. In general, an insider knowing the exact working details of the system has the potential to execute mimicry attacks.

In this section, a mimicry attack on the proposed n-gram based approach is demonstrated, using the banking case study, where an informed adversary can discover account balances of victims without detection. Suppose that, an insider, who regularly helps the bank's clients to carry out transactions, determines the sequence of queries that passes the detection system.

This malevolent insider, wanting to know the current account balances of several clients, evades the intrusion detection system, by articulating queries that fit the n-gram in shown Figure 4.5. Figure 4.6 shows a possible shape that a malicious insider's queries can take. Two valid transactions TX_1 and TX_2 are shown in Figure 4.6. These

TX_1	Q_9	`SELECT first_name, last_name, status, current_balance FROM bankdb WHERE account_number = 029192;`
	Q_{10}	`UPDATE bandb SET current_balance = 900 WHERE account_number= 029192;`
	Q_{11}	`SELECT first_name, last_name, status, current_balance FROM bankdb WHERE account_number = 073846;`
	Q_{12}	`UPDATE bandb SET current_balance = 1384 WHERE account_number = 073846;`
TX_2	Q_{13}	`SELECT first_name, last_name, status, current_balance FROM bankdb WHERE account_number = 29192;`
	Q_{14}	`UPDATE bankdb SET current_balance = 1000 WHERE account_number = 29192;`
	Q_{15}	`SELECT first_name, last_name, status, current_balance FROM bankdb WHERE account_number = 073846;`
	Q_{16}	`UPDATE bankdb SET current_balance = 1284 WHERE account_number = 073846;`

Figure 4.6: This figure depicts queries made by an inside attacker. First, an amount is transferred from the first victim's account to the second victim's account, and then the same amount is transferred back to first victim's from second victim's account.

transactions represent the transfer of the amount from one account to another, thus resulting in a sequence of queries $Q_9, Q_{10}, Q_{11}, Q_{12}, Q_{13}, Q_{14}, Q_{15}, Q_{16}$. These transactions are kept simple in this example for the purpose of clarity. Only Q_9 and Q_{13} are needed to reveal `current_balance` of victims accounts to a malicious insider. While queries like Q_9 and Q_{13}, if made in a sequence, would be expected to give rise to a detectable anomalous behaviour when the n-gram approach is deployed. However, the inside attacker bypasses the n-gram based detection system by padding the remaining queries with Q_9 and Q_{13}, to make the transaction look legitimate, thus revealing account balances of clients.

4.7 Resisting Mimicry Attacks

It is desirable to have a detection system that detects this type of mimicry attack. A way to detect these attacks is to consider additional features while modelling querying behaviours and therefore, making the normative profile more precise. The above mentioned n-gram model is an approximation of querying behaviour, the addition of more features (context/data/syntax-centric) into an n-gram model makes it more precise; therefore, making it hard for an inside attacker to execute mimicry attacks. One way to add further features is to consider query analytics with respect to the time frames in which the queries are made – a context-centric feature. In the next section, we show that considering such query analytics does provide more precision. In subsequent sections, an approach to constructing a normative profile on the basis of this hypothesis is described.

4.7.1 Query Analytics-based Model of Normative Behaviour

The n-gram-based modelling of behaviour presented in Section 4.2 is augmented by considering the additional features of query analytics bounded by time periods. In the training phase, query frequency related analytics are computed and made part of the normative profile, while in the detection phase, query analytics are computed at run-time, and are then compared against the ones in the normative profile.

Given the SQL statements in L are time-stamped. Let τ be the total duration of the transaction log which is divided into further time frames, $\tau = [\Gamma_1 , \Gamma_2 , \Gamma_3 , \ldots , \Gamma_l]$ where $l \in \mathbb{N}$. In order to build a model of normative behaviour based on query frequency, time intervals of uniform length were considered. Let $\Gamma_l = \langle [X_1 \bullet\!\!-\!\!\bullet Y_1], [X_2 \bullet\!\!-\!\!\bullet Y_2], [X_3 \bullet\!\!-\!\!\bullet Y_3], \ldots, [X_i \bullet\!\!-\!\!\bullet Y_i] \rangle = \langle P_1, P_2, P_3, \ldots, P_i \rangle_l = [X_1 \bullet\!\!-\!\!\bullet Y_i]_l$, where $i \in \mathbb{N}$ and X denotes the starting time, and Y the ending time. The number of queries made in P_i are φ_{P_i} while the number of queries made in P_i of time frame

Γ_l is denoted as φ_{P_i,Γ_l}. Thus, the number of queries made in the time frame Γ_l are $\varphi_{\Gamma_l} = \sum_{c=1}^{i} \varphi_{P_c,\Gamma_l}$. The number of queries made for the total duration of transactional log τ are $\varphi_\tau = \sum_{b=1}^{l} \sum_{c=1}^{i} \varphi_{P_c,\Gamma_b}$, so φ_τ is the total number of queries in L. For a fine-grained representation of normative behaviour defined in terms of query frequencies, the total number of each type of SQL statement are determined. These determined frequencies are represented as $\varphi_{time_frame}^{\texttt{Statement_type}}$ for example $\varphi_{P_2,\Gamma_3}^{\texttt{SELECT}}$ are the queries made in the sub-frame P_2 of the time frame Γ_3.

Further refinement can be made by considering the frequencies of combination of the SQL commands types with respect to various time frames. For demonstration purpose, the frequencies of only individual command types with respect to various time frames were considered in this work; however, the approach can be extended to consider the combination of multiple command types.

To construct a normative profile, it is assumed that the SQL queries in the audit log in the training phase are free from malicious sequences of queries. The constructed normative profile contains averaged frequencies of queries (or specific command type) over the time frame τ, that is, $Avg_{\varphi_{P_l}} = (\sum_{c=1}^{i} \varphi_{P_i,\Gamma_c})/l$ and $Avg_{\varphi_{\Gamma_l}} = (\sum_{b=1}^{l} \sum_{c=1}^{i} \varphi_{P_c,\Gamma_b})/l$. In the case of specific command type $Avg_{\varphi_{P_l}}^{Statement_type} = (\sum_{c=1}^{i} \varphi_{P_i,\Gamma_c})/l$ and $Avg_{\varphi_{\Gamma_l}}^{Statement_type} = (\sum_{b=1}^{l} \sum_{c=1}^{i} \varphi_{P_c,\Gamma_b})/l$. The choice of selecting τ is left to the user and depends upon the organization where it is deployed.

4.7.2 Evaluation

The following sections present the results of experiments for the evaluation of the proposed query analytics-based model.

4.7.2.1 Experimental Setup

In order to evaluate the query analytics-based model, the same banking-style application is used as described in Section 4.5, representing a banking scenario. It includes transactions including account open and close, withdrawal, deposit and transfer. The transactions are executed by an insider (a bank employee).

For evaluation, τ and Γ were set to existing time frames of 30 days, 24 hours, and 60 minutes, respectively. P_i was set to 60 minutes such that P_1 = 8:00:01 hrs to 9:00:00 hrs, P_2 = 9:00:01 hrs to 10:00:00 hrs and so forth.

4.7.2.2 Mimicry Attack Scenarios

Malicious transactions were crafted such that they pass as normal under the n-gram model, as shown in Section 4.6. Two attack scenarios were considered. In the first attack scenario, the inside attacker performed 15 malicious transactions in one day between 10:00:00 hrs to 11:00:00 hrs. In the second attack scenario, the inside attacker distributes 60 malicious transactions on an entire day. Figure 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12 shows the comparison of query analytics in normative profile with the ones generated for attack scenario 1 and attack scenario 2.

From figure 4.7, 4.8, 4.9, 4.10 and 4.11 the values of φ_{P_i}, $\varphi_{P_i}^{SELECT}$, $\varphi_{P_i}^{INSERT}$ and $\varphi_{P_i}^{UPDATE}$, in attack scenario 1 for the duration of 10:00:00 hrs to 11:00:00 hrs, significantly exceeds the threshold determined from the normative querying behaviours, thus resulting in an indication of an anomaly. For attack scenario 2, the value of $\varphi_{P_i}^{INSERT}$ in Figure 4.7 for the duration of 09:00 hrs to 10:00:00 hrs in attack scenario 2 significantly exceeds the value specified in the normative profile. Similarly, as shown in figure 4.8, 4.9 and 4.11 the values of φ_{P_i} (11:00:00 hrs to 12:00:00 hrs), $\varphi_{P_i}^{INSERT}$ (11:00:00 hrs to 12:00:00, 14:00:00 hrs to 15:00:00 hrs, 15:00:00 hrs to 16:00:00 hrs) and $\varphi_{P_i}^{DELETE}$ (14:00:00 hrs to 15:00:00 hrs, 15:00:00 hrs to 16:00:00 hrs, 16:00:00 hrs

to 17:00:00 hrs) are an indication of anomalies. Additionally, the value of φ_{Γ_l}, $\varphi_{\Gamma_l}^{UPDATE}$, $\varphi_{\Gamma_l}^{DELETE}$ and $\varphi_{\Gamma_l}^{SELECT}$ is also significantly higher then the average values determined in the normative profile.

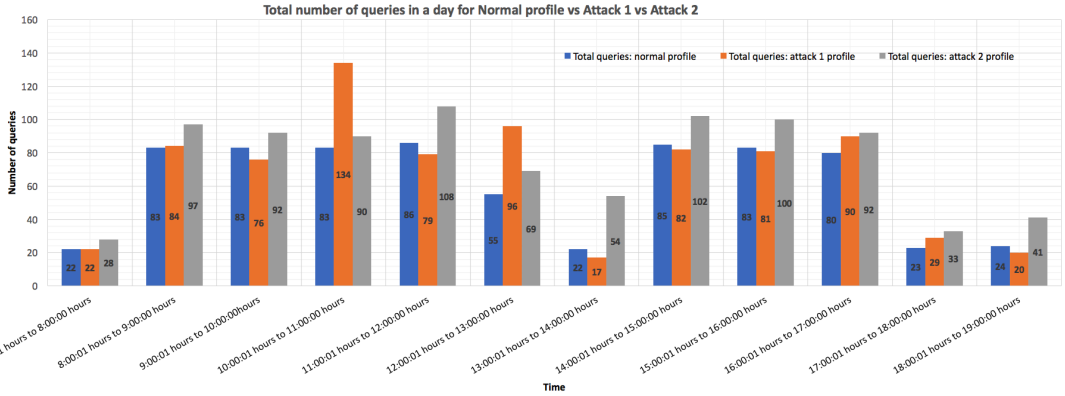


Figure 4.7: The figure shows the values of φ_{P_i} for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 2 (attack 2 profile).

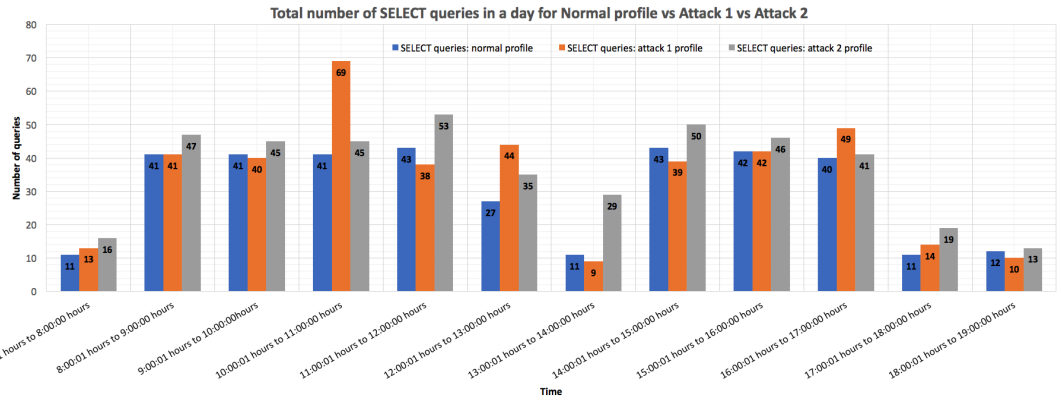


Figure 4.8: The figure shows the values of $\varphi_{P_i}^{SELECT}$ for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 1 (attack 2 profile).

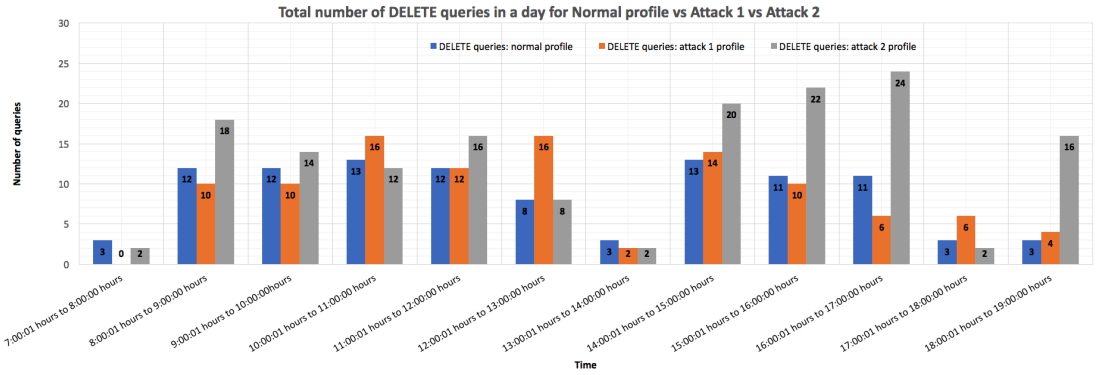


Figure 4.11: The figure shows the values of $\varphi_{P_i}^{DELETE}$ for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 2 (attack 2 profile).

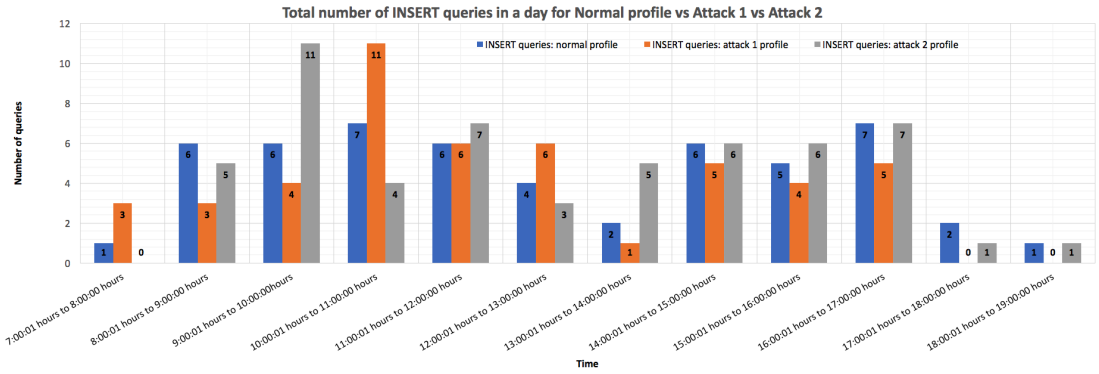


Figure 4.9: The figure shows the values of $\varphi_{P_i}^{INSERT}$ for the normative profile as compared to attack 1 profile and attack 2 profile.

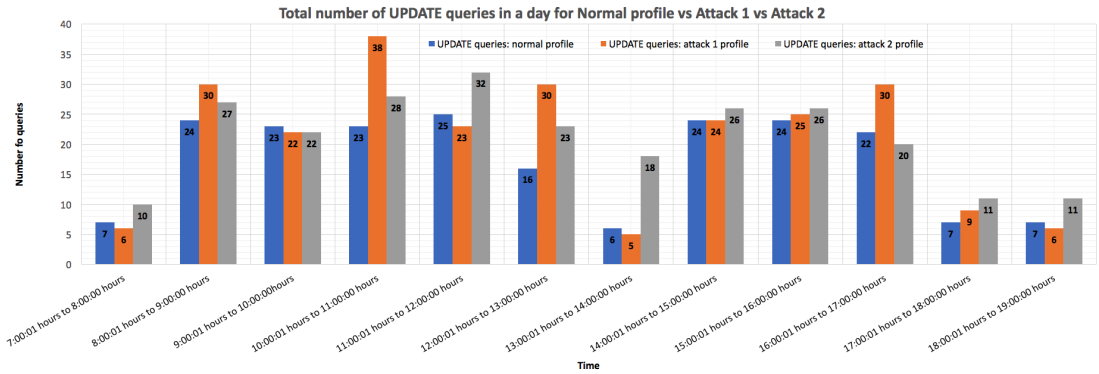


Figure 4.10: The figure shows the values of $\varphi_{P_i}^{UPDATE}$ for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 2 (attack 2 profile).

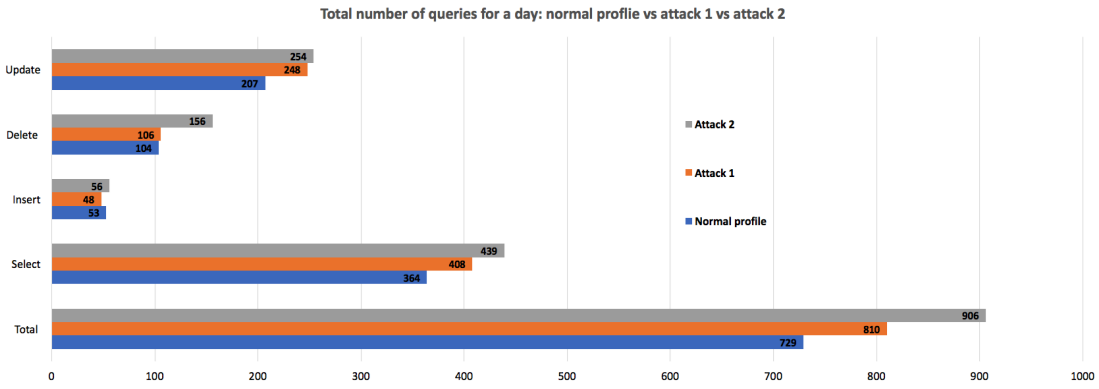


Figure 4.12: The figure shows the values of φ_{Γ_l} for the normative profile as compared to attack scenario 1 (attack 1 profile) and attack scenario 2 (attack 2 profile).

Table 4.3: The table shows the values of φ_{P_i}, averaged over the time frame of τ, made part of the profile of normative behaviour.

φ_{P_i}'s for duration 7:00:00 {Hours} to 13:00:00 {Hours}					
$\text{Avg}\varphi_{P_1}$	$\text{Avg}\varphi_{P_2}$	$\text{Avg}\varphi_{P_3}$	$\text{Avg}\varphi_{P_4}$	$\text{Avg}\varphi_{P_5}$	$\text{Avg}\varphi_{P_6}$
$Avg\varphi_{P_1} = 22$	$Avg\varphi_{P_2} = 83$	$Avg\varphi_{P_3} = 83$	$Avg\varphi_{P_4} = 83$	$Avg\varphi_{P_5} = 86$	$Avg\varphi_{P_6} = 55$
$Avg\varphi_{P_1}^{SELECT} = 11$	$Avg\varphi_{P_2}^{SELECT} = 41$	$Avg\varphi_{P_3}^{SELECT} = 41$	$Avg\varphi_{P_4}^{SELECT} = 41$	$Avg\varphi_{P_5}^{SELECT} = 43$	$Avg\varphi_{P_6}^{SELECT} = 27$
$Avg\varphi_{P_1}^{INSERT} = 1$	$Avg\varphi_{P_2}^{INSERT} = 6$	$Avg\varphi_{P_3}^{INSERT} = 6$	$Avg\varphi_{P_4}^{INSERT} = 13$	$Avg\varphi_{P_5}^{INSERT} = 6$	$Avg\varphi_{P_6}^{INSERT} = 4$
$Avg\varphi_{P_1}^{UPDATE} = 7$	$Avg\varphi_{P_2}^{UPDATE} = 24$	$Avg\varphi_{P_3}^{UPDATE} = 23$	$Avg\varphi_{P_4}^{UPDATE} = 30$	$Avg\varphi_{P_5}^{UPDATE} = 25$	$Avg\varphi_{P_6}^{UPDATE} = 16$
$Avg\varphi_{P_1}^{DELETE} = 3$	$Avg\varphi_{P_2}^{DELETE} = 12$	$Avg\varphi_{P_3}^{DELETE} = 30$	$Avg\varphi_{P_4}^{DELETE} = 23$	$Avg\varphi_{P_5}^{DELETE} = 12$	$Avg\varphi_{P_6}^{DELETE} = 8$
φ_{P_i}'s for duration 13:00:01 {Hours} to 18:00:00 {Hours}					
$Avg\varphi_{P_7}$	$Avg\varphi_{P_8}$	$Avg\varphi_{P_9}$	$Avg\varphi_{P_{10}}$	$Avg\varphi_{P_{11}}$	$Avg\varphi_{P_{12}}$
$Avg\varphi_{P_7} = 22$	$Avg\varphi_{P_8} = 85$	$Avg\varphi_{P_9} = 83$	$Avg\varphi_{P_{10}} = 80$	$Avg\varphi_{P_{11}} = 23$	$Avg\varphi_{P_{12}} = 24$
$Avg\varphi_{P_7}^{SELECT} = 11$	$Avg\varphi_{P_8}^{SELECT} = 43$	$Avg\varphi_{P_9}^{SELECT} = 42$	$Avg\varphi_{P_{10}}^{SELECT} = 42$	$Avg\varphi_{P_{11}}^{SELECT} = 11$	$Avg\varphi_{P_{12}}^{SELECT} = 12$
$Avg\varphi_{P_7}^{INSERT} = 2$	$Avg\varphi_{P_8}^{INSERT} = 6$	$Avg\varphi_{P_9}^{INSERT} = 5$	$Avg\varphi_{P_{10}}^{INSERT} = 5$	$Avg\varphi_{P_{11}}^{INSERT} = 2$	$Avg\varphi_{P_{12}}^{INSERT} = 1$
$Avg\varphi_{P_7}^{UPDATE} = 6$	$Avg\varphi_{P_8}^{UPDATE} = 24$	$Avg\varphi_{P_9}^{UPDATE} = 24$	$Avg\varphi_{P_{10}}^{UPDATE} = 24$	$Avg\varphi_{P_{11}}^{UPDATE} = 7$	$Avg\varphi_{P_{12}}^{UPDATE} = 7$
$Avg\varphi_{P_7}^{DELETE} = 3$	$Avg\varphi_{P_8}^{DELETE} = 12$	$Avg\varphi_{P_9}^{DELETE} = 11$	$Avg\varphi_{P_{10}}^{DELETE} = 11$	$Avg\varphi_{P_{11}}^{DELETE} = 3$	$Avg\varphi_{P_{12}}^{DELETE} = 3$

Table 4.4: The table shows the values of φ_{Γ_l}, averaged over the duration of 30 days, made part of the normative profile.

φ_{Γ_l} averaged over the duration of 30 days ($l = 30$) for a normative profile ($Avg\varphi_{\Gamma_l}$)
$\text{Avg}\varphi_{\Gamma_l} = 729$
$\text{Avg}\varphi_{\Gamma_l}^{SELECT} = 364$
$\text{Avg}\varphi_{\Gamma_l}^{INSERT} = 53$
$\text{Avg}\varphi_{\Gamma_l}^{UPDATE} = 207$
$\text{Avg}\varphi_{\Gamma_l}^{DELETE} = 104$

Insider attacks go unnoticed for months, and even years [7]. The query analytics approach complements other intrusion detection systems, including the n-gram approach, by raising the level of difficulty for an informed malicious insider. There can be a cases where the run-time query frequencies are significantly lower, for a given time frame, then the one specified in the normative profile. In those cases, there is a room for an adversary to execute malicious transactions to match the run-time query frequencies with the ones in the normative profile, thus bypassing the detection mechanism.

4.8 Conclusions

The original work of Forrest et al. [28] translated the application of n-grams from computational linguistics to computer security and demonstrated the effectiveness while modelling a behaviour (self) of the computer system using system calls. The objective of the chapter is to demonstrate that one can use Forrest et al. [28] style n-gram approach to capture querying behaviours and build an anomaly-based intrusion detection system. This chapter demonstrates the effectiveness of using the n-grams to construct expressive profiles from an audit log of SQL queries (or query abstractions) such that a normative (self) behaviour of user (role) querying the system is captured. The construction of the normative profile gives rise to an anomaly style database intrusion

detection system that detects SQL queries made by malicious insiders in DBMS logs. A query abstraction was chosen, as it gives a reasonable level of precision. A naïve query abstraction considers only the SQL command type rendering in the less precise representation of user's querying behaviour and increases the number of false negatives. Usage of an entire SQL query leads to strict rules for normal behaviours thus resulting in high number of false positives as well as huge profiles. Future research could look at other query abstractions, as considered in [3, 10] for single uncorrelated queries. The experiments do show that it is possible to build a useful query abstraction and that n-grams of these queries do capture the short-term correlations inherent in the application. This chapter demonstrates an example of mimicry attack on an n-gram based approach. It is shown that inclusion of additional features, like query analytics, capture behaviours that raise the difficulty level of an adversary to execute mimicry attacks.

A query (and its abstraction) that has the same semantics as a query covered by a normative profile but differs syntactically can give rise to a false positive and approaches that detect anomalous access to databases are susceptible to it. However, recently, an approach to query regularization [166] that unfold queries into a syntactic normal form, has been proposed. The approach attempts to compare queries in terms of their semantics. The query regularization approach has the potential to improve the SQL query abstraction part of the presented approaches in this dissertation. Therefore, as future work, query regularization can be explored to have abstractions of the regularized version of queries to construct n-gram profiles.

Chapter 5

A Semantic Approach to Frequency-based Anomaly Detection of Insider Attacks

"Many of the things you can count, don't count. Many of the things you can't count, really count."

Albert Einstein (1979 - 1955)

5.1 Introduction

The previous chapter demonstrated the modelling of user query behaviours using n-grams and proposed a *role/user-oriented profile* based approach to anomaly detection. A question that arises is whether one can build approaches to model normative behaviour from a different perspective? This chapter considers this question (a part of the second research question stated in Chapter 1) by exploring a DBMS/record-oriented

perspective of modelling access in DBMS and proposes the construction of *record-oriented profiles*.

In this chapter, the focus is to construct record-oriented profiles, and for that purpose the considered features to construct these profiles falls under the category of data-centric features. The data-centric features include, but are not limited to, the amount of data returned or any other statistics related to the resultant data. Anomaly-based IDS approaches that construct profiles using data-centric features can also be referred to as semantic approaches, as they focus on semantics rather than on query syntax [11] as discussed in Chapter 2.

In this context, we consider the detection of *frequent observation attacks* whereby an insider, or group of insiders, make numerous malicious accesses to the same record in the DBMS. These malicious observations can be in collaboration with others or in isolation. The attack is detected by the change in the frequency patterns of access. Real-world examples of frequent observation attacks were reported in [167, 168] where insiders (hospital staff) looked up the medical records of patients in the public eye. In this instance, employees at Palisades Medical Center in New Jersey were suspended after accessing the personal medical records of actor George Clooney who was taken to the hospital after a motorcycle accident. The employees were suspended because they violated a HIPAA regulation [167].

This chapter proposes an anomaly-based IDS that constructs record-oriented profiles by considering the frequency at which the records are accessed - as data-centric feature. The construction of the model utilizes Control Charts that are prevalent in Statistical Process Control (SPC) [14]. Section 5.2 describes the record-oriented model. Section 5.3 present how the record-oriented model can be transformed into role-oriented model. The record-oriented model is evaluated in Section 5.4, and Section 5.5 concludes this chapter.

5.2 Record-oriented Model of Normative Behaviour

In this section, a model for constructing record-oriented profiles of normative behaviour is proposed. Let $\mathcal{T}$ represent a database relation instance. For ease of exposition, it is assumed that each record/tuple r in $\mathcal{T}$ is unique. The proposed approach consists of a learning phase and a detection phase. In the learning phase, a record-oriented normative profile is constructed by determining a range for the number of times each record has been accessed in $\mathcal{T}$ in a given time period **t**. The time period is user-defined, which is dependent on the nature of the target application. For example, in the case of a hospital where a record of a specific patient is queried frequently as opposed to an electricity billing system where a specific record might be queried twice or thrice in a six month period.

We wish to determine the number of times a record (or any attribute of the record) is queried in the time period **t**, referred to as record access frequency. Let $\mathcal{F}_i$ be a numeric counter for the i^{th} record $r_i \in \mathcal{T}$. In contrast to φ that represented the query frequency (as defined in Chapter 4), $\mathcal{F}_i$ represents the record access frequency. When the i^{th} record, or any attribute value in the i^{th} record, is accessed then the value of $\mathcal{F}_i$ is incremented by one. The values of $\mathcal{F}_i$ for each day are stored. The value of time period **t** is left on the organization and the nature of the application. However, for the purpose of exposition, the time period is set to 24 hours (one day) for this work.

The training dataset is denoted by $\mathcal{L}$ that represents the record access frequencies extracted from log L. However, it is possible that the training dataset may have included those behaviours that are infrequent (unusual values for $\mathcal{F}_i$) which we refer to as outliers. Here, an outlier is a record access frequency that is significantly different from record access frequencies for rest of the records in a relation $\mathcal{T}$ within a given time period and potentially is an indication of malicious accesses. Thus, in order to determine the spectrum of normal values for $\mathcal{F}_i$, two scenarios are considered, one that is free from outliers and the other that is susceptible to (with) outliers.

This work utilizes control charts from SPC, which are described in the next section.

5.2.1 Statistical Process Control and Control Charts

Statistical Process Control (SPC) [14] originated from performance monitoring in manufacturing processes. SPC was originally proposed with its application in manufacturing industries where it was used to observe if the process is working as expected during production in order to detect defective products. In SPC, measurements are computed from samples of items produce/manufactured. SPC can be used in the case where a large number of items are being produced. For instance, in the case of shampoo manufacturing, it is not technically possible to fill in exactly the same amount of shampoo in every shampoo bottle as such process are susceptible to variability. Lesser than the claimed amount of shampoo in the bottle may lead the customer to file complaints while filling in extra than claimed leads to a loss in revenue. Thus the aim of SPC is to provide an indication that the production plant is filling the bottles with too much/too little shampoo and it has deviated from the acceptable limits of variability in quantity of shampoo.

SPC uses control charts to provide a history of a running process and to monitor the quality of the processes. Control charts, in essence, are graphs that show measurements and variation among the measurements that are plotted against predetermined limits, during a specific time period. This time period is the time during which the process was being observed. For instance, in the case of the shampoo production plant, the plotted measurement on a control chart is the quantity of the shampoo filled in a single shampoo bottle measured for a sample of produced shampoo bottles on a given day.

A conventional control chart (Shewhart Chart) has several components including, a center line, specification limits (upper and lower), and control limits (upper and lower). The center line is usually the mathematical average of the data. The customers define specification limits, that is, the acceptable tolerance of defects for products or services

while control limits are defined to determine if the process is in statistical control.

In the case of the proposed model, a Centre Line, an Upper Limit, and Lower Limit are used to define constraints, that is, if record access frequencies fall outside these limits, then they are considered as anomalies. The adopted variation of the control chart is shown in Figure 5.1. In the case of the proposed record-oriented model, upper and lower limits are computed using training datasets as described below and the measurements plotted against these predetermined limits are the values for the number of times a specific record is accessed in a certain time period at run-time.

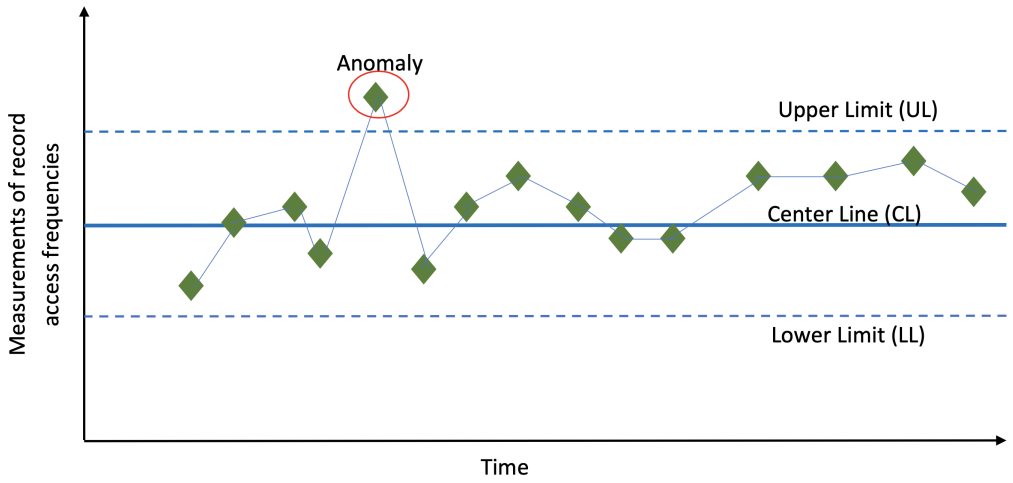


Figure 5.1: The adopted variation of control chart.

In this chapter, an application of SPC in the scenario of detecting malicious access to databases is explored. Where limits of normal access to the database is determined in training phase and in detection record access frequency (analogous to a sample) are plotted on control chart to monitor deviation from normal access frequency.

The reason to use a statistical process control analogy, and specifically control charts, is rooted in the aim of control chart. Their aim is to show how the process changes over time, and in our case, the changes in the record access frequencies over time. Knowing this change in how the records being accessed enables us to fine-tune the computed specification limits in order to handle shifts in record access behaviour in detection phase.

The detection phase of the proposed approach can be integrated with a live system where the record access frequencies can be collected and then plotted over the control chat at run-time. This requires features, typically found in DBMS auditing and monitoring tools, to obtain record access frequencies at a run-time. However, the value of time period **t** needs to determined before the system is live. The selection of an appropriate time period **t** depends on the organization and the nature of the application where the mechanism is deployed.

5.2.2 Outlier-free Scenario

As mentioned earlier, in this chapter, two scenarios are considered, that is, when the training dataset is free from outliers and the one in which the training dataset has outliers. Therefore, to address both outlier-free and with-outlier training dataset scenarios, a different variation of a control chart is used by each scenario.

This section describes the scenario where an outlier-free training dataset is available. For the outlier-free scenario, mean and standard deviation are used to determine the center line and limits. Given functions $\mu()$ and $\sigma()$ that computes the mean and standard deviation, respectively, then $\mu(\mathcal{L})$ and $\sigma(\mathcal{L})$ gives us the mean and standard deviation for the record access frequencies $\mathcal{L}$, respectively. For the outlier-free scenario, the computed limits for the control chart are as follows Center Line = $\mu(\mathcal{L})$, Upper Limit = $\mu(\mathcal{L}) + 3\sigma(\mathcal{L})$ and Lower Limit =$\mu(\mathcal{L}) - 3\sigma(\mathcal{L})$. The record-oriented profiles contain these computed limits.

In the detection phase, a control chart is generated for each record. The counter $\mathcal{F}_i$, starts with the value zero at the beginning of a new time period **t**. The values of $\mathcal{F}_i$ are recorded in $\mathcal{L}_{run}$ for time period **t** where $\mathcal{L}_{run}$ denotes the set record access frequencies at run-time while $\mathcal{L}$ are the ones used for training. Values for $\mathcal{F}_i$ are recorded in $\mathcal{L}_{run}$ and are plotted on the control chart for each day. For the upper limit and lower limit, $\mu(\mathcal{L}) + 3\sigma(\mathcal{L})$ and $\mu(\mathcal{L}) - 3\sigma(\mathcal{L})$ are chosen, respectively. The $\pm 3\sigma()$ limits have shown

to cover 99.87% of the values in the dataset when normally distributed [169]. The upper and lower limits can be tuned depending on the environment. Values of $\mathcal{F}_i$ above and below $\mu(\mathcal{L}) + 3\sigma(\mathcal{L})$ and $\mu(\mathcal{L}) - 3\sigma(\mathcal{L})$, respectively, are considered anomalies.

5.2.3 Handling Outliers

An assumption made before designing anomaly-based database intrusion detection systems is the availability of an outlier-free dataset. The data used in the training phase must be free from any malicious behaviours so that the behaviour captured in the training phase is an accurate reflection of normative behaviour. However, in real-world scenarios, it may become a challenge to ensure whether the training data satisfies this assumption. Therefore, modelling of normative behaviour from the training data that may have malicious behaviours (outliers) needs attention and considered in this section.

In the scenario, where outliers influence data, Median Absolution Deviation (MAD) [170] is considered more robust than other measures of central tendency such as standard deviation [169]. In the case of standard deviation, the distances from the mean are squared; thus an outlier can have a strong influence on standard deviation, which is not the case in MAD as it is a distance from the median.

MAD is less sensitive to the presence of outliers in the data and is therefore used to determine the spectrum of normal values for $\mathcal{F}_i$. Let $\mathbf{m}(\mathcal{L})$ denote the median value for training dataset $\mathcal{L}$. The median absolute deviation is defined as $MAD(\mathcal{L}) = \mathbf{m}(|\mathcal{L}_i - \mathbf{m}(\mathcal{L})|)$.

For the with-outlier scenario, the limits are given by Center Line = $\mathbf{m}(\mathcal{L})$, Upper Limit = $\mathbf{m}(\mathcal{L}) + 2MAD(\mathcal{L})$ and Lower Limit = $\mathbf{m}(\mathcal{L}) - 2MAD(\mathcal{L})$. Values of $\mathcal{F}_i$ above and below $\mathbf{m}(\mathcal{L}) + 2MAD(\mathcal{L})$ and $\mathbf{m}(\mathcal{L}) - 2MAD(\mathcal{L})$, respectively, are considered anomalies for the corresponding time period. In both scenarios, the anomalies can be further

inspected by a security officer.

Past literature has not considered frequent observation attacks, carried out while insiders are working in collaboration. The advantage of the record-oriented model is that when several roles/users (employees) are accessing the same record, the record-oriented model detects these attacks of collaborative nature.

5.2.4 Oversight-anomalies

The detection mechanism aims to detect records that are frequently queried, such as the incidents reported in [167, 168]. However, the proposed approach also enables the detection of records that are less frequently queried as compared to what is normal, that is when the record access frequency drops below the lower limit. This type of anomaly is referred to as *oversight-anomaly*. For instance, in a scenario of a hospital where a doctor or a nurse missed the daily check-up of a patient represents an absence of normative behaviour.

5.2.5 Redefining Limits at Run-time

In the detection phase of this proposed approach, a graphically represented control chart shows the pattern of access to a record over a series of time periods. This gives us insights that can potentially be used to update the upper and the lower limit for a particular record. The concept is analogous to feedback loops in control systems where feedback loops enable the system to fine-tune its performance. For example, consider the mean, the upper limit and the lower limit for record r_0 were 33.5, 49.25 and 17.75, respectively. In the detection phase, access frequency of record r_0 with respect to one day was plotted for each day for 30 days. After 30 days, a pattern emerged that access frequency of record r_0 is always above 40.25. Using this insight, the upper and lower limits for r_0 can be re-defined provided that this behaviour was inspected by a security

officer and marks as normal for the case of this particular record r_0.

In the scenario described above, record r_0 exhibited a continuous consistent level of deviation from the initially determined limits. There can be a case of an infrequent accesses relative to the dataset, where a particular record is accessed every quarter or with a therefore, the fine tuning may be unfitting. This approach of monitoring the past behaviour in the detection phase enables, in scenarios where there is a particular record exhibits a continuous consistent level of deviation from the initially determined limits, the refinement of the existing record-oriented profile. A point worth mentioning is that it might take time for the record access patterns to emerge.

Refined specification limits are that it allows us to handle shifts in record access behaviour in the detection phase. The recalculation of specification limits can be automated, which requires recalculating the specification limits after regular intervals. For example, in the case where a record access frequency of a record is always outside the specification limits for 30 days, then specification limits are redefined for that record.

5.3 Translating the record-oriented Model into a Role-oriented Model

The record-oriented model does not consider which role is accessing the record. A role-oriented profile model can be derived from the record-oriented model. In the proposed role-oriented approach, each record r_i has several counters $\mathcal{F}_i^{role}$, one for each role. For example, $\mathcal{F}_i^{clinicalspecialist}$ gives a count of accesses to i^{th} record by a user in the clinical specialist role.

In the training phase, a normative profile is constructed by determining a range for the number of times records are accessed by a specific role. The profiles are constructed in a similar manner as the record-oriented approach, with the difference that in the

record-oriented profile construction approach, there is a control chart for each record, while in this approach, there is a separate record control chart for each role.

Record (r)	Patient ID	First name	Last name	Date of birth	Gender	Room	Diagnosis	...
r_1	7301	Robert	Green	26-03-1964	Male	829	Diabetes	...
r_2	7302	Melvin	Allen	11-06-1972	Male	893	Flu	...
r_3	7303	Betty	Crain	03-09-1968	Female	824	Heart Disease	...
r_4	7304	Jonathan	Moro	20-11-1996	Male	854	Leg Fracture	...
r_5	7305	Nora	Vargas	17-08-1974	Female	890	Flu	...
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
r_{18}	7318	Jennifer	Ryan	25-03-1984	Female	938	Diabetes	...
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Figure 5.2: A fragment of sample relation $\mathcal{T}$ from the Patient database.

5.4 Evaluation

As mentioned in Section 4.5, given the difficulty of obtaining real-world data, synthetic datasets are used. For evaluation of the proposed model, a synthetic data generator mimicking a health-care application was used. The health-care dataset includes roles of specialist, house officer, consultant, nurse, IT administrator, clinical specialist, and medical record clerk. The generated transactions consist of patient's record lookup queries mimicking routine lookups made by the hospital staff. A transactions generated using the data generator included single lookups queries, as well as involved multiple queries for looking up multiple attributes pertaining to a single patient's record, multiple queries for updating a patient's record, and insertion of a new record.

The generated training datasets (query logs) included a dataset for the outlier-free scenario L^{OF}, a dataset for the with-outlier scenario L^{WO}, datasets for the outlier-free and with-outlier scenario when roles are considered $L^{OF}_{RoleName}$ and $L^{WO}_{RoleName}$, respectively.

The generated outliers for the datasets were usually high or low number transactions accessing a particular record. For the purpose of notational consistency query logs are denoted as L while the dataset $\mathcal{L} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4, \ldots, \mathcal{F}_i\}$ is the set of record access frequency for each record extracted from L.

For training, the datasets were generated for 10 days in both with-outlier and outlier-free scenarios. A fragment of the queried relation $\mathcal{T}$ is shown in Figure 5.2. Record access frequencies for each record in relation $\mathcal{T}$ were extracted for each day for the duration of 10 days and were then averaged over the duration of 10 days. The training dataset $\mathcal{L}$, for evaluation, consisted of averaged record access frequencies. For example, $\mathcal{L}^{WOR}_{Consultant}$represents the averaged frequencies of record accesses by the role consultant in the with-outlier scenario.

5.4.1 Record-oriented: Outlier-free Scenario

Experiments were carried out in order to construct an example record-oriented normative profile for $\mathcal{T}$ in the outlier-free scenario. The computed center line $\mu(\mathcal{L}^{OF})$, upper limit $\mu(\mathcal{L}^{OF}) + 3\sigma(\mathcal{L}^{OF})$, and lower limit $\mu(\mathcal{L}^{OF}) - 3\sigma(\mathcal{L}^{OF})$ for $\mathcal{L}^{OF}$ are 274.7, 318.4 and 231.2, respectively. Frequent observations attacks in-terms of frequent queries were made to various records by a single role as well as in observations of a particular record made by several roles. These frequent queries were manifested in run-time log for each day and from which the record access frequency for each record for that was exacted.

The control chart in Figure 5.3 shows the run-time values of $\mathcal{F}_i$ in this scenario; only a fragment of records are shown for the purpose of demonstration. It can be seen that r_4 is accessed more than usual on day 1 and day 5, and r_3 is accessed more than usual on day 5 and 21. In the case of r_{18}, it is observed from days 1 to 15 that the number of times r_{18} is accessed above $\mu(\mathcal{L}^{OF}) + 3\sigma(\mathcal{L}^{OF})$. This persistent high record access frequencies for r_{18} are an indication that either it is an anomaly or a false positive in which case

the limits needs to be re-defined. Refined limits for r_{18} are computed by looking at the past behaviour of record for days 1 to 15 provided that this behaviour is inspected by the security officer and is concluded as a safe behaviour. The refined limits for r_{18} are $\mu(\mathcal{L}^{OF}_{record18}) = 361.6$, $\mu(\mathcal{L}^{OF}) + 3\sigma(\mathcal{L}^{OF}_{record18}) = 444.9$ and $\mu(\mathcal{L}^{OF}) - 3\sigma(\mathcal{L}^{OF}_{record18}) = 278.2$. These refined limits are used for record r_{18} from day 16 and onwards, as shown in Figure 5.3.

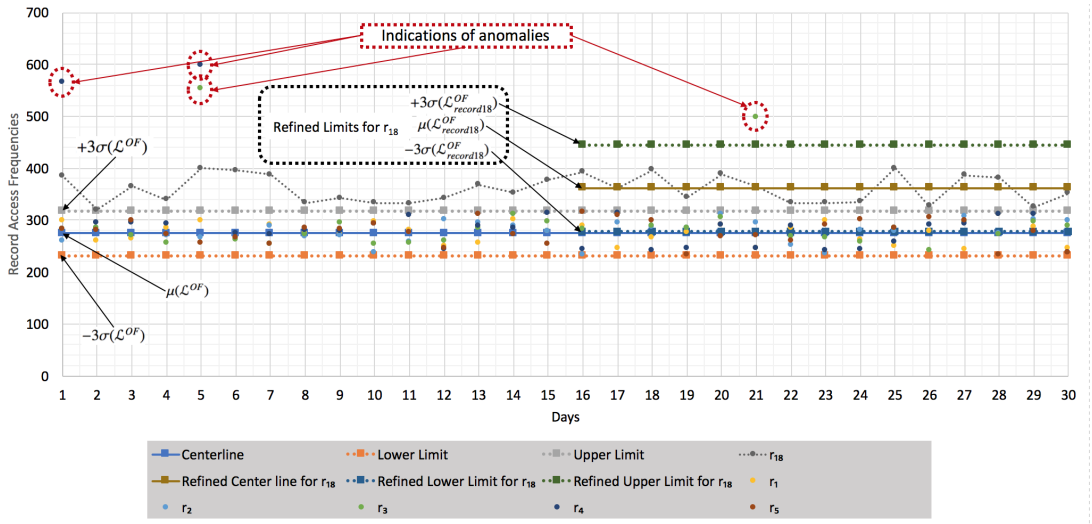


Figure 5.3: The figure shows the control chart developed during the detection phase, while the training dataset was outlier-free. The solid coloured circles (•) represents record access frequencies. The anomalies are indicated in red circles.

5.4.2 Record-oriented: With-outlier Scenario

A training dataset with outliers ($\mathcal{L}^{WO}$) was generated to evaluate the approach in with-outlier scenario. The computed center line $\mathbf{m}(\mathcal{L}^{WO})$, upper limit $\mathbf{m}(\mathcal{L}^{WO})+2MAD(\mathcal{L}^{WO})$ and lower limit $\mathbf{m}(\mathcal{L}^{WO}) - 2MAD(\mathcal{L}^{WO}))$ for $\mathcal{L}^{WO}$ are 274, 306, and 242, respectively. Malicious frequent observations were made in this scenario as well and indicated in Figure 5.4, where r_5 was frequently accessed relative to normal access on day 2, and r_2 was frequently accessed relative to normal access on day 5 and day 8.

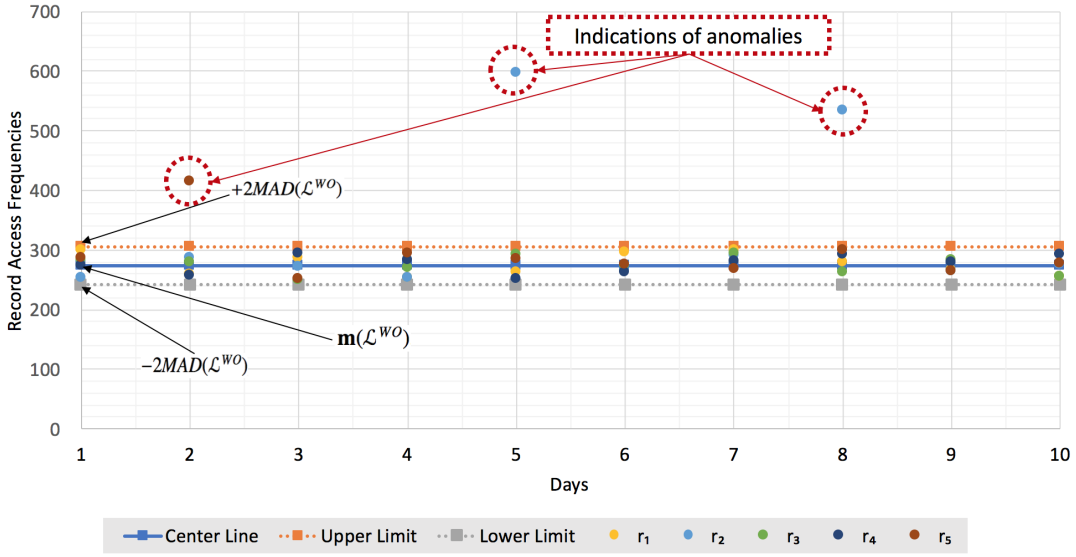


Figure 5.4: The figure shows the record access frequencies plotted for 10 days over the control chart while the training dataset is with-outlier.

5.4.3 Role-oriented: Outlier-free And With-outlier Scenario

In order to demonstrate the proposed approach in role-oriented settings in the outlier-free scenario, the role of Consultant was considered, and a training dataset $\mathcal{L}^{OF}_{Consultant}$ was generated. For role-oriented settings in with-outlier scenario, role of Nurse was considered and a $\mathcal{L}^{WO}_{Nurse}$ mimicking accesses made by the role Nurse with outliers was generated.

Malicious frequent observations were made in both scenarios. Figure 5.5 shows the control chart in role-oriented settings in the outlier-free scenario, where the access made by role consultant for 10 days is plotted where access frequencies of r_3 and r_5 on day 6 and day 8 are indications of anomalies. Figure 5.6 shows the control chart in role-oriented settings in the with-outlier scenario, where the record access frequencies for r_5 and r_1 on day 4 and day 9 are also indicated as anomalies as they are below the lower limit and are examples of oversight-anomalies. These oversight-anomalies, in essence, are indications of the absence of normative behaviours.

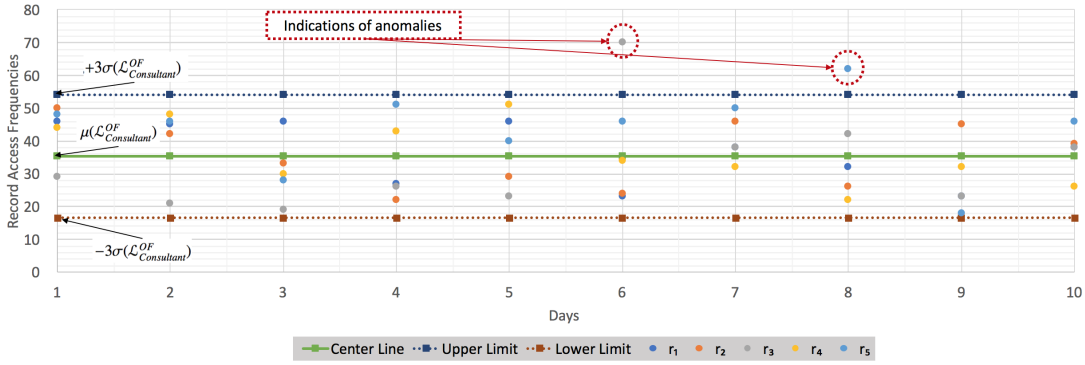


Figure 5.5: The figure shows the control chart for the role of Consultant in the outlier-free scenario. The control chart presents the frequency of record accesses by the role Consultant for the duration of 10 days.

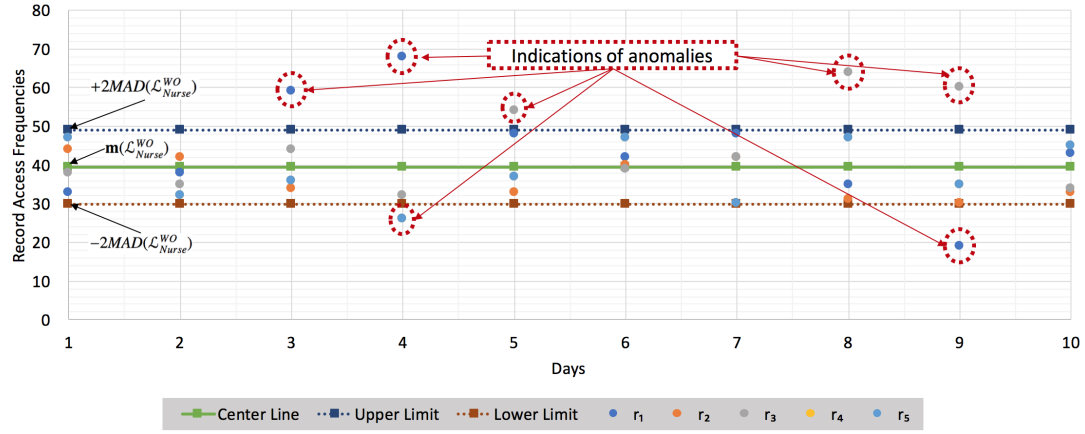


Figure 5.6: The figure shows the control chart for the role of Nurse in the with-outlier scenario.

5.4.4 Observations and Limitations

During the experimentation, it was observed that in the with-outlier scenarios, that the limits $\mathbf{m} + MAD()$ and $\mathbf{m} - MAD()$ resulted in a higher number of false positives while limits $\mathbf{m} + 2MAD()$ and $\mathbf{m} - 2MAD()$ resulted in lesser number of false positives; therefore, these limits were chosen. These limits can be fine-tuned by the user of the system.

It was observed that a control chart with mean and standard deviation was suitable when the dataset was without outliers, while the control charts with median and median absolution deviation was suitable for the case where the training data was suscep-

tible to outliers. It is demonstrated from the experiments that one can model querying behaviours from a different perspective. In comparison to existing approaches in the literature (discussed in Chapter 2) the proposed record-oriented approach offers the following advantages:

- The focus of the existing approaches in the literature is to detect malicious accesses made by a user or a role in isolation. The proposed record-oriented perspective captures collaborative behaviours of roles (or users) thus enabling the detection of frequent observation attacks - observation of the same record made by several roles (or users). These attacks may be collaborative in nature.

- Exiting approaches model behaviours in such a way that they only detect the presence of malicious behaviours as anomalies. It is demonstrated that the proposed record-oriented approach has the potential to detect normative behaviours that should have been present but are not present at run-time - oversight-anomalies. The detection of oversight anomalies are useful in scenarios such as the ones in the hospital where the presence of normative behaviour requires an immediate indication. One needs to select a time period that is appropriate for the environment where the system is being deployed. For instance, in the case of the hospital scenario, where check-up of patients is critical, it is desirable to detect the missed check-up as early as possible thus the time period, in that case, should be a short one and anything that falls immediately below the lower specification limit must be notified promptly.

- An assumption made before modelling normative behaviours, for the majority of IDSs, is the availability of outlier-free data. The proposed approach, in with-outlier scenario, has the potential to model normative behaviour in the case where the training data is susceptible to outliers.

The observations made in the above experiments are dependent on the scenarios considered, for example, the health-care system where queries are made to a patient's

records on a regular/daily basis. However, there are scenarios where queries are not made that regularly to the database, for example, a mobile customer service where only the records of those customers are accessed who have contacted mobile customer service is probably facing some issue with the mobile network. The majority of the customers of the mobile company may not have contacted or will contact the customer service. Therefore, the results may change for these type of scenarios. It is likely that the proposed approach may not be suited for every scenario where one wishes to detect malicious accesses to databases.

Another limitation is that it is unlikely that all the records are accessed with the exact same frequency: this creates room for maliciously accessing records while staying within the determined limits. This will allow an adversary, knowing the determined limits in the normative profiles, to maliciously access records and bypass the detection system.

5.5 Conclusions

The chapter introduced the record-oriented model of normative behaviour for construction of record-oriented profiles that considers data-centric features. The construction of the profiles utilizes control charts from Statistical Process Control as a way to detect anomalies. Two scenarios were considered, in the first scenario, the training data is outlier-free, and in the second scenario, the training data is susceptible to outliers.

The presented record-oriented models of normative behaviour are easy to integrate with existing systems because of their simplicity, as well as it can complement other detection systems. The proposed model enables the detection of frequent observation attacks, where these attacks can be carried out in isolation (when a single insider carries out the attack) or collectively (several insiders carry out the attack). It is possible that these insiders may not be collaborating while accessing a particular record

with malicious intentions. The experiments have demonstrated the effectiveness of the proposed approach in the detection of frequent observation attacks as well as anomalies (oversight-anomalies) introduced due to human negligence/errors (that is the case where the doctor/nurse missed the daily check-up of a patient). The proposed model can be transformed into a model for the construction of role-oriented profiles.

Chapter 6

Detecting Malicious Access to DBMS using Item-set Mining

"Fortunately, most human behaviour is learned observationally through modelling from others."

Albert Bandura (1925 – Present)

As discussed, previous chapters that organizations are diligent in how they manage, access, store, use and disseminate their application data, and leakage or misuse of this application data has severe implications for an organization in terms of financial loss and reputational damage in cases where this data consists of personal information. As the individuals (customers) may have given only limited consent to how data about them can be used or processed. Detection of insider threats is an active area of research, which has been looked at from several different perspectives [171] including the complexity of insider threats to DBMS [172]. The approach proposed in Chapter 4 considered sequences of SQL queries to model behaviours.

6.1 Introduction

Detection of insider threats is an active area of research and needs to be investigated from several different perspectives. In the previous chapters, we proposed behavioural-based approaches that enable the detection of malicious accesses made by the insiders to the databases. This chapter considers the second research question stated in Chapter 1, namely whether one can develop approaches, other than the ones described in previous chapters, to build models of querying behaviour to detect malicious accesses to the databases. In this chapter, we build a model of behaviour based on constraining the bound of the frequency of accesses to the database.

We take the position that frequently repeated behaviours can be considered normative behaviour, while rare behaviour is potentially malicious behaviour. This has also been pointed out by behaviour analysis research [173]. Therefore, we are interested in modelling frequent queries as normative behaviour while rare queries represent potential malicious behaviour.

To build models of behaviour in terms of frequency of queries, we investigate Item-set Mining in order to explore its interpretation for building behavioural profiles. Before presenting the model, Section 6.2 provides an overview of item-set mining. Section 6.3 describes the proposed model, based on item-set mining, which is evaluated in Section 6.4. Some conclusions are drawn in Section 6.5.

6.2 Item-set Mining

Item-set Mining can be categorized into frequent item-sets mining and rare item-sets mining. The aim of frequent item-sets mining it to discover frequent patterns within a dataset. A collection of items is called an item-set and which can be a singleton set. An item-set that occurs frequently is known as a frequent item-set. One of the

most common examples that are used to describe frequent item-set mining is in market basket analysis. In a scenario of a superstore that sells multiple products (items), the market basket analysis attempts to discover patterns between purchased items by customers, especially the items that are frequently purchased together from that store. This provides insight, such as a customer who buys bread, usually buys eggs as well. This helps the store decide which products should be placed close to each other on the shelves. In order to determine whether an item-set is frequent, it has to satisfy a minimum threshold value for support. Let *support* represent the number of purchase transactions in which an item-set appears compared to the total number of purchase transactions.

In the literature, there are several frequent item-set mining algorithms and the well-known ones include the Apriori algorithm [174], LCM [175, 176], Eclat [177], FP-GROWTH [178], H-Mine [179], and PrePost$^+$ [39].

In contrast to frequent item-sets mining, one can mine rare item-sets that occur infrequently. Algorithms falling under the rare item-sets mining includes AprioriInverse [177], AprioriRare [38], RP-Growth [180], and CORI [181]. In order to determine whether an item-set is rare, the number of times it occurs has to be below a specified value for *support*.

6.3 Querying Behaviour Modelled via Item-set Mining

In the context of item-set mining, the input of any item-set mining algorithm is a *transaction database*. A transaction database is a set of transactions where each transaction is a set of items. For example, continuing the example described in the previous section, a transaction would be the set of items bought together by a customer, i.e. {eggs, bread, butter}. We refer to a database of these transactions as a *query-set table* in order to distinguish it from conventional database transactions. It is important to draw this

distinction as the order of the queries in the query-set is not considered. A query-set QS_i is a set of queries (query abstractions) analogous to an item-set, while each query (query abstraction) is analogous to an item in item-set.

Given an audit log consisting of SQL queries, let a mapping function *QSmap()* map SQL queries in the audit log to the query-set table. Section 6.3.1.1 considers how this function might be constructed. An example query-set table is shown in the Table 6.1.

Table 6.1: An example of a query-set table where a query-set QS_i is analogous to an item-set, while each query is analogous to an item in item-set.

QS_i	Queries in QS_i
QS_1	{ Q_5, Q_6, Q_7 }
QS_2	{ Q_2, Q_3, Q_4 }
QS_3	{ Q_4, Q_7, Q_8, Q_9 }
QS_4	{ Q_5, Q_6, Q_7 }
QS_5	{ Q_5, Q_6, Q_7 }
QS_6	{ Q_4, Q_7, Q_9 }
QS_7	{ Q_1, Q_2, Q_3, Q_4, Q_5 }
QS_8	{ Q_4, Q_7, Q_9 }

Based on the conjecture that frequent querying behaviour is representative of normative behaviour, we mine frequent sets of queries (query abstractions), where frequent in the context of item-set mining is defined by a support value. The support value is defined as the number of times query-set appears in the query-sets table relative to the total number of rows in the transaction table. If the query-set is above the chosen minimum support it is considered to be frequent. As an example, Table 6.2 shows frequent query-sets (item-sets) mined for a chosen minimum support of 3 or 37.5%, i.e. (specified value for minimum support / total number of entries in query-set table)×100.

Table 6.3 shows all the rare query-sets with support of less than 3. However, mining all the rare query-sets below a specified support for an audit log may result in a large number of rare query-sets, effectively repetitions manifesting as subsets of a rare query-sets as seen in Table 6.3. Therefore, a way to capture all anomalies manifesting as rare query-sets and at the same time minimizing the number of mined rare query-sets is

Table 6.2: The table shows frequent query-sets mined from the query-set table shown in Table 6.1. The minimum support for the frequent query-set is set to 3 (or 37.5%).

#	Frequent Item-sets	Support
1	{ Q_9, Q_4, Q_7 }	3
2	{ Q_6, Q_5, Q_7 }	3
3	{ Q_9, Q_4 }	3
4	{ Q_9, Q_7 }	3
5	{ Q_6, Q_5 }	3
6	{ Q_6, Q_7 }	3
7	{ Q_5, Q_7 }	3
8	{ Q_4, Q_7 }	3
9	{ Q_9}	3
10	{ Q_7}	6
11	{ Q_6}	3
12	{ Q_5}	4
13	{ Q_4}	5

Table 6.3: Rare query-sets for the query-set table shown in Table 6.1. The query-sets in this table have the support of less than 3 (or 37.5%).

#	Rare Item-sets	Support
1	{ Q_8 }	1
2	{ Q_7, Q_8 }	1
3	{ Q_4, Q_8 }	1
4	{ Q_4, Q_7, Q_8 }	1
5	{ Q_8, Q_9 }	1
6	{ Q_7, Q_8, Q_9 }	1
7	{ Q_4, Q_8, Q_9 }	1
8	{ Q_4, Q_7, Q_8, Q_9 }	1
9	{ Q_1}	1
10	{ Q_1, Q_4}	1
11	{ Q_1, Q_5}	1
12	{ Q_1, Q_4, Q_5 }	1
13	{ Q_1, Q_2 }	1
14	{ Q_1, Q_2, Q_4 }	1
15	{ Q_1, Q_2, Q_5 }	1
16	{ Q_1, Q_2, Q_4, Q_5 }	1
17	{ Q_1, Q_3 }	1
18	{ Q_1, Q_3, Q_4 }	1
19	{ Q_1, Q_3, Q_5 }	1
20	{ Q_1, Q_3, Q_4, Q_5 }	1
21	{ Q_1, Q_2, Q_3 }	1
22	{ Q_1, Q_2, Q_3, Q_4 }	1
23	{ Q_1, Q_2, Q_3, Q_5 }	1
24	{ Q_1, Q_2, Q_3, Q_4, Q_5 }	1
25	{ Q_3 }	2
26	{ Q_3, Q_5 }	1
27	{ Q_3, Q_4, Q_5 }	1
28	{ Q_3, Q_4 }	2
29	{ Q_2, Q_3 }	2
30	{ Q_2, Q_3, Q_4 }	2
31	{ Q_2, Q_3, Q_5 }	1
32	{ Q_2, Q_3, Q_4, Q_5 }	1
33	{ Q_2 }	2
34	{ Q_2, Q_4 }	2
35	{ Q_2, Q_5 }	1
36	{ Q_2, Q_4, Q_5 }	1
37	{ Q_4, Q_5 }	1

required.

In the context of item-set mining, rare item-sets can be narrowed down to minimal rare item-sets or perfectly rare item-sets. An item-set is minimal if it is not among frequent item-sets and all of its proper subsets are frequent item-sets. An item-set is perfectly rare if it is not among the frequent item-sets, and all of its proper subsets are also not among the frequent item-sets.

Similarly, in the context of mining query-sets, there can be minimal rare query-sets or perfectly rare query-sets. The query-set # 37 in Table 6.3 is an example of minimal rare query-set while query-set # 3, 5, 13, 17, and 29 in Table 6.3 are examples of perfectly rare query-set.

Perfectly rare query-sets capture anomalies manifesting from the malicious group of queries made together and the subsets of (and individual) queries within the malicious group are also malicious. The minimal rare query-sets capture anomalies manifesting from a group of queries made together that forms a malicious event while the subset of (individual) queries may not be malicious themselves.

Table 6.4 shows the perfectly rare and minimal rare query-sets for the query-set table shown in Table 6.1. A total of 10 rare query-sets were discovered as a result of mining perfectly rare and minimal rare query-sets, while 37 query-sets were discovered in the case of mining all the rare query-sets. Additionally, the discovered perfectly rare and minimal rare query-sets, shown in the Table 6.4, are sufficient indication of the presence of rare (malicious) query-sets and covers the majority of the rare query-sets with lesser number of mined query-sets in the case of detecting malicious queries. For this reason, in this work, instead of mining all the rare query-sets, we focus on mining perfectly rare and minimal rare query-sets.

The next section describes how this interpretation of modelling querying behaviour translates to a behavioural-based anomaly detection system.

Table 6.4: The table shows Perfectly rare + Minimal rare query-sets mined from the query-set table shown in Table 6.1. The query-sets in this table have the support of less than 3 (or 37.5%).

#	Perfectly rare + Minimal rare query-sets	Support
1	{ Q_1 }	1
2	{ Q_2 }	2
3	{ Q_3 }	2
4	{ Q_8 }	1
5	{ Q_1, Q_2 }	1
6	{ Q_1, Q_3 }	1
7	{ Q_2, Q_3 }	2
8	{ Q_8, Q_9 }	1
9	{ Q_4, Q_5 }	1
10	{ Q_1, Q_2, Q_3 }	1

6.3.1 Query-set Mining-based Malicious Query Detection System

Similar to conventional anomaly detection systems, the proposed approach has a training phase and a detection phase. In the training phase, the profile of normative behaviour is constructed from the DBMS audit logs. However, the detection phase of the proposed approach is different from the detection phase of conventional anomaly detection systems. While comparing the normative profile with the run-time profile, the detection phase also mines rare querying behaviour from run-time DBMS audit logs. Any instance of deviation from a normative profile and any mined infrequent (rare) query-sets are labelled as anomalies. Figure 6.1 depicts the detection phase of the proposed anomaly detection system.

6.3.1.1 Constructing Behavioural Profiles using Item-set Mining

Deployed SQL query abstraction The deployed specialization of SQL query abstraction, as shown in Table 4.1 is used. For simplicity, we refer to sets of query abstraction and sets of queries both as query-sets.

Mining Frequent Query-sets *PrePost*$^+$ [39], claimed to be one of the fastest algo-

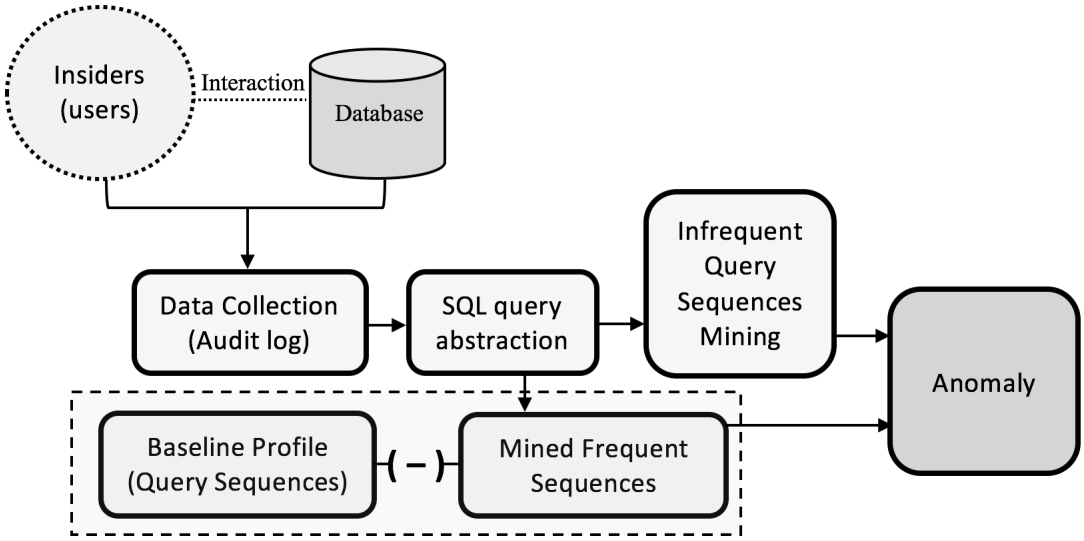


Figure 6.1: Detection Phase: the first step is data collection; the second step is of query abstraction, followed by run-time profile construction. The run-time profile is compared against the normative profile (baseline profile), and infrequent query-sets are mined.

rithms, is used to generate behavioural profiles.

An SQL query abstraction corresponds to an item in an item-set (query-set). Let $QSmap(L) = QST_L$ be the mapping from SQL queries in audit log L to query-sets table QST_L. For this work, for ease of exposition, we use a naïve mapping function that splits the queries in L into a table of equal-sized groups of queries where each group maps to one query-set in QST_L. It was observed during the experimentation that a query-set of size between 7 to 10 was sufficient to have a query-sets table to discover malicious query-sets. A transaction size outside this range resulted in an inaccurate representation of querying behaviour. However, a more fine-grained approach would be to identify transactions from the audit log and then transform them into a query-sets table. As future work, we are interested in investigating the application of the general-purpose framework, proposed in [182] for inferring transaction like patterns from logs to determine $QSmap(L)$ and construct a query-set table. Let $PrePostPlus(Abs(QSmap(L))$ be the set of frequent query-sets that are mined from $Abs(QSmap(L))$. Given an audit log free from attacks L_{norm} the baseline profile is

denoted as $\beta_{\mathcal{N}} = PrePostPlus(Abs(QSmap(L_{norm})))$.

The generated logs used in this Chapter consisted of transactions having a size between 3 to 7 thus a query-set size of greater than the size of the transactions, i.e. 10, was selected. The query-set size to construct a query-set table is dependent on the underlying data; therefore, it is to be selected based on the size of transactions where the system is to be deployed. Ideally having a mapping function that maps each query-set in the query-set table to each transaction would be desirable. However, different transactions having different sizes in the logs adds an additional layer of complexity, and in the case of millions of transactions per day, it may result in computational overhead.

6.3.1.2 Comparing Frequent Query Profiles

In the detection phase, a run-time profile is constructed that is then compared with the baseline profile $\beta_{\mathcal{N}}$. Given an audit log collected at run-time L_{run} the run-time profile is denoted as $\beta_{\mathcal{R}} = PrePostPlus(Abs(QSmap(L_{run})))$. The set of anomalies (mismatches) when the baseline profile and the run-time profile are compared, is given by $S^{miss}_{\beta_{\mathcal{R}},\beta_{\mathcal{N}}} = \beta_{\mathcal{R}} - \beta_{\mathcal{N}}$. The comparison between $\beta_{\mathcal{R}}$ and $\beta_{\mathcal{N}}$ detects attacks manifesting themselves in frequent DBMS accesses that are considered not normal.

6.3.1.3 Mining Rare Query-sets

We use the algorithim AprioriInverse [177] to mine perfectly rare query-sets and AprioriRare [38] to mine minimal rare item-sets. Mining all the rare query-sets results in a large number of query-sets and most of them are just repetitions in terms of subsets of mined rare query-sets. That is one malicious sequence results in multiple rare-query sets; however, only one query-set suffice as an indication of that malicious sequence. During evaluation we discovered that the minimal rare (AprioriRare) and perfectly rare (AprioriInverse) query-sets were a suitable approximate cover for the malicious rare queries.

Given the run-time audit log L_{run} the set of rare item-sets is denoted as $Z_{rare}^{L_{run}} = AprioriInverse(Abs(QSmap(L_{run}))) \cup AprioriRare(Abs(QSmap(L_{run})))$.

6.3.1.4 Selecting a Threshold Value For Support

In the context of item-set mining, the usual practice is that the user defines the threshold value for support [183]. However, in the case of an anomaly detection system based on item-set mining, an approximate value for support can be learned from past 'safe' logs. Mining past 'safe' logs with a threshold value of 0% will result in all the query-sets along with their support. Using this information one can determine the normative value for support, for instance, if a majority of the discovered query-sets have support equal to or more than 40% in the safe log then 40% can be used as a threshold value in the detection phase. A learning-based threshold value can be an extension of the proposed approach. In this chapter, for clarity, we keep the value for support user-defined.

6.4 Evaluation

In order to evaluate the proposed model in this chapter, the banking-style application, described in Section 4.5.1, is used to generate test datasets. The application system was executed multiple times, once to collect audit logs to construct baseline profile L_{norm} and multiple times to construct malicious query sequences similar to those generated in Chapter 4: malicious modification of records, malicious observation of records, or malicious deletion of records from the database. Malicious observation of a record pertaining to an individual is considered a form of a violation of an individual's privacy.

Malicious sequences of queries were inserted into the attack logs L_{attack_1}, L_{attack_2}, and L_{attack_3} at random locations. Table 6.5 shows the length of the attack logs, the length of the rare query sequences and the number of times the rare query sequence was inserted in the attack logs. An example of malicious sequences of queries included queries

accessing the multiple records of individuals. The length of the query sequence represents that the number queries the malicious insider executed, for instance, to browse through the records on individuals in the database. The inserted instances were the number of instances when the malicious insider carried out the malicious query sequence where each instance had different queries, for example, L_{attack_1} was inserted with two instances of malicious queries, both the instances had the same length but were consisted of different queries. The idea behind having multiple instances of attacks, inserted at a different location in the logs, was to evaluate the approach more rigorously, for instance, to detect that the attack is detected not based on a SQL query in isolation instead based on that malicious sequence.

Table 6.5: The table showing the make up of attacks and attack logs in the case of mining rare query-sets.

i	Attack Log	Size of Attack Log	Malicious Sequence Length	# of Inserted Instances of Malicious Sequence
1	L_{attack_1}	1364	6	2
2	L_{attack_2}	1035	2	1
3	L_{attack_3}	1337	4	4

For the experiments, we used a naïve *QSmap()* function that maps the query logs to the query-set table, such that it splits the queries in logs into equal-sized (size chosen arbitrarily 10) groups of queries where each group maps to one query-set in the query-set table. For experimentation, the threshold for support was set to 40%, that is, query-sets having support 40% and above are considered as frequent while query-sets having support less than 40% are considered as rare. However, as mentioned above the threshold value of support can be learned instead of user-defined. The choice of selection of a threshold for support is left to the discretion of the user of the system.

Table 6.6 shows the number of mined rare malicious query-sets that are an indication of anomalies. In the experiments, it was discovered that all of the inserted malicious sequences were depicted in the mined rare query-sets. Each query-set had one of the malicious queries from the malicious query sequence in the mined rare query-sets.

The results are an indication that our initial conjecture, that is, frequently repeated behaviour can be considered normative behaviour, while rare behaviour is potentially malicious behaviour, holds when it is interpreted in the scenario of modelling querying behaviour for detecting malicious queries. However, it is not necessarily the case that the conjecture always holds in the scenario of modelling querying behaviour that we discuss in the next section.

Table 6.6: The number of mined rare query-sets that are an indication of anomalies.

i	Attack Logs	$\lvert Z_{rare}^{L_{attack_i}} \rvert$
1	L_{attack_1}	40
2	L_{attack_2}	3
3	L_{attack_3}	15

6.4.1 Mimicry Attacks: Frequent Queries as Malicious Behaviour

An aware malicious insider can repeat the malicious sequence enough times to exceed the threshold value set for support and thus bypasses the anomaly detection system. This, in essence, is a type of mimicry attack as the adversary mimics the frequency of normal queries. The conjecture that frequently repeated behaviour can be considered normative behaviour does not hold in the case of mimicry attack. However, the frequent query-sets are mined from anomaly free logs. Therefore, even if the adversary repeats the malicious query sequence, the mined frequent query-sets from run-time logs that are representatives of those malicious repeated query sequence will result in mismatches when the run-time profile is compared to the normative profile.

For evaluation we considered three attack scenarios where three groups of repeated malicious sequences of queries were executed, and the resultant query sequences were made part of the attack logs L_{attack_4}, L_{attack_5}, and L_{attack_6}. Profiles were constructed corresponding to L_{norm}, L_{attack_4}, L_{attack_5}, and L_{attack_6}, that is, β_{N}, β_{attack_4}, β_{attack_5}, and β_{attack_6}.

Table 6.7 shows the length of the attack logs, the length of the rare query sequences and the number of times the frequent query sequence was inserted in the attack logs.

Table 6.7: The table shows the length of the attack logs, the length of the query sequences and the number of times the rare query sequence was inserted in the attack logs.

i	Attack Log	Size of Attack Log	Malicious Sequence Length	# of Inserted Instances of Malicious Sequence
4	L_{attack_4}	1366	13	113
5	L_{attack_5}	1431	8	127
6	L_{attack_6}	1253	7	140

Table 6.8 shows number of mismatched query-sets when β_N was compared with attack profiles. The inserted repeated malicious sequences were represented by the mismatched query-sets and were indication of anomalies. Each query-set had one of the malicious queries from the malicious query sequence in the mined mismatched frequent query-sets.

Table 6.8: The number of mismatches when the normative profile is compared with attack profiles.

i	Attack Profile	$\lvert S^{miss}_{\beta_{attack_i},\beta_N} \rvert$
4	β_{attack_4}	139
5	β_{attack_5}	124
6	β_{attack_6}	96

The number of discovered rare query-sets, shown in Table 6.6, can be referred to as mismatches. The number of mismatches (and discovered rare query-sets) shown in Table 6.8 and Table 6.6 are indication of deviation from normative behaviour. During the experimentation, we observed that all mismatches were an indication of both malicious rare and frequent queries. The number of mismatches are quite high as compared to the number of inserted malicious group (rare and frequent) of queries. However, the high number of mismatches for small number of malicious groups of queries may lead to the wrong perception that the audit logs might have a large number of malicious queries.

6.4.2 Frequent, Rare, and In-between

In this chapter, we have used a common threshold support value, so query-sets are either frequent or rare. The question then arises: why not use two different threshold values for support, one for mining frequent query-sets and other for mining rare query-sets? For example, query-sets having support value of and above 60% are considered as frequent while query-sets having 20% or less are considered as rare. The scope of the chapter was to test our conjecture; therefore, the experiments were setup in a way to demonstrate whether one can model rare queries as malicious behaviour. Given this, it was important to generate data with a support less than a synthetic cut-off point, which is irrelevant in this scenario as the aim was to test whether or not the conjecture holds.

Moreover, having two different threshold support values for rare (e.g. 20%) and frequent (e.g. 60%) query-set may leave room for an aware adversary to carry out attacks that may bypass the anomaly detection system. A malicious adversary can make malicious queries enough times that the representative query-sets of those malicious sequences may well fall in-between the rare and frequent support values (>20% and 60%<), thus bypassing the anomaly detection system.

6.4.3 Comparison with N-gram Approach

A query-set is an unordered set of distinct query abstractions. The item-set mining-based approach ignores the order of queries while mining the query-sets. The n-gram approach in Chapter 4 builds a model of behaviour in terms of query sequences. In the n-gram approach, the temporal order is important and requires that the event must be contiguous; that is, the events must be next to each other. Based on this temporal order, the n-gram approach captures the short-range correlation between events. In the case of the item-sets mining-based approach, the correlation can be termed as a long-range

correlation but is bounded by the length of query-sets in the query-sets table which in our case was chosen to be 10.

Another aspect of the comparison between both the approaches is that in the case of the item-sets mining approach, the number of mismatches that were an indication of malicious queries was high as compared to the mismatches flagged by the n-gram approach. In principle, the number of mismatches for a single malicious query in case of n-gram approach is equal to the size of n-gram, while in the case of item-sets mining approach the mismatches depends on the algorithm that is used to mine query-sets and are in large number as depicted in Table 6.8 and Table 6.6.

6.4.4 Complexity of Item-set Mining Approach

The complexity of the item-set mining approach is mainly rooted in the complexity of deployed item-set mining algorithms. The training phase consists of the algorithm for *QSmap()* function and the item-set mining algorithm PrePost$^+$. The algorithm for *QSmap()* function has a linear complexity. PrePost$^+$ is claimed to be one of the fastest frequent-item set mining algorithms, and its complexity is discussed in [184]. In the detection phase, we have an algorithm for comparing profiles with the linear complexity along with the algorithms for mining rare query-sets, AprioriRare and AprioriInverse.

6.4.5 Potential Application

Enterprise database systems typically produce a large number of logs, for instance, it is reported that within a 19 hour time frame, approximately 17 million SQL queries were made in a major US bank [9]. Manual inspection of an entire set of logs requires a huge amount of resources in terms of time, and associated cost, therefore, considered impractical. A practical approach in industry for performing an audit for the purpose

of detecting anomalies using logs is to audit random samples of audit logs instead of the entire audit log. The drawback of auditing only random samples is that it is possible that the random selection of samples may miss those samples that have traces of malicious accesses, thus significantly reducing the effectiveness of the random sample inspection approach. Approaches that enables one to audit an entire set of logs in an automated way with a reasonable time of auditing and its associated cost would be a good alternative to the random sampling approach. Item-set mining algorithms have been shown to scale in the literature [185]. For example, in [186], it is used to process transactions ranging from up to 100 million. The proposed approach in this chapter is a suitable candidate, for a reason being it built upon existing off-the-shelf machine learning algorithm, requires minimum manual interference, applied to entire sets of logs in an automated way, and with a reasonable processing time as item-set mining algorithms are quite fast.

6.4.6 Sequential Pattern Mining

We saw in previous sections that item-sets mining ignores the ordering of the items within the sets. Retaining the order in which the queries were made while mining the audit logs may have the potential to construct a more accurate model of behaviour while minimizing the scope for sophisticated mimicry attacks. Another aspect of considering the order between the queries is that it may possibly reduce the numbers of mined rare sequences and the number of mismatches shows in the Table 6.6 and Table 6.8, respectively.

A realted domain to item-set mining within data mining, known as sequential pattern mining, considers ordering between items [187]. Sequential pattern mining discovers sub-sequences in a set of sequences, and this discovery is typically based on the frequency of occurrence of the sub-sequences.

A numbers of algorithms exist for mining frequent sequential patterns, including, Pre-

fixSpan [188], GSP [189], SPADE [190], LAPIN [191], FAST [192], ClaSP [193], BIDE+ [194], and MaxSP [195]. However, little attention has been paid on algorithms for mining rare sequential patterns. Recently two algorithms for mining rare sequential patterns [196, 197] have been proposed in the literature. Interpretation of sequential pattern mining in the context of modelling querying behaviour is another area for further exploration.

6.5 Conclusions

This chapter investigates whether one can develop approaches, other than the ones presented in previous chapters, to build models of querying behaviour to detect malicious accesses to the databases. In attempting to answer this question, this chapter presented an interpretation of item-sets mining for modelling querying behaviour. The interpretation resulted in an anomaly detection system and is based on the conjecture that frequent queries represent normative behaviour, while rare queries represent potentially malicious behaviour. In order to realize the anomaly detection system based on item-set mining, algorithms like *PrePost*$^+$ for frequent, and *AprioriInverse* and *AprioriRare* for mining rare query-sets were used.

The evaluation demonstrated that it is possible to detect a malicious group of queries (rare or frequent) with the proposed item-sets mining-based approach.

As future work, we plan to use the technique proposed in [182] to infers transaction like repetitive patterns from logs to construct a query-set table. Additionally, we plan to explore the interpretation of sequential pattern mining in the context of modelling querying behaviours.

Chapter 7

Privacy Interpretation of Behavioural-based Approaches

"It's dangerous when people are willing to give up their privacy."

Noam Chomsky(1982 - Present)

7.1 Introduction

It was demonstrated in Chapter 4, that the n-gram based anomaly detection system was effective in detecting anomalies. Looking closely at the nature of some of the anomalies, in particular, the malicious data observation attacks, they are a type of privacy violations. The behavioural-based detection approaches presented in previous chapters, including, the n-gram, the semantic and the item-sets mining-based approach demonstrate the detection of privacy violations. This raises the question as to whether

these anomalies are true privacy anomalies or whether they are simply a form of security/access control anomalies, which we know these techniques can detect. In order to answer the question, a privacy-based semantics for anomaly detection is needed. This chapter considers the third key research question stated in Chapter 1: whether behavioural-based anomaly detection approaches have a privacy semantic interpretation and the detected privacy anomalies can be related to the conventional definitions of privacy semantics. In order to distinguish them from conventional security anomalies, these violations are referred to as *privacy-anomalies*.

In order to answer the above-mentioned questions, the notion of *'Privacy-Anomaly Detection'*(*PAD*) is introduced in this chapter. PAD learns privacy criteria from past interactions (audit logs) and uses this criteria to check whether the current behaviour is different from past behaviour with respect to privacy. The PAD architecture falls within an interactive query system setting for microdata release.

We describe a naïve instantiation of PAD using *k*-anonymity privacy criteria which we refer to as (*k*-anonymity)-PAD. A study is carried out to investigate whether a security-anomaly detection system, in particular, the n-gram approach presented in Chapter 4, can detect these (k-anonymity)-PAD privacy-anomalies. Additionally, in order to demonstrate that a PAD system can be realized with other privacy metrics (or combination of privacy metrics), an instantiation based on the composition of *k*-anonymity and a more recent privacy metric – *Discrimination Rate Privacy Metric* – is presented. This instantiation is referred to as (*k*-anonymity, DR)-PAD. The *k*-anonymity and discrimination rate based instantiation of privacy-anomaly detection system can be termed as a semantic instantiation of PAD as it captures the semantics as it learns the criteria from distribution of attribute values in the database.

In this chapter, we show that PAD-based interactive mechanisms are vulnerable to privacy attacks. We further investigate: whether these types of privacy attacks (inferences) can potentially manifest themselves as anomalies and whether one can interpret

a security-anomaly detection system in such a way that it can detect privacy attacks as privacy-anomalies. We present the result that privacy attacks (like inferences) can be detected by applying security-anomaly detection system over the logs of interactive querying mechanisms on the basis of a PAD interpretation.

The rest of this chapter is organized as follows. Section 7.2 presents a design of a privacy-anomaly detection system and an instantiation based on *k*-anonymity. Section 7.3 investigates whether there is a correlation between privacy and security anomalies. Section 7.4 introduces an interpretation of a privacy-anomaly that is based on the composition of *k*-anonymity and a privacy metric known as *discrimination rate*. Section 7.5 considers some privacy attacks on these mechanisms. Section 7.6 presents an application of security-anomaly detection system to detect (unknown) privacy attacks as privacy-anomalies. Section 7.7 concludes this chapter.

7.2 Privacy-Anomaly Detection (PAD) System

In this section, we introduce the notion of privacy-anomaly detection and present a naïve instantiation of it based on *k*-anonymity. We argue that this naïve instantiation constitutes the basis for a more advanced form of a privacy-anomaly detection system, analogous with *k*-anonymity which constitutes the basis for more sophisticated formal privacy definitions. The reasons are as follows. Firstly, this is an exploratory study to consider the questions discussed in Section 7.1; therefore, using a well-understood privacy model like *k*-anonymity enables better understanding of the subject being explored and helps to avoid underlying complexities associated with other more complex privacy definitions. Secondly, *k*-anonymity served as a foundation of many subsequent formal privacy definitions, which is a good indicator of the applicability of this study onto other privacy definitions.

Table 7.1: A fragment of relation `temp_table`.

age	zipcode	county	gender	salary
>55	989234	Cork	Male	60K
>55	989234	Cork	Male	92K
>55	989234	Cork	Male	77K
>45	839523	Cork	Male	50K
>35	839777	Dublin	Male	60K
>35	839777	Dublin	Male	63K
>35	839777	Dublin	Male	85K
>35	839777	Dublin	Male	70K
>35	839777	Dublin	Male	60K
>50	839567	Cork	Female	72K
>50	839567	Dublin	Female	62K
>50	839567	Cork	Female	92K
>50	839567	Dublin	Female	77K
>50	839567	Cork	Female	68K

7.2.1 A *k*-Anonymity based Privacy-profile

In the proposed model k-anonymity is used to specify a privacy limit $[\![k\ ,\ q]\!]$, whereby k individual must share the same quasi identifier q values in the result of a query. Intuitively, this means for that particular response, for a sufficient value of k, an adversary can only narrow down to k individuals. In the case where an adversary has a secondary dataset with overlapping quasi-identifier values, then the query response can be linked to k different individuals, therefore minimizing the risk of re-identification. In the model the privacy-profile is defined as a set of privacy limits. In terms of privacy, each privacy limit means that in a particular instance of a query response an adversary won't be able to distinguish an individual's quasi-identifier values from k individuals for the set of quasi-identifiers that appeared in the query response.

Consider a relation `temp_table`, as shown in Table 7.1, having several attributes including a sensitive attribute `salary`, and quasi-identifiers `age`, `gender`, `zipcode`, and `county`. For ease of exposition we assume the values for attribute `age` are aggregated into age ranges, for instance, all the values for attribute `age` above 55 are represented as >55. Given a mined privacy limit $[\![3, \{\texttt{age}, \texttt{zipcode}\}]\!]$, in privacy-profile,

Table 7.2: A relation $\mathcal{T}_{R1}$ resulting from the query `SELECT age, zipcode FROM temp_table WHERE gender = 'Male';`.

age	zipcode	salary
>55	989234	60K
>55	989234	92K
>55	989234	77K

then the response to the analyst query `SELECT age, zipcode FROM temp_table WHERE gender = 'Male' AND county = 'Cork' AND age > 55;` as shown in Table 7.2 is not anomalous since the value of k for the the quasi-identifiers {`age`, `zipcode`} in the response is greater than or equal to 3.

7.2.1.1 Mining k-anonymity based Profiles for PAD

The privacy-anomaly detection consists of two phases, similar to traditional anomaly detection approaches, that are, learning phase and a detection phase. The instances of the privacy model are mined from audit logs in order to generate privacy-profiles. We refer to a privacy-profile that is mined from past logs in the learning phase as a normative privacy-profile. The idea is to learn the k values for sets of quasi-identifier(s) by mining past audit logs and interpret those mined 'privacy limits' as 'normal'.

Given an audit log L^*, consisting of query responses (L in previous chapters represented logs of queries), $Pri(L^*)$ gives a privacy-profile consisting of privacy limits mined from log L^*, where $q \in QI$ represent a set of quasi-identifier. A normative privacy-profile is generated from an anomaly-free past log L^*_{norm} and is denoted by $Pri(L^*_{norm}) = \{ [\![k_1, q_1]\!], [\![k_2, q_2]\!], \ldots, [\![k_m, q_m]\!] \}$. For example, consider the relation $\mathcal{T}_{R2}$ shown in Table 7.3, the mined value of k for the set of quasi-identifiers {`age, zipcode, county`} is 4, that is, $[\![4, \{age, zipcode, county\}]\!] \in Pri(L^*_{norm})$. In essence we are constructing privacy limit (L^*, q) which returns k as a limit to the privacy in the table for a given q. The normative privacy-profile is effectively a set of these privacy limits mined against the logs for a given set of quasi-identifiers. Intuitively, the tuples

Table 7.3: A relation $\mathcal{T}_{R2}$ resulting from the query `SELECT age, zipcode, county FROM temp_table WHERE gender = 'male';`.

age	zipcode	county	salary
>55	839523	Cork	60K
>55	839523	Cork	92K
>55	839523	Cork	77K
>45	839523	Cork	50K
>35	839777	Dublin	60K
>35	839777	Dublin	63K
>35	839777	Dublin	85K
>35	839777	Dublin	70K
>35	839777	Dublin	60K

in the normative privacy-profile shows to what extent one narrows down to individuals records in normative settings.

7.2.1.2 Detecting Privacy-anomalies

The detection phase, in terms of privacy, checks if an adversary is able to narrow down to fewer than k individuals for a given set of quasi-identifiers in the normative profile. In the instance, where the adversary is able to narrow down to fewer than specified k individuals for a given set of quasi-identifier then this instance is labelled as a privacy-anomaly and poses higher risk of re-identification relative to normal. During the detection phase, the run-time profile $Pri(L^*_{run})$ constructed given a run-time log L^*_{run}. $Pri(L^*_{run})$ is the constructed run-time profile. Given privacy limits $[\![k_i, q_i]\!]$ and $[\![k_j, q_j]\!]$ then $[\![k_i, q_i]\!]$ subsumes $[\![k_j, q_j]\!]$ (denoted $[\![k_i, q_i]\!] \preceq [\![k_j, q_j]\!]$) if imposing privacy limit $[\![k_j, q_j]\!]$ instead of $[\![k_i, q_i]\!]$ leads to no additional loss of privacy. Formally,

$$[\![k_i, q_i]\!] \preceq [\![k_j, q_j]\!] \equiv q_i \subseteq q_j \wedge k_j \geq k_i$$

In the case where $[\![k_i, q_i]\!] \in Pri(L^*_{norm})$ and $[\![k_j, q_j]\!] \in Pri(L^*_{run})$ then $[\![k_i, q_i]\!] \preceq [\![k_j, q_j]\!]$ means that $[\![k_j, q_j]\!]$ can be safely replaced by $[\![k_i, q_i]\!]$ without any loss of privacy. If

Table 7.4: A relation $\mathcal{T}_{R3}$ resulting from the query `SELECT age, zipcode FROM temp_table WHERE gender = 'female';`.

age	zipcode	salary
>50	839567	72K
>50	839567	62K
>50	839567	92K
>50	839567	77K
>50	839567	68K

a privacy limit subsumes another intuitively it means if the subsumed privacy limit is replaced by the one that subsumes it then there is no loss of privacy.

Consider the response of a query at run-time shown in Table 7.4, and that there exists a privacy limit $[\![3, \{age, zipcode\}]\!]$ in $Pri(L^*_{norm})$. The mined value k of the set of quasi-identifier $\{age, zipcode\}$ is greater than 3 therefore this privacy limit $[\![5, \{age, zipcode\}]\!]$ in $Pri(L^*_{run})$ is considered to be subsumed by the privacy limit $[\![3, \{age, zipcode\}]\!]$ in $Pri(L^*_{norm})$. In terms of privacy, it means given that this instance of query response an adversary can narrow down so many individuals as one normally is able to for a given set of quasi-identifiers. This naïve instantiation acts as a stepping stone to describe later investigations on privacy-based interpretations of behavioural-based approaches presented in this chapter.

7.2.2 Computational Complexity

The algorithm for profile generation in the training phase of privacy anomaly detection system based on k-anonymity has linear complexity. The detection phase consists of an algorithm to run-time profile generation and an algorithm for matching privacy limits with both the algorithms having linear complexity. The overall complexity of privacy anomaly detection system based on k-anonymity is O(n).

7.3 Security-anomaly Detection System Detecting Privacy-anomalies

This section explores whether privacy-anomalies (as identified by the model in this chapter) are also identified as security-anomalies (as identified by the model in Chapter 4). We consider a variation of the hospital dataset, a fragment of the dataset is shown in Table 7.5. Logs were generated for construction of a normative profile and another for the construction of a run-time profile. The training logs (anomaly-free) for the n-gram based approach are denoted by L^{hosp}_{norm}, while the anomalous run-time logs for the hospital datasets are denoted by L^{hosp}_{run}.

To construct normative and run-time profiles using the n-gram model, selection of an appropriate value of the size of n-gram was desirable for the hospital dataset. To select an appropriate size of an n-gram in this scenario, test logs L^{hosp}_{test1} and L^{hosp}_{test2} were generated in a safe environment (anomaly-free). N-gram profiles were constructed with varying n-gram size, that are, $ngram(Abs(L^{hosp}_{test1}), n)$ and $ngram(Abs(L^{hosp}_{test2}), n)$, and generated profiles were compared. Figure 7.1 depicts the number of n-gram mismatches arising when comparing the normal test $ngram(Abs(L^{hosp}_{test1}), n)$ and $ngram(Abs(L^{hosp}_{test2}), n)$, for different values of n. Experiments along the lines, as shown in Section 4.5.2 were carried out to determine the suitable value for n. From the experiments, the n-gram of the size of 4 ($n = 4$) was considered optimal as it resulted in an acceptable number of mismatches.

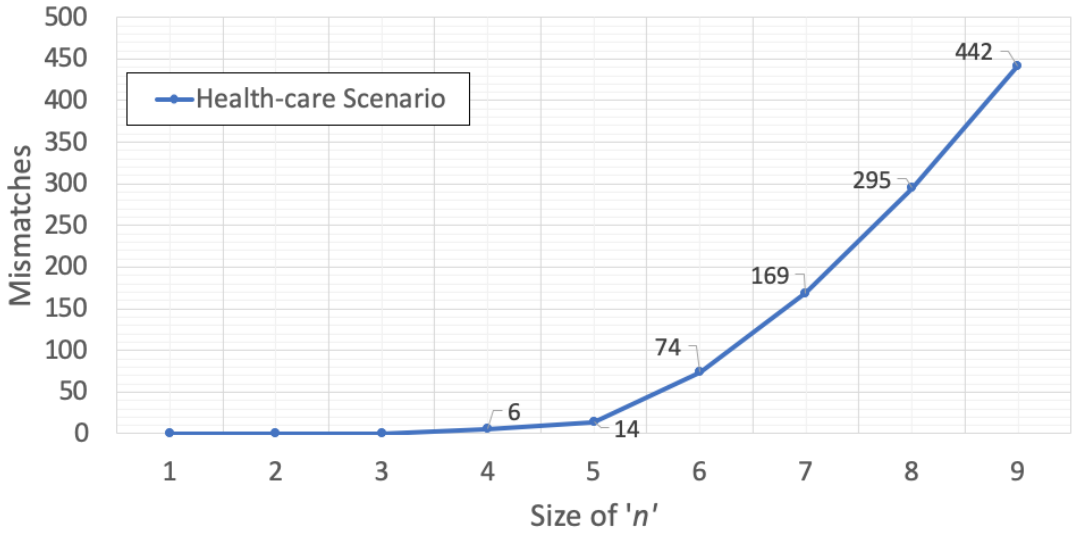


Figure 7.1: The figure shows the number of mismatches between $ngram(Abs(L^{hosp}_{test1}), n)$ and $ngram(Abs(L^{hosp}_{test2}), n)$ for different values of n.

Once the value of n was decided upon, the normative and run-time profiles were constructed for the experiments. Given the training logs L^{hosp}_{norm} and L^{hosp}_{run} n-gram profiles were constructed such that $ngram(Abs(L^{hosp}_{norm}), 4)$ and $ngram(Abs(L^{hosp}_{run}), 4)$, and subsequently the normative and runtime profiles were compared.

The same queries in logs L^{hosp}_{norm} and L^{hosp}_{run} were executed in the presence of the privacy-anomaly detection system (described in Section 7.2) resulting in logs of query responses L^{hosp*}_{norm} and L^{hosp*}_{run}. Subsequently, a normative privacy-profile $Pri(L^{hosp*}_{norm})$ and a run-time $Pri(L^{hosp*}_{run})$ profiles were constructed and compared.

The attribute `patient_ID` and `e-mail_ID` were considered as a unique identifier, the attribute `diagnosis` was considered as a sensitive attribute while the rest of the attributes including `first_name`, `last_name`, `status`, `dob`, `gender`, `city`, and `marital_status` were considered as quasi-identifiers. For the experimentation, two categories of privacy-anomalies were injected as described in Table 7.6. Using this anomaly-containing run-time log, from 15 privacy-anomalies 13 were detected by the n-gram based security-anomaly detection system proposed in Chapter 4 and the privacy-anomaly detection system proposed in this chapter.

Table 7.5: A fragment of hospital dataset.The strike-through attribute values represents a deleted row.

dob	city	gender	diagnoses	country	...
1981	Dublin	Male	Flu	Ireland	...
1981	Dublin	Male	Flu	Ireland	...
1981	Dublin	Male	Diarrhoea	Germany	...
1920	Cork	Male	Heart Disease	Ireland	...
1981	Galway	Female	Acne	Ireland	...
1984	Galway	Male	Flu	Spain	...
1984	Galway	Male	Diabetes	Ireland	...
1984	Galway	Male	Hypertension	Ireland	...
1984	Galway	Male	Leg Fracture	France	...
...	...	...	...	...	...
...	...	...	...	...	...
~~1981~~	~~Dublin~~	~~Male~~	~~Flu~~	~~Germany~~	...

Table 7.6: Description of Privacy-anomalies injected.

Description of privacy-anomalies	Number of anomalies injected
Addition of one or more attributes to the base relation shown in Table 7.5. For instance, a new attribute, like `country`, was inserted in the relation and queries were made to retrieve this attribute values.	5
Update or Deletion of records from relation shown in Table 7.5	10

7.3.1 Detected Privacy-anomalies

The n-gram based security-anomaly detection system detected all those privacy-anomalies that were generated by injecting one more attribute into the relation. The privacy-anomalies injected by adding one more attribute were identified as privacy-anomalies by both systems. The reason that they were identified was because there were no n-gram that contained a reference to new attribute in its query abstraction.

One of the detected privacy-anomalies corresponds to the query shown below.

```
SELECT diagnoses, dob, city, country
FROM hospitalDB
```

Table 7.7: Response to a undetected privacy-anomalous query.

dob	city	diagnoses
1920	Cork	Heart Disease

```
WHERE dob = '1981'
 AND city = 'Dublin';
```

The normative privacy-profile contains no privacy limit reference to the new (or combination of new) attribute.

7.3.2 Undetected Privacy-anomalies

A privacy-anomaly undetected by the n-gram based approach but detected by the privacy model is:

```
SELECT dob, city, diagnoses
FROM hospitalDB
WHERE dob = '1920'
 AND city = 'Cork' ;
```

The query returns a relation with one record as shown in Table 7.7. It is identified as a privacy-anomaly by the privacy model for the reason being that the specified value of k for the specified set of quasi-identifier meant that an adversary was able to single out an individual. This anomaly is undetected by n-gram based security-anomaly detection approach because there was an n-gram in normative profile contained a reference to this query abstraction.

A similar set of experiments were carried out in the banking scenario and results along the same lines as of hospital scenario were obtained.

In the examples above, the privacy-anomalies illustrated are based on a single query rather than a query sequence. In Section 7.5 we consider privacy-anomalies that are based on query correlations, where sequences of queries in isolation are safe, but taken

together as a sequence result in a privacy-anomaly.

7.3.3 Identifying Appropriate Privacy Limits

In order to find the appropriate values of k, in the mining process, in theory, all the combinations of quasi-identifiers need to be considered. This, in essence, is a combinatorial explosion, especially in the case of a large number of quasi-identifiers. Additionally, one may discover either very large or very small values of k in practice for certain combinations of quasi-identifiers. Therefore, in order to discover reasonable values of k, one may define a range while mining the values of k such that the values falling within the range and their corresponding combinations of quasi-identifiers are considered for privacy-profiles.

7.4 *k*-anonymity and Discrimination based Privacy-anomaly Detection System

For ease of exposition, k-anonymity was picked as a foundation for the study and we argue that our approach can be adapted to other definitions of privacy. In this section, we consider another definition of privacy that is based on a combination of k-anonymity and a recently proposed privacy metric known as Discrimination Rate Privacy Metric (DRPM) , and we refer to it as (k-anonymity, DR)-PAD.

In terms of privacy, discrimination rate provides a way to compute the identification capability of an attribute (or set of attributes) according to how knowledge of particular attribute values effects the adversary's capability to re-identify someone in the dataset. Interpretation of DRPM for interactive query settings provides a way to compute the identification capability of an SQL query instead of computing the identification capability of a single attribute or a set of attributes. Intuitively, the identification capability

of SQL query means that how likely it is for an adversary to re-identify someone from the dataset if that particular query is to be executed. In the next section, we review DRPM with an example, and use DRPM to the compute identification capability of an SQL query in (k-anonymity, DR)-PAD system.

7.4.1 Discrimination Rate (DR)

The discrimination rate [40] measures how the knowledge of a particular attribute values (or set of attributes values) refines the adversary capability to re-identify someone from the dataset. The discrimination rate privacy metric considers the attribute as a discrete random variable, while the result set is considered as the set of outcomes of another discrete random variable. For instance, consider two discrete random variables, X and Y, where X is the set of outcomes, and Y is the attribute for which the measurement of the identification capacity is desired. $H(X)$ (entropy) represents the amount of information carried by X. The entropy of X conditioned on Y, that is, $(H(X|Y))$ [40], is computed as the measure of the effect of Y on X. Therefore, the amount of information carried by Y (attribute) according to X is given by $H(X) - H(X|Y)$. $H(X) - H(X|Y)$ is divided by $H(X)$ to get a normalized value. The discrimination rate of an attribute is a value from the interval $[0, 1]$. The discrimination rate value 0 for an attribute means that the attribute does not contribute to refining adversary's knowledge who is attempting to re-identifying a subject(s). Attributes are treated as discrete random variables where X are the identifier attribute and Y is the attribute for which the identification capability is being computed. The discrimination rate is computed using Equation 7.1.

$$DR_X(Y) = 1 - \frac{H(X|Y)}{H(X)} \tag{7.1}$$

Where $H(X)$ is the entropy of a discrete random variable X, and is computed by Equation 7.2. $H(X|Y)$ is the conditional entropy of a discrete random variable X given a

discrete random variable Y, and is computed by Equation 7.3. The discrete random variable can take values from $\mathcal{S}$ with probability $p(x)$.

$$H(X) = -\sum_{x \in \mathcal{S}} p(x) log(p(x)) \tag{7.2}$$

$$H(X|Y) = -\sum_{x \in \mathcal{S}_1} \sum_{y \in \mathcal{S}_2} p(x,y) log(p(x|y)) \tag{7.3}$$

Where $p(x,y)log(p(x|y)$ are the joint and conditional probability distributions for discrete random variables X and Y. The discrimination rate is computed for the combination of attributes that is known as Combined Discrimination Rate (CDR). A discrimination rate value of 1 for an attribute means that the knowledge of the values of this attribute leads to re-identification of a subject(s). The combined discrimination rate for $Y_1, Y_2, \ldots Y_n$ given X is shown in Equation 7.4 (see [40] for more information).

$$CDR_X(Y_1, Y_2, \ldots Y_n) = 1 - \frac{H(X|Y_1, Y_2, \ldots Y_n)}{H(X)} \tag{7.4}$$

As an example, consider the relation shown in Table 7.8. The discrimination rate values for the attributes shown in the relation of Table 7.8 are shown in Tables 7.9 and 7.10.

Table 7.8: An example relation `companyTable`.

firstName	designation	location	drivingLicense
Zacharin	Application Developer	London	Yes
Alex	Project Manager	Rome	Yes
Sylvie	Application Developer	London	Yes
Nisha	Project Manager	Paris	Yes
Lukas	Security Analyst	London	Yes
Scott	Project Manager	New York	Yes
Cynthia	Application Developer	London	Yes
Ariel	Security Analyst	New York	Yes
Kate	Security Analyst	New York	Yes
James	Security Analyst	New York	Yes
Tom	Project Manager	London	Yes
Steve	Application Developer	Stuttgart	Yes

Table 7.9: Discrimination rate values for attribute shown in Table 7.8.

Attribute	Discrimination Rate (DR) Value
firstName	1
designation	0.442
location	0.544
drivingLicense	0

Table 7.10: Computed combined discrimination rate (CDR) values for attributes in the relation shown in Table 7.8.

Set of Attributes	CDR
firstName, designation	1
firstName, location	1
firstName, drivingLicense	1
designation, location	0.779
designation, drivingLicense	0.422
location, drivingLicense	0.544
firstName, designation, location	1
firstName, designation, drivingLicense	1
firstName, location, drivingLicense	1
firstName, designation, location, drivingLicense	1
designation, location, drivingLicense	0.779

The computational complexity of simple discrimination rate and combined discrimination rate is linear O(n). An aspect of combined discrimination rate is to get all the combinations of the quasi-identifiers, typically the number of quasi-identifiers are limited; therefore even though in theory the complexity of the algorithm that outputs all combination is $O(2^n)$, the process is practically is less computational costly because of the limited (small input size) number of quasi-identifier.

7.4.2 *DRSQL*: Computing Identification Capability of SQL Queries

The query abstraction used in order to translate the discrimination rate for SQL queries is shown in Table 7.11. The specialization of SQL query abstraction for *DRSQL* considered in this section, includes the attribute names as part of the abstraction. Given an

SQL query Q_i, its abstraction is denoted as $Abs(Q_i)$, where the elements of $Abs(Q_i)$ are the attributes in the SQL query.

Table 7.11: Deployed specialization of SQL query abstraction. The elements of $Abs(Q_i)$ are the attributes in the SQL query.

Query (Q_i)	SQL Query Abstraction Abs(Q_i)
`SELECT firstName, designation FROM companyTable;`	`{firstName, designation}`
`SELECT firstName, location FROM companyTable;`	`{firstName, location}`

Let the attributes in relation $\mathcal{T}$ be denoted as $\{attr_1, attr_2, attr_3, \ldots, attr_n\}$. Let L be an audit log consisting of SQL queries executed on relation $\mathcal{T}$. The abstraction of L is represented by $Abs(L)$ and the abstraction of an individual SQL query $Q_i \in L$ is represented as $Abs(Q_i)$. The identification capability of an SQL query Q_i is denoted as $DRSQL(Abs(Q_i)) = CDR_X(attr_1, attr_2, \ldots, attr_k)$ where $attr_1, attr_2, \ldots, attr_k \in Abs(Q_i)$. In a case where only a single attribute $attr$ is queried, then the identification capability of that query Q_i is $DRSQL(Abs(Q_i)) = CDR_X(attr) = DR_X(attr)$.

Table 7.9 and Table 7.10 show the computed discrimination rate values for the attributes in the table shown in Table 7.8. Here, X represents the number of unique records or individuals in the relations. In the case of the table shown in Table 7.8, there are 12 records where each record belongs to a unique individual. For instance, the identification capability of query Q_1, shown in Table 7.12, is $DRSQL(Abs(Q_1)) = CDR_X(\texttt{firstName, designation, drivingLicense}) = 1$. Similarly, the identification capability of query Q_4, shown in Table 7.12, is computed as $DRSQL(Abs(Q_4)) = CDR_X(\texttt{designation, location}) = 0.779$. This means that the response of Q_1 results in re-identification of subjects in the table while the response of Q_4 may results in re-identification of subjects if combined with other attributes with high discrimination rate.

Table 7.12: Sample SQL queries used to query the companyTable relation shown in Table 7.8.

Q_1	SELECT designation, firstName, drivingLicense FROM companyTable;
Q_2	SELECT designation, firstName FROM companyTable;
Q_3	SELECT designation, firstName, location FROM companyTable;
Q_4	SELECT designation FROM companyTable WHERE location = 'London';
Q_5	SELECT designation FROM companyTable WHERE location = 'New York';

7.4.3 Application of *DRSQL*: Privacy Comparison v/s Simple Comparison

A stand-alone application of computing identification capability of SQL queries is in the comparison of SQL queries. The higher the identification capability of an SQL query, the higher the risk of re-identification it carries when compared to an SQL query with lesser identification capability. There exist extensive literature on the comparison of SQL queries [93], however, the literature lacks approaches to compare two SQL queries using the privacy-risk they carry - *privacy comparison*.

In order to demonstrate the difference between a simple comparison and privacy comparison of two SQL queries, consider the relation companyTable as shown in Table 7.8. Suppose a first SQL query asks for the values for the attributes location, designation and drivingLicense. While a second query asks for the values of the attributes location and designation. A simple comparison of both queries syntactically as well as in terms of results indicates that both the queries differ by one attribute and one can say the result of second query is a subset of the results returned by first query. However, in terms of privacy, both the queries are equal as the attribute drivingLicense does not affect the privacy of an individual in the relation.

To facilitate the privacy comparison between SQL queries, we propose Privacy ordering relations in the next section.

7.4.3.1 Privacy Ordering Relations

To perform a privacy comparison between SQL queries, we define the following relations: *privacy-equivalence* relation, *less-private* relation and *more-private* relation. Given two SQL queries Q_i and Q_j and their abstractions as Q_i is more-private than Q_j or Q_j is less-private than Q_i denoted as $Abs(Q_i) \sqsubseteq Abs(Q_j)$. Intuitively, if the contribution of Q_i in refining the adversary's ability to re-identify subject(s) is lesser than of query Q_j, then Q_i is said to be more private than Q_j. Formally,

$$Q_i \sqsubseteq Q_j \equiv Abs(Q_i) \subseteq Abs(Q_j) \wedge DRSQL(Abs(Q_i)) \geq DRSQL(Abs(Q_j))$$

Given two SQL queries Q_i and Q_j and their abstractions $Abs(Q_i)$ and $Abs(Q_j)$, then the privacy equivalence between $Abs(Q_i)$ and $Abs(Q_j)$ is defined as $Abs(Q_i) \sqsubseteq Abs(Q_j) \wedge Abs(Q_j) \sqsubseteq Abs(Q_i)$. We denote the privacy equivalence relation as $\stackrel{p}{=}$ i.e. $Q_i \stackrel{p}{=} Q_j$.

In order to assist the privacy comparison and for a better explanation to compute the identification capability of the SQL queries, based on the discrimination rate and combined discrimination rate, one can then articulate a privacy-aware attribute relationship diagram for a relation $\mathcal{T}$. A fragment of the privacy-aware attribute relationship diagram is shown the Figure 7.2.

Consider the relation shown in Table 7.8 that is queried using the SQL queries Q_1, Q_2, Q_3, Q_4, and Q_5 shown in Table 7.12. The records returned in response to Q_2 are a subset of those returned in response to Q_1, and the records returned in response to Q_2 are also a subset of those returned in response to Q_3. Q_1 and Q_2 have privacy equivalence; however, Q_2 and Q_3 are not privacy equivalent. For the table shown in Table 7.8, the discrimination rate for the attribute `drivingLicense` is 0; therefore we

deduce that Q_1 and Q_2 hold privacy equivalence between them i.e. $Q_1 \overset{p}{=} Q_2$. While Q_1 and Q_3 do not hold privacy equivalence between them because of the Discrimination Rate for the attribute `location` is not zero, even though the result set of Q_2 is a subset of the result set of Q_3, i.e., $Q_2 \overset{p}{\neq} Q_3$.

Similarly, the case where one query is a subset of another query is considered as well; for instance, consider the SQL queries Q_2 and Q_3 from Table 7.12. The CDR for the sets of attributes `{firstName, designation, location}` and `{firstName, designation}` is computed. The CDR for `{firstName, designation, location}` is greater than the CDR of `{firstName, designation}`; therefore one can say Q_2 is a privacy subset of Q_3, that is, Q_2 is more-private then Q_3 or Q_3 is less-private then Q_2. The *DRSQL* value for Q_4 and Q_5 is 0.779 where the combined discrimination rate value for `{designation, location}` is 0.779.

In order to facilitate the comparison of queries, a privacy-aware attribute relationship diagram can be formulated, a fragment of which is shown in Figure 7.2. As an example, suppose one wants to compare the following two queries:

- `SELECT location, designation FROM companyTable;`
- `SELECT location FROM companyTable;`

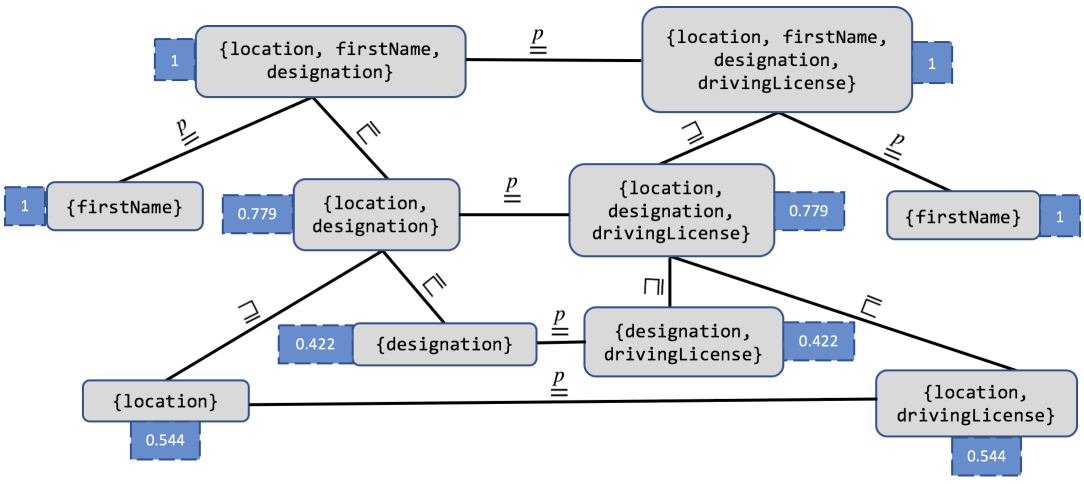


Figure 7.2: A fragment of privacy-aware attribute relationship diagram.

From the privacy-aware attribute relationship diagram it is implied that the set {location} is a subset of {location, designation} therefore, (SELECT location FROM companyTable;) $\sqsupseteq$ (SELECT location, designation FROM companyTable;).

The identification capability of an SQL query can be computed in many ways and with various specialization of query abstractions. One way to compute the identification capability of an SQL query is to compute it over the returned records. Identification capability depends on the distribution of attributes values in the database. For instance, a fine-grained specialization of query abstraction, denoted by $Abs^*(Q)$ can be employed where the attribute values and the operators used in the WHERE clause are part of query abstraction. For example, consider the following SQL query: Q = SELECT designation FROM companyTable WHERE location = 'Stuttgart'; $Abs^*(Q)$ = {designation, $\texttt{location}_{=\text{'Stuttgart'}}$ } will result in $DRSQL(Abs^*(Q)) = 1$.

7.4.4 A *k*-Anonymity and Discrimination Rate based Privacy-profile

There are various formal definitions of privacy in the literature. A question that arises is whether one can compose these definitions in a way, in the context of interactive querying, such that if one definition for the response is satisfied, then it is labelled as 'normal'. An alternative way of satisfying one privacy definition to be considered as 'normal' is compose the privacy definitions in a way such that if all of the privacy definitions are satisfied then it is considered as 'normal'. This is a much stricter notion of privacy which intuitively means less utility. We argue that for ease of exposition that satisfying one notion/definition of privacy constitutes 'normal'. We are interested in mining privacy limits using different privacy metrics to build a normative privacy-profile. For this reason, we use discrimination rate privacy metric along with *k*-anonymity to mine privacy limits and we refer to this instantiation of PAD as

(*k*-anonymity, *DR*)-PAD. For this proposed model, privacy is defined in the scenarios where records are returned in the response of a query where *k* individuals must share the same quasi-identifier values in the result of the query as well as the identification capability of that particular query must remain under some threshold. Intuitively, this means that for a particular instance of a relation (response of a query) one of the following two conditions must be satisfied: i) for a sufficient value of *k*, an adversary can only narrow down to *k* individuals, or ii) the knowledge of response does not increase adversary's capabilities to re-identify someone in the database.

7.4.4.1 Composing Privacy Criteria

We compose the privacy criteria *k*-anonymity and discrimination rate privacy metric whereby a query is said to be *query passed* if it satisfies one of the privacy criteria in the (*k*, anonymity, DR)-PAD profile. Given a *k*-anonymity privacy limit $[\![k_i, q_i]\!]$ with quasi identifiers q_i and a discrimination rate based privacy limit $DRSQL(Abs(Q_n))$ then define the aggregate privacy limit measure $P^{k_{q_i},Q_n}$ as the minimum of the *k*-anonymity privacy limit measure and the discrimination rate privacy limit measure, where minimum implies either of the privacy limit is satisfied.

Learning Phase In the learning phase, the values of *k*, for quasi-identifier (a set quasi-identifiers), are mined in a similar fashion to (*k*-anonymity)-PAD. The values of *DRSQL* are also learned for queries in the training phase. Using this information, a privacy limit $P^{k_{q_i},Q_n}$ is formed. We refer to the computed during the learning phase as $P^{k_{q_i},Q_n}_{Norm}$ and that computed at run-time during the detection phase as $P^{k_{q_j},Q_m}_{Run}$. In this instantiation of PAD, the normative privacy-profile $Pri(L^*_{norm})$ consists of privacy limits $P^{k_{q_i},Q_n}_{Norm}$.

In terms of privacy, the privacy limit in the normative privacy-profile represents what normative values of *k* are acceptable for a set of quasi-identifiers and what are the normative identification capabilities values associated with the SQL statements, both

learned from safe settings.

Detection Phase The detection phase, in terms of privacy, checks whether an adversary is able to narrow down fewer than k individuals or the adversary learns more from the query response as compared to what is normative. Where k for a given set of quasi-identifier is defined in the normative profile In the instance where the adversary is able to narrow down fewer than k individuals or the adversary learns more from the query response as compared to what is normative, then this instance is labelled as a privacy-anomaly as in terms of privacy it means higher risk of re-identification relative to normal.

Given privacy limits $P^{k_{q_i},Q_n}$ and $P^{k_{q_j},Q_m}$ then $P^{k_{q_i},Q_n}$ is no less private than $P^{k_{q_j},Q_m}$ if imposing privacy limit $P^{k_{q_j},Q_m}$ instead of $P^{k_{q_i},Q_n}$ leads to no additional loss of privacy.

Formally,

$$P^{k_{q_i},Q_n} \preceq P^{k_{q_j},Q_m} \equiv \quad (q_i \subseteq q_j \wedge k_j \geq k_i) \vee (Q_m \sqsubseteq Q_n)$$

In the detection phase, for a given query Q_m, if at run-time value $P^{k_{q_j},Q_m}_{Run} \in Pri(L^*_{Run})$ if $P^{k_{q_i},Q_n}_{Norm} \preceq P^{k_{q_j},Q_m}_{Run}$, then it is labelled as a safe, in all other cases the query is labelled as an anomaly.

7.4.4.2 Example Run of (*k*-anonymity, *DR*)-PAD

Consider the queries shown in Table 7.14 the were executed on the `table_smp` relation shown in Table 7.13.

Suppose that the mined value of k for the set of quasi-identifier q_i in Q_1 and Q_2 is 4 and the mined $DRSQL(Abs(Q_1))$ and $DRSQL(Abs(Q_2))$ is 0.5. The result set of query Q_1 is shown in Table 7.15 with the value of $k = 5$ for the set of quasi-identifier q_i in Q_1 which is greater than the mined value of k. The value for $DRSQL(Abs(Q_1))$ is 0.4184

Table 7.13: A sample relation table_smp of records for the running example.

Name (DR =1)	MaritalStatus (DR =0.4184)	City (DR =0.4184)	Age (DR =0.21)	Salary (sensitive attribute)
Mark	Single	New York	[30 - 40]	112k
James	Single	New York	[30 - 40]	34k
John	Single	New York	[30 - 40]	56k
Henry	Single	New York	[30 - 40]	78k
Imran	Single	New York	[30 - 40]	91k
David	Married	London	[30 - 40]	112k
Alice	Married	London	[30 - 40]	30k
Bob	Married	London	[30 - 40]	45k
Aron	Married	London	[30 - 40]	115k
Harry	Married	London	[30 - 40]	180k
Jordan	Separated	Cork	>40	65k
Ryan	Separated	Cork	>40	100k
Bentley	Separated	Cork	>40	80k

Table 7.14: Sample SQL queries executed over the relation table_smp shown in Table 7.13

Q_1	SELECT MaritalStatus, Salary, *age* FROM table_smp WHERE City ='New York';
Q_2	SELECT MaritalStatus, Salary, *age* FROM table_smp WHERE City=Cork;

that is lesser than the mined value of $DRSQL(Abs(Q_1))$. The query Q_1 satisfies both the thresholds, and it can be said to query Q_1 as *Query passed*. The result set of query Q_2 is shown in Table 7.16. In the case for query Q_2, the value of k at run-time is 3 which is less then the mined value. The $DRSQL(Abs(Q_2))$ is 0.4184 which is lesser than the mined value of $DRSQL(Abs(Q_2))$. As one of the threshold is passed so it can be said that query Q_2 as *Query passed*. Intuitively, for both the queries the adversary does not learn about the individuals in the dataset more than what is considered as normative and cannot further narrow down to less then k individuals.

Table 7.15: Records returned in the response to query Q_1.

MaritalStatus	Age	Salary (sensitive attribute)
Single	[30 - 40]	112k
Single	[30 - 40]	34k
Single	[30 - 40]	56k
Single	[30 - 40]	78k
Single	[30 - 40]	91k

Table 7.16: Records returned in the response to query Q_2.

MaritalStatus	Age	Salary (sensitive attribute)
Separated	>40	65k
Separated	>40	100k
Separated	>40	80k

7.5 Privacy Attacks (Inferences)

This section demonstrates a privacy attack whereby an adversary discovers information about an individual while the privacy-anomaly detection system is in place. Consider the relation shown in Table 7.17 that is the same relation as shown in Table 7.13 but with one more record added.

Suppose the sequence of queries Q_1, Q_2 and Q_3, shown in Table 7.18, are executed over the relation `updated_table_smp` shown in Table 7.17.

Query Q_1 is labelled as an privacy-anomaly (and the query response is suppressed) because the threshold is not satisfied as *k*-anonymity and *DRSQL* is not satisfied by the query Q_1. Whereas query Q_2 passes the threshold and the response to the query is shown in Table 7.19 that results in 14 records, that is, the value of attribute `Salary`, being returned. Query Q_3 also passes the privacy criteria and returns 13 records, as shown in Table 7.20. The adversary, knowing that `Simon's` record is in the table (as background/external knowledge) and that Simon lives in Rennes, reveals that last remaining entry blocked by the query mechanism is of '`Simon`' and the corresponding salary attribute value is 150k.

Table 7.17: Relation updated_table_smp: updated version of the table_smp relation shown in Table 7.13.

Name		City	Age	Salary (sensitive attribute)
Mark	Single	New York	[30 - 40]	112k
James	Single	New York	[30 - 40]	34k
John	Single	New York	[30 - 40]	56k
Henry	Single	New York	[30 - 40]	78k
Imran	Single	New York	[30 - 40]	91k
David	Married	London	[30 - 40]	112k
Alice	Married	London	[30 - 40]	30k
Bob	Married	London	[30 - 40]	45k
Aron	Married	London	[30 - 40]	115k
Harry	Married	London	[30 - 40]	180k
Jordan	Separated	Cork	>40	65k
Ryan	Separated	Cork	>40	100k
Bentley	Separated	Cork	>40	80k
Simon	Married	Rennes	>40	150k

Table 7.18: Sequence of queries executed over the relation updated_table_smp shown in Table 7.17.

Q_1	SELECT Salary FROM updated_table_smp WHERE city = 'Rennes';
Q_2	SELECT Salary FROM updated_table_smp;
Q_3	SELECT MaritalStatus, Salary, Age FROM updated_table_smp WHERE City= 'New York' AND city ='London' AND city = 'Cork';

In particular, the described attack is a form of a differencing attack [198]. Differencing attacks have been seen previously on aggregates. This demonstrates that k-anonymity in interactive settings is also susceptible to these differencing attacks. For the purpose of demonstration, the example of a differencing attack is kept simple; however, real world differencing attacks can take more sophisticated forms, where the adversary can make multiple queries to narrow down the aggregate data until the subject's information is not revealed.

Table 7.19: Records returned in the response to query Q_2.

Salary (sensitive attribute)
112k
34k
56k
78k
91k
112k
30k
45k
115k
180k
65k
100k
80k
150k

Table 7.20: Records returned in response of query Q_3.

MaritalStatus	City	Age	Salary (sensitive attribute)
Single	New York	[30 - 40]	112k
Single	New York	[30 - 40]	34k
Single	New York	[30 - 40]	56k
Single	New York	[30 - 40]	78k
Single	New York	[30 - 40]	91k
Married	London	[30 - 40]	112k
Married	London	[30 - 40]	30k
Married	London	[30 - 40]	45k
Married	London	[30 - 40]	115k
Married	London	[30 - 40]	180k
Separated	Cork	>40	65k
Separated	Cork	>40	100k
Separated	Cork	>40	80k

7.6 Applying Security-anomaly Detection to Detect Unknown Privacy Attacks

In general, interactive query mechanisms are susceptible to the attacks described in the previous section, and as a consequence there is little privacy-preserving interactive query mechanisms (specifically for microdata release) in the existing literature. The

existing differentially private interactive mechanisms allow a limited number of interactive queries for aggregate data. Restricting the number of queries is a significant barrier for an analyst in achieving the true potential for data analytics. Privacy attacks, similar to the one presented in the previous section, are violations of formal privacy definitions.

Another aspect of these privacy attacks is that the querying pattern to infer information about the subject(s) is unknown, therefore, we refer to them as unknown privacy attacks. Unknown privacy attacks lead to inferring information about the subject(s). An adversary can articulate the queries in different ways to reveal information about a subject(s). In this work, inference implies privacy attacks, that is, the adversary infers information about the subject(s) while a formal privacy definition is in place resulting in a violation of formal privacy definition. Additionally, these privacy attacks are based on query correlation, that is, an individual query is safe in terms of privacy but when considered as sequence they result in the violation of formal privacy definition. This section presents an investigation into whether the inferences can be detected as anomalies.

We present a novel perspective on the detection of privacy attacks by proposing an interpretation of the behavioural-based detection approach. We investigate the application of n-gram approach, proposed approach in Chapter 4, to detect unknown privacy attacks as anomalies in the next section.

7.6.1 Detecting Inferences as Anomalies

A behavioural-based approach to detect inferences as anomalies is described in this section where the n-gram based approach is applied to the audit logs of the (k-anonymity)-PAD system. The idea is to model querying behaviours in the presence of a privacy-preserving interactive query mechanism and compare the normative querying behaviour with the run-time querying behaviour to detect deviations.

For the SQL query abstraction, the specialization of the abstraction, as discussed in Section 4.3.2, is employed. Examples of the employed SQL query abstraction technique are shown in Table 4.1.

The n-gram profiles are generated in the same manner as discussed in Section 4.3.3 that is given a safe audit log of SQL query L^{pp}_{norm} and a run-time log L^{pp}_{run} then the constructed normative profile and run-time profile are $\beta_{norm} = ngram(Abs(L^{pp}_{norm}), n)$ and $\beta_{run} = ngram(Abs(L^{pp}_{run}), n)$. The mismatches are given by $S^{miss}_{\beta_{run}-\beta_{norm}} = \beta_{run}$ - β_{norm}.

In order to evaluate the detection of privacy attacks by applying the n-gram based approach to the logs of interactive querying mechanism, a synthetic query generator was designed that had defined a set of SQL query templates. The underlying database was populated with a fragment version of well-known Census (Adult) dataset [199]. Query templates were designed to be executed on the Census dataset. The queries were count queries mimicking a data analytics scenario. For example, the count queries were for the form: how many subjects are Female, how many subjects have a Bachelors degree, so on and so forth. For the experimentation, a safe log L^{pp}_{norm} was generated for the construction of normative profile using the synthetic data generator.

In order to construct privacy attacks, five unique records were inserted into the database that leads to inferences, where unique implies that one of the attribute or combination of the attributes existed only once in the entire database. For example, a record was inserted with `occupation` as `post-doc`, that is, in the underlying database there was only one record where the value for `occupation` was `post-doc`. Another record was inserted where the value for `native-country` was set to `Malaysia`, that is, there was only one record where the `native-country` was `Malaysia`. Table 7.21 shows the make-up of the inserted records to enable privacy attacks.

Queries were made to infer the associated salaries for these five unique records. Table 7.22 shows the number of queries made to reveal the salary for the records shown in Table 7.21. The length of the sequences differs for each attack; this is because, in

Table 7.21: Inserted unique records in the database to enable privacy attacks.

#	Description of Unique Record
1	Attribute `occupation` with value as `post-doc`
2	Attribute `native-country` with value as `Malaysia`
3	Attribute `native-country` with value as `Spain` and Attribute `age` as `33`
4	Attribute `native-country` with value as `Singaporean` and Attribute `age` as `32`
5	Attribute `native-country` with value as `Singaporean` and Attribute `occupation` as `Academics`

each scenario, the number of queries required to reveal information about salary was different. This can be best described from a simple example; for instance, consider a database with 100 employee records where record consisted of their salaries as well. There are 99 male employees and 1 female employee. In order to find the salary for 1 female employee, an adversary will have to execute the first queries to discover the sum of salary for all the employees and then a query to discover the sum of salaries for all the male employees. The difference between the sum of the salaries of all the employees and the sum of the salaries for male employes will reveal the salary for the one female employees. Thus, in this case, the adversary has to execute only two queries to reveal the salary for the female employees. This can become complex depending on the what personal information the adversary is aiming to reveal about an individual in the database and the length of the sequence increases or decreases based on that. These malicious query sequences were made part of the other logs L^{pp}_{run} for the construction for the run-time profile.

Table 7.22: Length of the query sequences to reveal salaries.

Privacy Attack #	Inference Sequence Length
1	7
2	9
3	17
4	13
5	10

A normative profile β_{norm} and a run-time profile β_{run} were constructed using L^{pp}_{norm} and

L^{pp}_{run}, respectively and compared. The size of the n-gram was kept at 4 for the generation of profiles. The sequence of queries made to infer injected unique records was labelled as anomalies the detection phase by the n-gram approach. The Table 7.23 shows the number of mismatches for each privacy attack.

Table 7.23: Detection of privacy attacks (inferences) as anomalies: the table shows the number of mismatches that resulted from each of the 5 privacy attacks with the n-gram of size 4.

Privacy Attack #	$\lvert S^{miss}_{\beta_{run}-\beta_{norm}} \rvert$
1	5
2	7
3	19
4	15
5	13

The query sequences resulting in inferences were detected as privacy-anomalies, which is a indication of potential effectiveness of n-gram based approach to detect inference and unknown privacy attacks as privacy-anomalies.

7.7 Conclusions

While previous chapters considered anomalies arising from anomalous queries of users, this chapter explores a anomalies characterised in terms of formal definitions of privacy. This chapter explores privacy semantic notion of behavioural-based anomaly detection systems. The notion of privacy-anomaly detection (PAD) introduced in this chapter enables one to learn a privacy model from the past log of interaction with the DBMS and detects deviations as privacy-anomalies. Two instantiations of PAD was presented, one based on k-anonymity and the second instantiation (k-anonymity, DR)-PAD was based on the composition of k-anonymity and discrimination rate privacy metric. A study was carried out to examine whether the privacy violations based on a single query detected the privacy-anomaly detection system are also detected

by n-gram security-anomaly detection system as anomalies. Results showed a number of single query based privacy violations that had no reference n-gram in normative profiles were labelled as anomalies by n-gram based anomaly detection system. This chapter also considered privacy attacks that were violations of formal privacy definitions and were based on query correlation where a single query is not privacy-anomalous but a sequence of queries results in a violation of formal privacy definition. Results showed that an n-gram based approach detected these privacy attacks as privacy-anomalies. This led to a discovery of another aspect of the n-gram based approach whereby when it is applied on the logs generated by interactive query settings with the presence of formal privacy definition, it has the potential to detect privacy attacks based on query correlation as privacy-anomalies.

As a topic of future work, we plan to explore how to compose and compare multiple privacy definitions using multi-criteria decision-making method found in fuzzy logic [200, 201], in particular, known as *triangular-norms* (*t-norms*).

Chapter 8

PriDe: A Quantitative Measure of Privacy

"Obviously, the highest type of efficiency is that which can utilize existing material to the best advantage."

Jawaharlal Nehru (1889 - 1964)

8.1 Introduction

This chapter considers the question of whether it is possible to quantitatively measure the deviation from normative with respect to privacy. In order to address this question, we introduce the notion of **Pr**ivacy **D**istanc**e** (PriDe). The privacy distance is the distance in terms of privacy between normative querying behaviour and run-time querying behaviour. We refer to the privacy distance as a (single) score because PriDe is a semi metric as discussed in Section 8.3.2; therefore, we refer to it as privacy-loss score. Section 8.2 describes the design of PriDe and how the score is computed. Sec-

tion 8.3 describes the computation of cumulative score. Section 8.4 evaluates the PriDe model. Conclusions have been drawn in Section 8.5 .

8.2 PriDe - The Model

In this section, the design of the PriDe model is outlined. Intuitively, PriDe quantitatively measures the information gain by the user (can potentially be an adversary) or the amount of privacy lost by individuals due to the disclosure of information while the adversary was engaged in the interactive querying session. The proposed approach uses n-gram profiles, constructed in Chapter 4, to model the querying behaviour of a user. Once n-gram profiles are generated, a comparison of the profiles is carried out for computation of the score.

In this work, we consider an adversary model where a user/analyst (Insider) who have been granted access to the database, potentially have malicious intent and attempt to gain insights about the database by sporadically querying the database. The adversary (insider) is *information greedy*; therefore, the goal of the adversary is to gain as much information as possible. The database is maintained by a contemporary organization (data curator), and the database consists of distinct records of numerous individuals.

In the next section, we demonstrate using a simple example privacy score by demonstrating the difference between a naïve calculation for the amount of data released in the response of SQL queries made by an analyst and the distance in terms of privacy.

8.2.1 Difference between Naïve Calculation and PriDe Model

Consider the relation having attributes including `firstName`, `lastName`, `department`, `gender`, `city`, and `departmentHead`, as shown in Table 8.1. Assume that every entry for the attribute `city` in the relation is `Vancouver`. Suppose that the very first five

queries that an analyst makes to the relation are of selecting only the value for the attribute `city`. A naïve calculation of the amount of data retrieved by the analyst would be 5 values (the number of records selected) for the attribute `city`; however, the returned values (`Vancouver`) do not affect the privacy of any individual in the database.

The PriDe model calculates the amount of information, through attribute values, released to the analyst in-terms of privacy. From 'in-terms of privacy' we mean whether or not this release affects the privacy of any individual. Suppose an analyst makes queries to get values for the attribute `gender` followed by queries to get values for the attribute `department` by specifying the condition in the `WHERE` clause of the SQL statement as `WHERE city = 'Vancouver'`. We would expect that this behaviour results in increasing the score. Again, if the analyst makes another query asking for the value for the attribute `firstName`, this further increases the score. However, after making these queries, the analyst makes a query asking for the value of attribute `city`, then this does not increase the score as this query does not affect the privacy of any individual in the database.

Another aspect in computing the score is of the consideration of 'safe' past behaviour or 'safe' queries. For instance, we know that it is a usual practice that a query to get the value of the attribute `department` is followed by the query to get the value of the attribute `departmentHead` and vice versa. This sequence of queries, if appeared at run-time, does not increase the score.

8.2.2 The Design

The computation of the score is based on behavioural profiles representing querying behaviour. In PriDe model the behavioural profiles are inferred from a DBMS log of the SQL queries (audit logs) using the n-gram based approach presented in Chapter 4. The idea is to capture the normative querying behaviour of a user in one profile (baseline profile) and capture the posterior querying behaviour of the user in another profile

Table 8.1: An example table having records of five individuals.

firstName	lastName	gender	department	city	departmentHead
John	Smith	Male	Oncology	Vancouver	Dr.George
Bob	Lopez	Male	Oncology	Vancouver	Dr.George
Alice	Miller	Female	Oncology	Vancouver	Dr.George
Bob	Smith	Male	Cardiology	Vancouver	Dr.Albert
John	Wilson	Male	Cardiology	Vancouver	Dr.Albert

(run-time profile). Then a comparison (in a privacy sense) of the baseline profile and the run-time profile results is quantified in a single score.

8.2.3 Constructing Profiles of Querying Behaviour

Behavioural profiles are n-gram profiles inferred from a DBMS log of the SQL queries (audit logs) in the same manner from audit logs as described in Section 4.3.3. The audit log L consists of SQL queries including `SELECT`, `UPDATE`, `INSERT` and `DELETE` statements. The queries in the audit log L are a mix of simple queries as well as complex queries that involve joins, `GROUP BY` statements, `HAVING` clauses, nested queries, and so forth. Let $Abs(L)$ denote the log with each query abstracted. The first element of a deployed query specialization consists of the command type, that is, `SELECT`, `UPDATE`, `DELETE`, `INSERT`. The second element of is the attribute and relation names of the command. Furthermore, to differentiate the attribute values queried and the attributes in a `WHERE` clause, the attributes of the `WHERE` clause are affixed with a subscript '$_w$' indicating that the attribute occurs in the `WHERE` clause, for example, $\texttt{gender}_w$. An example of deployed query abstraction specialization is shown in Table 8.2. The set $ngram(Abs(L), n)$ gives us the set of all of the sub-sequences of size n that appear in $Abs(L)$. The unique n-grams in the set $ngram(Abs(L), n)$ results in a profile consisting of unique n-grams of the SQL query abstractions.

Table 8.2: Deployed SQL query abstraction.

Query	SQL Query Abstraction

`SELECT lastName, dob, company`
`FROM tabledb`
`WHERE firstName = 'Smith';`
	{`SELECT, lastName, dob, company,`
	`tabledb,` $\texttt{firstName}_w$}

`SELECT lastName, street`
`FROM tabledb`
`WHERE firstName = 'John';`
	{`SELECT, lastName, street,`
	`tabledb,` $\texttt{firstName}_w$}

8.2.3.1 Comparison of Profiles

Intuitively, the privacy score, indicates the objective changes between the past and the current querying behaviour of the analysts in terms of privacy. The proposed approach computes score by comparing past and current analyst querying behaviour. Both of these behaviours are represented by n-gram profiles. We denote an n-gram profile as β. SQL audit logs collected for the construction of baseline n-gram profile and run-time n-gram profile are denoted as $L_{\mathcal{N}}$ and $L_{\mathcal{R}}$, respectively. The baseline profile and the run-time profile constructed from these logs are denoted as $\beta_{\mathcal{N}} = ngram(Abs(L_{\mathcal{N}}), n)$ and $\beta_{\mathcal{R}} = ngram(Abs(L_{\mathcal{R}}), n)$, respectively.

Let the set of mismatched n-grams, when the baseline profile and the run-time profile are compared against each other, is denoted by $S_{miss}^{\beta_{\mathcal{R}},\beta_{\mathcal{N}}} = \beta_{\mathcal{R}} - \beta_{\mathcal{N}}$. Let $|S_{miss}^{\beta_{\mathcal{R}},\beta_{\mathcal{N}}}|$ be the number of mismatches or the number of elements in the set $S_{miss}^{\beta_{\mathcal{R}},\beta_{\mathcal{N}}}$. We denote a mismatched n-gram as $G_i^{miss} \in S_{miss}^{\beta_{\mathcal{R}},\beta_{\mathcal{N}}}$.

We need to go beyond the simple comparison of n-gram profiles (subtracting baseline profile from run-time profile) and would like to have a more fine-grained comparison at n-gram level and subsequently at query (abstraction) level for the following two reasons:

- We are interested in determining the closest match in baseline profile $\beta_{\mathcal{N}}$ for the mismatched n-gram G_i^{miss}. Because when we perform a simple subtraction comparison, we tend to do a binary comparison, that is, either the n-gram is the same as the other n-gram, or it is not. For instance, consider following three n-grams G_1 = ⟨(`SELECT`, `firstName`), (`SELECT`, `department`)⟩, G_2^{miss} = ⟨(`SELECT`, `firstName` , `lastName`), (`SELECT`, `department`)⟩, G_3^{miss} = ⟨(`SELECT`, `city`), (`SELECT`, `gender`)⟩ where $G_1 \in \beta_{\mathcal{N}}$ and G_2^{miss}, $G_3^{miss} \in S_{miss}^{\beta_{\mathcal{N}},\beta_{\mathcal{R}}}$. If we compare G_2^{miss} and G_3^{miss} against G_1, intuitively, G_2^{miss} have some degree of similarity with G_1, while on the other hand, G_3^{miss} is entirely different from G_1. Thus we need to take into account the degree of the similarity of the mismatched

n-gram from its closest match if we desire to have a 'refined' comparison between two profiles.

- When we talk about privacy then we have to consider the *zero-identifier* that is an attribute (or a set of attributes) if known to the attacker, does not contribute in affecting the privacy of the individual in the database. A zero-identifier is an identifier for which the Discrimination Rate (DR) privacy metric value is 0 [40]. In our running example relation shown in Table 8.1, the attribute `city` is a zero-identifier. The adversary, who is engaged in an interactive query session, initially may not know the values for zero-identifier. The adversary can discover the values of zero-identifier in response to the queries; however, knowledge of zero-identifier values does not affect the privacy of the data subjects.

8.2.3.2 Distance Between n-grams

In order to find the closest match of the mismatched n-gram $G_i^{miss} \in S_{miss}^{\beta_N,\beta_R}$, that is, how far G_i^{miss} is from its closest match in β_N, a measure to compare two n-grams is desired. To find the closest match, we deploy the strategy of comparing the corresponding SQL query abstraction of two n-grams. In SQL query (abstraction) similarity research, Jaccard distance has been commonly used to find similarity between two SQL query abstraction [202, 203, 204]. For comparing two query abstractions using Jaccard distance, we have to consider an SQL abstraction as a set for comparison. The Jaccard distance between two query abstraction $Abs(Q_i)$ and $Abs(Q_j)$ is given in Equation 8.1.

$$JaccardD(Abs(Q_i), Abs(Q_j)) = \frac{(|Abs(Q_i) \cup Abs(Q_j)| - |Abs(Q_i) \cap Abs(Q_j)|)}{|Abs(Q_i) \cup Abs(Q_j)|} \quad (8.1)$$

If two SQL queries abstractions are entirely dissimilar, then the value is 1, and if they are exactly the same, then the value is 0. We define the function for the comparison of two n-grams as follows: given two n-grams G_i and G_j the distance between them is

computed using the function $Dist(G_i, G_j)$ shown in Equation 8.2.

$$Dist(G_i, G_j) = \sum_{r=1}^{n} JaccardD(G_{i_r}, G_{j_r}) \tag{8.2}$$

Where r in Equation 8.2 is the index (position) of the item in the n-gram. N-grams of length n result in n comparisons of SQL query abstractions. The value of $Dist(G_i, G_j)$ fall in the interval $[0, n]$, where if two n-grams are identical then the value is 0 and if two n-grams of length n are distinct, then one gets the value n.

Table 8.3: Sample SQL queries.

Q_1	`SELECT lastName, gender, city`
	`FROM hospitaldb;`
	`WHERE firstName = 'Bob'`

Q_2	`SELECT lastName, gender`
	`FROM hospitaldb;`
	`WHERE firstName = 'Bob'`

Q_3	`SELECT lastName, gender, department` `FROM hospitaldb;`

Q_4	`SELECT lastName, department`
	`FROM hospitaldb ;`
	`WHERE firstName = 'Alice'`

8.2.3.3 Privacy Equivalence

Consider the relation shown in Table 8.1 is queried using the SQL queries Q_1, Q_2, and Q_3 shown in Table 8.3. The records returned by Q_2 is a subset of that returned by Q_1, and the records returned by Q_2 is also a subset of that returned by Q_3 in-terms of privacy. Q_1 and Q_2 have privacy equivalence; however, Q_2 and Q_3 are not privacy equivalent. The notion of privacy equivalence relation $\stackrel{p}{=}$ introduced in Section 7.4.3.1 was employed here to compare two SQL query abstraction.

For the relation shown in Table 8.1 the Discrimination Rate for the attribute `city` is 0, therefore, we deduce that for the queries shown in Table 8.3, Q_1 and Q_2 hold privacy equivalence between them while Q_1 and Q_3 do not hold privacy equivalence between

them because the Discrimination Rate value for the attribute `department` is not zero, although the result set of Q_2 is a subset of the result set of Q_3. An example of the comparison of n-grams with privacy equivalence relation is shown in Example 1.

Example 1. Consider the following two n-grams $\langle Abs(Q_1), Abs(Q_3)\rangle$ and $\langle Abs(Q_2), Abs(Q_4)\rangle$ where the queries Q_1, Q_2, Q_3, and Q_4 are shown in Table 8.3. Using the technique in Section 7.4.2 and Section 7.4.3, we determine $Abs(Q_2) \stackrel{p}{=} Abs(Q_1)$ for the Table 8.1. The distance between the two above mentioned n-grams is given by $Dist(\langle Abs(Q_1), Abs(Q_3)\rangle, \langle Abs(Q_2), Abs(Q_4)\rangle)$,

$$
\begin{aligned}
&Dist(\langle Abs(Q_1), Abs(Q_3)\rangle, \langle Abs(Q_2), Abs(Q_4)\rangle) = \\
&JaccardD(Abs(Q_1), Abs(Q_2)) + JaccardD(Abs(Q_3), Abs(Q_4)) \\
&= 0 + (6-4)/6 \\
&= 0.33
\end{aligned}
$$

□

8.2.3.4 Computing The Score

The score is essentially the sum of the distances between the closest match, in the baseline profile, of the mismatched n-grams. We drive formula for the score as follows: consider $S_{miss}^{\beta_{\mathcal{N}},\beta_{\mathcal{R}}} = \{G_1^{miss}, G_2^{miss}, \ldots, G_k^{miss}\}$ and $\beta_{\mathcal{N}} = \{G_1, G_2, \ldots, G_m\}$. Each n-gram $G_i^{miss} \in S_{miss}^{\beta_{\mathcal{N}},\beta_{\mathcal{R}}}$ is compared with each n-gram G_i in the baseline profile $\beta_{\mathcal{N}}$. Thus resulting in the total number of $k \times m$ comparisons that is $\langle Dist(G_1^{miss}, G_1), Dist(G_1^{miss}, G_2), Dist(G_1^{miss}, G_3), \ldots, Dist(G_1^{miss}, G_m)\rangle$, $\langle (G_2^{miss}, G_1), Dist(G_2^{miss}, G_2), Dist(G_2^{miss}, G_3), \ldots, (Dist(G_2^{miss}, G_m)\rangle, \ldots, \langle Dist(G_k^{miss}, G_1), Dist(G_k^{miss}, G_2), Dist(G_k^{miss}, G_3), \ldots,$

$Dist(G_k^{miss}, G_m)\rangle$. We denote each iteration as $Iter_l = \langle Dist(G_i^{miss}, G_j), Dist(G_i^{miss}, G_{j+1}), Dist(G_i^{miss}, G_{j+2}), \ldots, Dist(G_i^{miss}, G_{j+m})\rangle$. Where a single iteration here is defined as the comparison of one n-gram from $S_{miss}^{\beta_\mathbb{N},\beta_\mathbb{R}}$ with every n-gram in $\beta_\mathbb{N}$. We take the minimum value of *Dist* from each iteration, i.e., $Min(Iter_l)$ that belongs to the interval $[0, n]$. Subsequently, the summation of all the $Min(iter_l)$ results in a deviation score.

Formally, given two n-gram profiles, the baseline profile $\beta_\mathbb{N}$ and the run-time profile $\beta_\mathbb{R}$ then the score between these n-gram profiles is denoted as $P_{\langle\beta_\mathbb{N},\beta_\mathbb{R}\rangle}$ and computed using Equation 8.3.

$$P_{\langle\beta_\mathbb{N},\beta_\mathbb{R}\rangle} = \sum_{i=1}^{k}\sum_{j=1}^{m} Min(Dist(G_i^{miss}, G_j)). \tag{8.3}$$

Where $G_i^{miss} \in S_{miss}^{\beta_\mathbb{N},\beta_\mathbb{R}} = \beta_\mathbb{R} - \beta_\mathbb{N}$ and $G_j \in \beta_\mathbb{N}$. It is worth mentioning that the design of PriDe can be considered as modular in nature. Therefore, one can use in any other similarity measure for query abstractions. Example 2 shows a comparison of two n-grams. The process of profile comparison is also depicted in Figure 8.1.

Example 2. Consider the following two n-grams $\langle Abs(Q_1), Abs(Q_2)\rangle$ and $\langle Abs(Q_3), Abs(Q_4)\rangle$ where queries Q_1, Q_2, Q_3, and Q_4 are shown in Table 8.3. The comparison of the two n-grams is given as follows,

$$\begin{aligned} &Dist(\langle Abs(Q_1), Abs(Q_2)\rangle, \langle Abs(Q_3), Abs(Q_4)\rangle) = \\ &JaccardD(Abs(Q_1), Abs(Q_3)) + JaccardD(Abs(Q_2), Abs(Q_4)) \\ &= (7-4)/7 + (6-4)/6 \\ &= 0.76 \end{aligned}$$

□

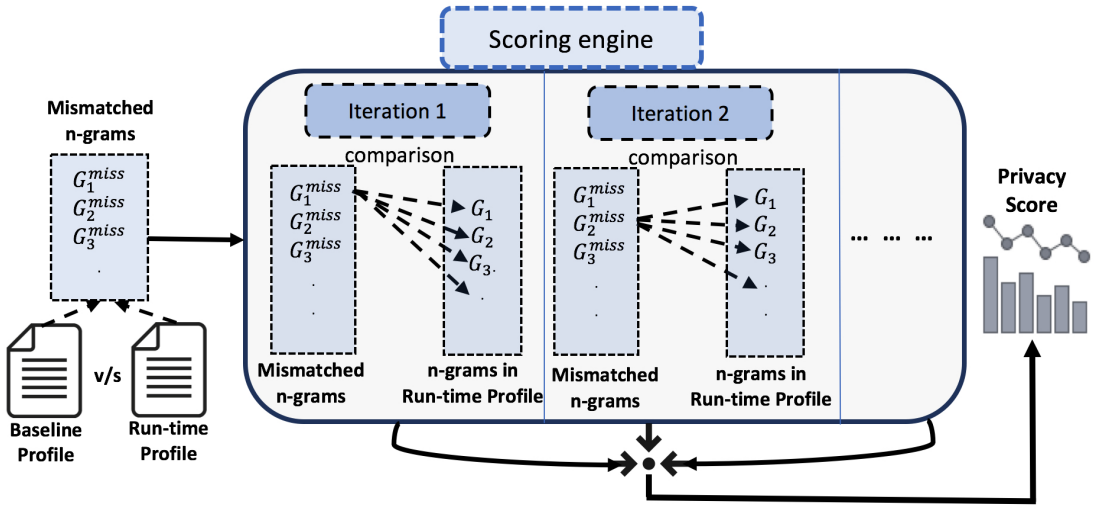


Figure 8.1: The run-time profile is compared with the baseline profile resulting in a score. Each n-gram from the set of mismatched n-grams is compared with each n-gram in the baseline profile. The minimum value of the Jaccard distance is taken from one iteration of comparison and, subsequently, all the minimum values are added together to get the score.

8.2.4 The Scenario of Cold Start

The PriDe model also considers the special scenario of the cold start, which is the absence of normative querying behaviour. In this scenario, only a run-time profile is generated, and the score is computed solely using this run-time profile because a baseline profile to compare against the run-time profile is unavailable. In the case of a cold start, the privacy score is the distance between run-time querying behaviour and point zero. One can think of the comparison in the cold start scenario as comparing the run-time profile with an empty (baseline) profile. The score in this scenario is computed as the *'total number of unique n-grams* $\times$ *n'*, where *n* is the size of the n-gram in the run-time profile i.e. $|\beta_{\mathcal{R}}| \times n$.

8.3 Cumulative Score

To compute the score, one generates a baseline profile and a run-time profile and subsequently compares both the profiles with each other. The question that arises is, 'when to construct the run-time profile?'. As for the construction of the baseline profile, one can construct it before the information system is up and running. On the other hand, the run-time profile is constructed when the information system is operational, meaning access to the analyst has been granted, and the analyst has started making queries. A possible answer to the question of 'when to construct the run-time profile?' is to divide the time horizon into equal intervals. The time horizon is the total time period for which the analyst was granted access to the database. For simplicity, we opt to construct a run-time profile by the end of each day. That is to say, audit logs are collected by the end of each day and then, the run-time profile is generated using these logs.

We denoted, the run-time profile constructed by the end of the x^{th} day as β_X. β_0 represents profile constructed on day 0 or the baseline profile. It is worth mentioning that the baseline profile is generated only once. We denote the score computed on x^{th} day as $P_{\langle\beta_0,\beta_X\rangle}$ for example, the score computed on day 2 is denoted as $P_{\langle\beta_0,\beta_2\rangle}$. In the cold start scenario, where the baseline profile is unavailable, we denote the score as $P_{\langle\{\},\beta_1\rangle}$ computed on day 1 and similarly score computed on day 2 as $P_{\langle\{\},\beta_2\rangle}$, and so forth.

Imagine the user of the system gets a score for each day but desires a cumulative score till day 10. A naïve addition of scores for all 10 days (from day 1 to day 10) is not an accurate representation of the cumulative score. For instance, a user can make identical queries in the same order everyday for 10 days, thus resulting in identical n-gram profile for each day. Therefore, one must take into account a combination of n-gram profiles for 10 days in such a way that only unique behaviours (unique n-grams) for all 10 days are part of the combined n-gram profile of all 10 days. To unravel this, we denote the cumulative score as Δ. The cumulative score from day 1 to day X is denoted as $\Delta[<\beta_0>, <\beta_1,\beta_2,\ldots,\beta_X>]$, where $<\beta_1,\beta_2,\ldots,\beta_X> = (\beta_1 \cup \beta_2 \cup \beta_3 \cup \beta_4, \ldots, \cup\beta_X)$

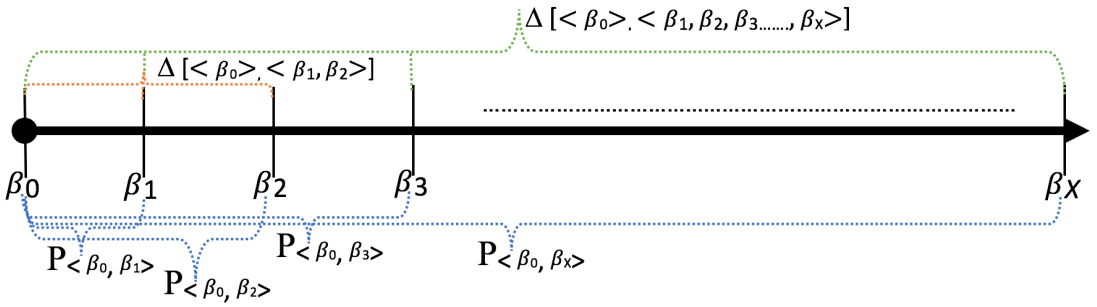


Figure 8.2: Variations of score computations: This figure shows a variety of ways in which individual scores and cumulative scores can be computed. For example, $\Delta[<\beta_0>,<\beta_1,\beta_2>]$ represents the cumulative score for day 1 and day 2. $P_{\langle\beta_0,\beta_3\rangle}$ represents the individual score for day 3.

$=\beta_{(1,2,....,X)}$. The cumulative score is computed using Equation 8.4.

$$\Delta[<\beta_0>,<\beta_1,\beta_2,....,\beta_X>] = \sum_{i=1}^{k}\sum_{j=1}^{m} Min(Dist(G_i^{miss}, G_j)) \tag{8.4}$$

Where $G_i^{miss} \in S_{miss}^{\beta_0,\beta_{(1,2,....,X)}} = \beta_{(1,2,....,X)} - \beta_0$. The union defined over the profiles considers the profiles as sets and n-grams being the elements of the sets.

The algorithm for score computation, where the baseline profile is available is shown in Algorithm 1. Figure 8.2 shows some scenarios of the score computation with respect to a baseline profile (reference point). In the case of the cold start scenario, the cumulative score for day 1 to day 3 is denoted as $\Delta[<\{\}>,<\beta_1,\beta_2,\beta_3>]$. The cumulative score in the cold start scenario from day 1 to day X is computed as follows: $|\beta_1 \cup \beta_2 \cup, \ldots, \beta_X|$ $\times$ size of n-gram.

8.3.1 Max (worst-case) Score

One, for example the data curator, could be interested to know what could have been the maximum or worst-case score possible? Availability of the worst-case score enables the data curator to make well-informed decisions by comparing the worst-case score with the actual score. For instance, an insignificant difference between the

worst-case score and the actual score is indicative of potentially malicious querying behaviour. The Max (worst-case) score is denoted by $\mathcal{M}_{\beta_X}$. In the proposed model, the maximum score (worst-case score) is calculated using the Equation 8.5.

$$\mathcal{M}_{\beta_X} = |\beta_{\mathcal{R}}| \times n. \tag{8.5}$$

Where n in Equation 8.5 is the size of the n-gram in the run-time profile.

8.3.2 Properties of PriDe

We examined the PriDe function from the point of view of the properties it fulfils. The metric space considers properties, including non-negativeness, identity, symmetry, and triangle inequality, to be fulfilled for a function that defines the distance between two points. Consider the three profiles $\beta_0, \beta_1, and \beta_2$ and PriDe function $P_{\langle,\rangle}$. PriDe function meets the non-negativeness property as the distance between two profiles is positive, i.e. $P_{\langle\beta_0,\beta_1\rangle} \geq 0$. In terms of identity, the distance given by PriDe from a profile to itself is zero, i.e. $P_{\langle\beta_0,\beta_0\rangle} = 0$. For symmetry, the distance between β_0 to β_1 is the same as the distance from β_1 to β_0. The symmetry does not hold for PriDe as $S^{\beta_0,\beta_1}_{miss} = \beta_0 - \beta_1 \neq S^{\beta_1,\beta_0}_{miss}$ $= \beta_1 - \beta_0$, therefore, $P_{\langle\beta_0,\beta_1\rangle} \neq P_{\langle\beta_1,\beta_0\rangle}$. The triangle inequality property requires that the direct distance between β_0 to β_2 is less than or equal to the sum of the distance between β_0 to β_1 and β_1 to β_2. The triangle inequality property holds for PriDe as the distance between β_0 to β_2 will always be less than or equal to the sum of the distance between β_0 to β_1 and β_1 to β_2 i.e. $P_{\langle\beta_0,\beta_2\rangle} \leq P_{\langle\beta_0,\beta_1\rangle} + P_{\langle\beta_1,\beta_2\rangle}$. This is because the property is reflect the computation for cumulative score that is $\Delta[<\beta_0>,<\beta_2>] \leq \Delta[<\beta_0>,<\beta_1>] + \Delta[<\beta_1>,<\beta_2>]$. Therefore, we refer to PriDe as a function that computes score.

8.4 Evaluation

We are interested in evaluating the proposed model in two scenarios. In one scenario we desired an application where generic users (or roles of users) have a standard behaviour, as a result, a baseline profile can be constructed and a contrasting scenario of a cold start. For the first scenario, we considered a synthetic banking application - a transaction-oriented system, as discussed in Section 4.5.1. Reasonably, a user with a role in the bank has similar behaviour to the users with the same role. For the cold start scenario, we considered health-care predictive analytics settings where users looked up information in the hospital database to gain insights. In the cold start scenario, when a new user is appointed, we don't have a baseline profile for their behaviour.

Audit logs for six days, i.e., the audit log of day 0, day 1, day 2, ..., day 5, were generated using the synthetic banking application. The audit log of day 0 was used to generate a baseline profile. The application system was run with 2500 random transactions for each day. In total generating 7200 SQL statements for each audit log.

In the health-care predictive analytics settings, audit logs for day 1, day 2, ..., day 5 were collected, as day 0 was undesired in the cold start scenario. Each audit log consisted of computer-generated queries such that more attributes (combination of attributes) were queried as compared to the attributes (combination of attributes) queried the previous day. The audit log of day 1, day 2, day 3, day 4, and day 5 consisted of 700, 1200, 2200, 3000 and 5500 queries made. We denote the profiles constructed from the audit log of each day as β_{day}. The profiles constructed on day 0, day 1, day 2, day 3, day 4, and day 5 were denoted as $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$, and β_5 for both scenarios. PriDe was deployed in both the scenarios and both the individual score and the cumulative score are shown for both scenarios in Figure 8.3 (with baseline profile) and Figure 8.4 (cold start scenario).

Figure 8.3 shows the scenario of the banking settings. The individual scores in banking

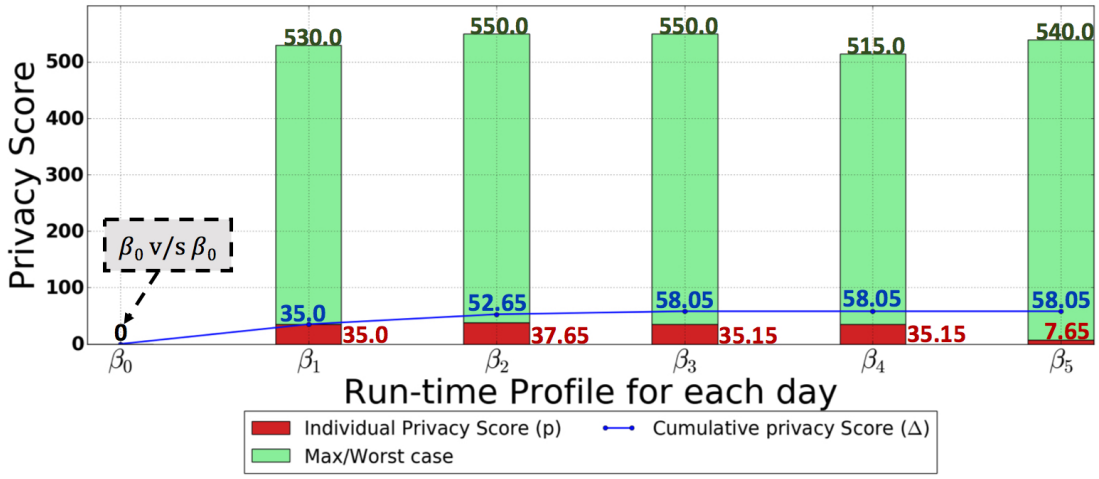


Figure 8.3: This graph shows the results for the scenario of the banking settings. The red bar in the figure shows the actual score while the green bar in the figure indicates the maximum possible score (worst-case score – $\mathcal{M}_{\beta_X}$) for each day. The blue line shows the cumulative score.

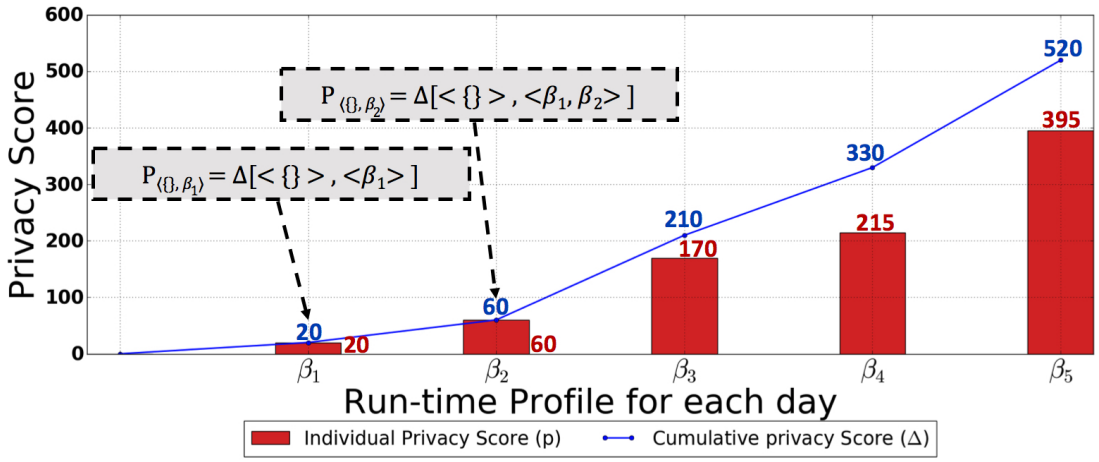


Figure 8.4: This graph shows the results of the cold start scenario. The red bar shows the actual score, and the blue line shows the cumulative score over the time horizon.

settings for day 1, day 2, day 3, day 4, and day 5 are 35.0, 37.65, 35.15, 35.15, and 7.65 are respectively. The red bar in Figure 8.3 shows the actual score while the green bar in Figure 8.3 indicates the max (worst-case) score $\mathcal{M}_{\beta_X}$. These scores for each day indicates that there are newly discovered querying behaviours resulting in a reduction of privacy with respect to the attribute values previously retrieved by the user. However, the score for each day when compared to potential worst-case score is insignificant, particularly, for day 5.

On the other hand, from the cumulative score, one can obtain more meaningful insights. The cumulative score for day 1 indicating the discovery of new querying behaviour resulting in a reduction of privacy; nevertheless, a further reduction in privacy is indicated by a minor increase in cumulative score for day 2. The little increase in cumulative score on day 3 implies that there are further new behaviours discovered apart from the ones discovered on day 1 and 2; therefore, a further reduction in privacy. Identical cumulative scores for day 3, 4, and 5 indicates that at this point the unknown querying behaviours have been discovered; however, these querying behaviours are repeated on day 3, 4, and 5 as indicated by individual scores. This kind of trend was expected because of the nature of the banking application as a set of identical transactions are being repeated daily resulting in less diverse SQL queries, and insignificant individual score each day while a stable cumulative score over several days provided that the baseline profile is well captured. However, an unexpected increase in cumulative score, as well as individual score, is an indication of peculiarities.

It could be the case where the baseline profile has some privacy cost that is not considered at run-time. However, it is worth mentioning that in the availability of baseline profile scenario, the idea is that the privacy cost associated with baseline profile is normative privacy cost in the settings where the PriDe system is deployed.

Figure 8.4 shows the individual and cumulative score in health-care predictive analytics settings. The individual scores for day 1, day 2, day 3, day 4, and day 5 are 20, 60, 170, 215, and 395, respectively. In contrast with banking settings, the baseline profile is not present; additionally, the audit logs do not consist of repeated transactions. In contrast to the cumulative score in banking settings, the cumulative score here is significantly increased each day. The cumulative scores on day 1 and day 2 are same as the individual scores indicating that querying behaviour on day 1 was repeated on day 2 along with new querying behaviours resulting in the reduction of privacy. There is an increasing trend for cumulative score for day 3, day 4, and day 5, these cumulative

score are higher than the corresponding day's individual scores indicating a reduction of privacy with respect to the attribute values retrieved on day 1 and day 2, by the analyst from the database. The increasing trend in cumulative score is the indication that the user is making diverse queries and this is being validated by the cumulative score.

8.4.1 Computational Complexity of PriDe

We look at the complexity of computation of privacy-loss score for each scenario. For the scenario where the baseline profile is available, we look at the complexity for each phase, that is, the phase when the baseline profile is generated and of scoring phase. The baseline profile generation phase has the same level of complexity as of the profile generation phase of the n-gram approach, that is, linear complexity O(n). The scoring phase requires two levels of comparison. The first level of comparison is of comparing the run-time profile with the baseline profile for which we have an algorithm with a linear complexity O(n). At the second level of comparison, the mismatched n-grams arising from this comparison is searched for there closest match and has quadratic complexity $O(n_2)$. The cold start scenario has linear complexity O(n), this is because, in the cold start scenario, the comparison of run-time profile with baseline profile is not carried out due to absence of baseline profile. The cumulative score computation also has linear complexity O(n). At the implementation level, the calculations can be performed in a much more efficient manner when using big data processing techniques like MapReduce [205] for processing large volumes of data. Overall the complexity of computing the score is quadratic $O(n_2)$ provided the identification capabilities of each attributes are already computed. This quadratic complexity is the worst case, that is, the algorithm won't take more than this number of steps as here we use big O notation for upper bounds (worst case) for algorithm analysis.

Algorithm 1: The scoring algorithm takes two profiles as an input and returns a score. The algorithm, *Dist*, computes the similarity between the two n-grams.

Algorithm: Quantitative Privacy Score

input : Two profiles: a baseline profile and a run-time profile
output: A Score

Procedure `PriDe(`$ProfileA$, $ProfileB$`):`
  $pscore \leftarrow 0$
  temp[m] $\leftarrow 0$
  $S^{miss} = ProfileB$ - $ProfileA$ **for** *every ngram 'i' in* S^{miss} **do**
    $k \leftarrow 0$ **for** *every ngram 'j' in* $ProfileA$ **do**
      temp[k] $\leftarrow Dist(i, j)$
      $k \leftarrow k + 1$
    end
    $pscore \leftarrow pscore + min$(temp[$j$])
  end
  return *PriDe*

input : Two ngrams
output: Distance between two n-grams

Procedure `Dist(`*ngram*1, *ngram*2`):`
  $D \leftarrow 0$
  for $l \leftarrow 0$ **to** *length-of-ngram* **do**
    $F \leftarrow 0$
    $E \leftarrow 0$
    F = JaccardD(ngram1[l], ngram2[l]) ▷ ngram1[l] and ngram2[l] is the l_{th} element of the n-grams $E = E + F$
  end
  return E

input : Two SQL Query Abstractions
output: Distance between two SQL query abstractions

Procedure `JaccardD(`*Abs1, Abs2*`):`
  $D \leftarrow 0$
  if *Priveq*($A(Q_1)$, $A(Q_2)$) == *true* ▷ $A(Q_1)$, $A(Q_2)$ *are the SQL query abstractions* **then**
    $D \leftarrow 0$
  else
    $D = (|A(Q_1) \cup A(Q_2)| - |A(Q_1) \cap A(Q_2)|)/|A(Q_1) \cup A(Q_2)|$
  end
  return D

```
input : Two SQL Query Abstractions
output: Distance between two SQL query abstractions
Procedure Priveq(ngram1, ngram2):
    if (Ab2.issubset(Ab1)) == true    ▷ issubset returns a Boolean stating whether the
     set is contained in the specified set.
     then
        if (DR.Ab2 == DR.Ab1)    ▷ DR. give us a precomputed discrimination rate
         value.
         then
            return True
        end
    else
        return False
    end
```

8.4.2 Use-case: Global Consistency

Consider three data analytics companies known as 'Acme Analytics', 'Wayne Analytics', and 'Smart Analytics'. An organisation named 'Ozon enterprise', that focuses on e-commerce, collected a huge amount of data over some period of time. The data consists of the purchases made via their online shopping portal. Motivated by the financial gains, the Ozon enterprise decided to grant access to their application data to all three data analytics companies. Following the defence-in-depth strategy, the Ozon enterprise already had a number of security controls in place. Due to growing concerns over data privacy, this time, the Ozon enterprise management reached a decision to increase a layer by adding a technology, PriDe, that monitor information gain in privacy sense. Granting access to third-parties has been a regular practice by Ozon enterprise, and no privacy incidents took place in past. However, Ozon enterprise has a policy to keep the audit logs of all the sessions of interactive when the access to their data is granted. Using these past logs with PriDe Ozon enterprise integrated PriDe in their privacy and security dashboards. Whereas the third-parties performed their analytics, the Ozon enterprise kept an eye over the scores of these third-parties to have a consistency check to monitor for peculiarities; for instance, if firm *Acme Analytics*'s score is much higher as compared to the rest of the firms, then this alerts the Ozon enterprise

to take imperative measures before any unforeseen breach is materialized.

8.5 Conclusions

In this chapter, we introduced the notion of **Pr**ivacy **D**istanc**e** (PriDe) that measures the privacy-loss between the current querying behaviour from the normative behaviour. Experiments were carried out to evaluate the approach in two scenarios; wherein one scenario, a dataset was available for constructing the baseline profile while the second scenario was of a cold start. Results suggest that PriDe provides with a quantitative score in-terms of privacy that enables organizations to monitor and gain insights about the data that is being shared with a third-party from a privacy perspective, allowing organizations to have more sense of control over their application data. The proposed approach serves as a suitable tool to provide global consistency. In cases where several data analytics firms are simultaneously granted access to the organization's database. Additionally, PriDe can be integrated into a privacy dashboard that provides the health of the system in terms of privacy and enables a check if something is wrong with the way the database is being accessed.

The score computed by PriDe model relies on the similarity metric for comparing SQL query abstractions that in our case, is Jaccard distance. As future work, we plan to explore approaches for comparing two query abstraction besides Jaccard distance for the PriDe model.

Chapter 9

Conclusions and Future Work

"Every end is a new beginning." – Proverb

9.1 Conclusions

Threats to a contemporary organization's data come from external and insider attackers. Traditional security controls sufficiently mitigate external attacks. A more subtle threat comes from insiders, whereby legitimate users of the system abuse the access privileges they hold. Insider threats can be of two types; one is an insider threat to data security (security attacks), and the other is an insider threat to data privacy (privacy attacks). The insider threat to data security means that insider steals or leaks sensitive personal information. The insider threat to data privacy is when the insider access information maliciously that results in the violation of an individuals privacy, for instance, browsing through customers bank account balances or attempting to narrow down to re-identify an individual who has the highest salary with malicious intents. This dissertation addressed data privacy and data security concerns by considering the

research questions stated in Chapter 1.

The first question (RQ1) that is addressed in this dissertation is whether one can build behavioural-based anomaly detection systems by considering the sequence of queries (query correlations) to model insider's querying behaviour to detect malicious accesses manifested in sequences of queries rather than a query in isolation, to DBMS. The dissertation (Chapter 4) proposed a scheme to model querying behaviour of an insider using n-grams that captures short-terms SQL query correlations. The model used abstractions of SQL query audit logs to construct insider profiles, a normative profile using safe logs and a run-time profile using run-time audit logs. The run-time profile is compared with normative profile and deviations are an indication of anomalies. Suitable size for 'n' was determined empirically. The model, in general, is useful in capturing the querying behaviour by considering query sequences, but there is potential for mimicry attack. Therefore, mimicry attacks were also considered and were then mitigated using query analytics-based technique. The dissertation described experiments to judge the effectiveness of using an n-gram based scheme to detect malicious sequences of queries (attacks based on query correlations) as anomalies in DBMS logs and thereby identified insider attacks.

Another question (RQ2) addressed in this dissertation is whether there are alternatives to n-grams for modelling insider querying behaviour. This dissertation looked at the modelling of behaviour from a DBMS's perspective (Chapter 5). A record/DBMS-oriented approach (also referred to as a semantic approach) is proposed that considers frequency-based correlations to detect insiders malicious accesses as anomalies. A notion of DBMS/record-oriented modelling of normative behaviour for construction of normative profiles that considers data-centric features, known as the semantic approach, is also introduced in this dissertation. The construction of the profiles utilizes control charts from statistical process control as a way to detect anomalies. Two scenarios were considered, in the first scenario, the training data for modelling normative

behaviour contains outlier. In the second scenario, the training data for modelling normative behaviour is free from outliers. The experiments demonstrated the effectiveness of the proposed approach in the detection of frequent observation attacks by insiders as anomalies. It was discovered that the semantic approach not only identified unseen behaviours but also identified behaviours that should have been present in the current behaviours as anomalies, which we refer to as *oversight-anomalies*. Oversight-anomalies are the anomalies introduced due to human negligence or human errors, for example, an instance where the doctor or the nurse (caregiver) missed a daily check-up of a patient. To the best of our knowledge, this is the first time the oversight-anomalies are considered in the DBMS setting. It was also demonstrated that the proposed model for the construction of record-oriented profiles could be transformed into a model for the construction of role-oriented profiles.

This dissertation examined further alternative approaches for modelling querying behaviour and introduced the notion that behaviours that are rare (infrequent) represent potentially malicious access by an insider, and frequent behaviours are possibly safe behaviours (Chapter 6). The domain of item-set mining was explored. Item-set mining algorithms including PrePost$^+$, Apriori-Inverse, and Apriori-Rare were adopted to mine frequent and rare query-sets to model querying behaviours. Results point towards the potential effectiveness of modelling insiders malicious querying behaviour as rare behaviour that also enables detection of insider's malicious databases accesses as anomalies.

The SQL query-based insider attacks considered in Chapter 4, 5, and 6, are referred to as security attacks, and these security attacks that were detected by the proposed approaches as anomalies were referred to as security-anomalies. As these approaches were considering security attacks and detecting them as anomalies, these approaches are referred to as security-anomaly detection approaches.

This dissertation also considers the question (RQ3) whether a privacy semantics for

these behavioural-based anomaly detection approaches can be provided or the notion of privacy-anomaly detection can be related to the conventional (formal) definitions of privacy semantics such as *k*-anonymity and the discrimination rate privacy metric. It was discovered that the anomalies identified by the n-gram based security-anomaly detection; some of those anomalies were a type of privacy violations. This dissertation proposed the notion of '*Privacy-Anomaly Detection*' (Chapter 7). The idea is to learn privacy criteria from past interactions (audit logs) and uses this criteria to check whether the current querying behaviour is different from past behaviour with respect to privacy. An instantiation of a privacy-anomaly detection system using *k*-anonymity referred to as (*k*-anonymity)-PAD was demonstrated. The privacy attacks considered initially for testing instantiations of (*k*-anonymity)-PAD were based on a single SQL query in isolation. It was shown that privacy-anomaly detection systems could be constructed by composing multiple existing formal privacy definitions and an instantiation using the composition of *k*-anonymity and discrimination rate privacy metric referred to as (*k*-anonymity, DR)-PAD was designed.

A spin-off of this work was a translation of the discrimination rate privacy metric that can be used to measure the identification capability of SQL queries (*DRSQL*). Chapter 7 describes how it can be used as a stand-alone metric to measure the identification capability of SQL queries and enables a privacy comparison of SQL queries. The majority of existing work on the comparison of SQL queries compares SQL queries syntactically. The proposed work on comparing SQL queries in terms of privacy provides a foundation on this topic.

Privacy attacks, violations of formal privacy definition, based on a sequence of SQL queries (query correlations) are also considered (Chapter 7). It is shown that interactive querying settings are vulnerable to privacy attacks based on query correlation. Investigation on whether these types of privacy attacks can potentially manifest themselves as anomalies, specifically as privacy-anomalies was carried out. A promising

result is that privacy attacks (violation of formal privacy definition) can be detected as privacy-anomalies by applying behavioural-based anomaly detection using n-gram over the logs of interactive querying mechanisms.

Lastly, Chapter 8 introduced the notion of privacy score and proposed *PriDe* a technique to quantitatively measures the privacy-loss in the form of a single score within the framework of the relational database management system. A strategy for a refined comparison of n-grams (query and query abstraction) in terms of privacy equivalence enables the computation of a single score. Experiments were carried out to evaluate the approach in two scenarios; wherein one scenario, a dataset was available for constructing the baseline (normative) profile while the second scenario was of a cold start (absence of baseline profile). Results suggests that PriDe is a step toward the quantification behavioural changes in-terms of privacy that enables organizations to monitor and gain insights about the data that is being shared with a third-party from a privacy perspective. PriDe allows organizations to have more sense of control over their application data and serves as an excellent tool to provide global consistency.

9.2 Future Work

This section outlined the planned future work that stems from this dissertation and consider challenges that were identified from the results.

9.2.1 Similarity Index for SQL Statements

This dissertation relies heavily on the ability to compare query abstractions. In this work, the abstractions were compared based on the syntax. The questions that arise are whether one considers semantics or just syntax while comparing two queries or query abstractions? Is an exact match required or is an approximation sufficient? Answers to these questions are desirable. Comparing two statements based only on the

syntax is challenging because of the inherent flexibility of SQL. Two SQL statements can be written differently though resulting in the same output. The literature lacks a query similarity index that measures how close two queries are syntactically as well as semantically.

Aditionally, a recently proposed approach [166] of query regularization is an interesting avenue to be explored, thus augmenting the query abstractions. The idea of query regularization is to regularizing SQL queries syntactically which are semantically similar. The query regularization approach can potentially improve the SQL query abstraction part of the presented approaches in this dissertation.

A new topic of research is introduced in this dissertation (Chapter 7) where a technique for comparing SQL queries in terms of privacy instead of semantics or syntax is proposed.

9.2.2 Scarcity of Real-world or Benchmark Datasets

During the course of our work it became evident that there is a lack of standard benchmark datasets and this impacts the veracity of results in the area. Because of this lack of real-world datasets, it is hard to assess the performance of a system in order to enable a comparison with the performances of other systems. This challenge is not only for the 'detection of malicious DBMS access' research but for the insider threat detection in general [206].

9.2.3 Handling Concept Drift in Behaviours

Concept drift, in the context of modelling querying behaviours, is not generally considered in research literature. Concept drift, in machine learning and data-mining, is defined as the *unexpected changes in underlying data distribution over time* [207, 208, 209, 210]. In the context of modelling querying behaviour, it is pos-

sible that the behaviour changes after the passage of time. That is to say that it is possible that behaviour that was legitimate last year and us malicious behaviour today. A straight forward approach to handle concept drift is to update the model regularly, thus, the normative profiles need to be generated periodically. This is a relatively new challenge in this line of research.

The semantic approach proposed in Chapter 5 (Section 5.4, Figure 5.3) captures the drift of normative behaviour overtime by relearning normative from current behaviour. Concept drift, in general, is a broad topic to be covered here. Approaches to handle concept drift tailored to modelling user's querying behaviour, such that as used in other domains [207, 208, 209, 210], is a topic for future research.

9.2.4 Defence in Depth: Privacy Perspective

Stronger notions of privacy definitions constrain data utility in interactive query settings, while weaker notions of privacy in an interactive query setting may result in privacy attacks (inferences and violations of formal privacy definitions). It is worth noting that the privacy attacks can manifest itself in a variety of unexpected ways. A way forward is to utilise defence in depth in privacy controls. For future work, we plan to compose various privacy controls, including privacy-anomaly detection system, security-anonymity detection system, and privacy score for a defence in depth approach from a privacy perspective.

9.2.5 Translation onto Other Data Models

The research work in this dissertation is based on a relational data model. Future research should consider how the techniques - n-gram based approach, DBMS/record-oriented model, the item-set mining-based approach, privacy-anomaly detection systems and PriDe - might be applied to other data models, for instance, deductive

databases and object-oriented databases. This requires, for example, determining the right level of abstraction for OQL statements in object oriented database [211].

9.2.6 Explaining Anomalies

Explaining anomalies is not only a challenge within the context of the work carried out in this dissertation, but it is a challenge in general in anomaly detection domain. In order to generate explanations for anomalies, systems that act as white-box are desirable, which will enable humans to easily comprehend why the anomaly is flagged off and subsequently generate an explanation for that anomaly. In the case where an anomaly is flagged, there are all set of possible things that could be wrong. A question that arises is which one of them is used to give an explanation about the anomaly and what will the simplest explanation. For instance, in the case where k-anonymity-based PAD flags an anomaly, a possible explanation is that there is a privacy issue because the query is narrowing down to less than k individuals. In order to form an explanation, one has to look at the available information in case of an anomaly in our case; this information includes the value of k, collection of violated n-gram, frequencies that were violated. It would be interesting to see how existing anomaly explanation techniques [212, 213, 214] would be used to provide a more understandable explanation for the proposed detection systems in this dissertation.

9.2.7 Instantiation of PAD with Multiple Privacy Models

In future work, an exploration of PAD with other variations of multiple privacy model is desirable. An exposition using hierarchical constraints framework to compose multiple privacy criteria will be an interesting line of research. Where an investigation into the trade-off between multiple privacy criteria, intuitively, improve privacy guarantees.

Bibliography

[1] R. J. Santos, J. Bernardino, M. Vieira, and D. M. L. Rasteiro, "Securing data warehouses from web-based intrusions," in *Web Information Systems Engineering - WISE 2012* (X. S. Wang, I. Cruz, A. Delis, and G. Huang, eds.), (Berlin, Heidelberg), pp. 681–688, Springer Berlin Heidelberg, 2012.

[2] "Data protection and privacy legislation worldwide," 2020. UNCTAD Report, Online at: https://unctad.org/en/Pages/DTL/STI_and_ICTs/ICT4D-Legislation/eCom-Data-Protection-Laws.aspx.

[3] S. R. Hussain, A. M. Sallam, and E. Bertino, "Detanom: Detecting anomalous database transactions by insiders," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, (New York, NY, USA), pp. 25–35, ACM, 2015.

[4] "2016 cost of data breach study: Global analysis," tech. rep., Ponemon Institute, 2016. Online at: https://www.ibm.com/downloads/cas/7VMK5DV6.

[5] "2015 vormetric insider threat report," tech. rep., Vormetric, 2015. Online at: http://go.thalesesecurity.com/rs/480-LWA-970/images/2015_Vormetric_ITR_European_R3.pdf.

[6] "Grand theft data data exfiltration study: Actors, tactics, and detection," tech. rep., Intel security and McAfee, 2015. Online at: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-data-exfiltration.pdf.

[7] "2016 data breach investigations report," tech. rep., Verizon, 2016. Online at https://conferences.law.stanford.edu/cyberday/wp-content/uploads/sites/10/2016/10/2b_Verizon_Data-Breach-Investigations-Report_2016_Report_en_xg.pdf.

[8] M. I. Khan and S. N. Foley, "Detecting anomalous behavior in DBMS logs," in *Risks and Security of Internet and Systems - 11th International Conference, CRiSIS 2016, Roscoff, France, September 5-7, 2016, Revised Selected Papers* (F. Cuppens, N. Cuppens, J. Lanet, and A. Legay, eds.), vol. 10158 of *Lecture Notes in Computer Science*, pp. 147–152, Springer, 2016.

[9] G. Kul, D. Luong, T. Xie, P. Coonan, V. Chandola, O. Kennedy, and S. Upadhyaya, "Ettu: Analyzing query intents in corporate databases," in *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, (Republic and Canton of Geneva, Switzerland), pp. 463–466, International World Wide Web Conferences Steering Committee, 2016.

[10] A. Sallam, D. Fadolalkarim, E. Bertino, and Q. Xiao, "Data and syntax centric anomaly detection for relational databases," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 6, pp. 231–239, 2016.

[11] S. Mathew, M. Petropoulos, H. Q. Ngo, and S. Upadhyaya, "A data-centric approach to insider attack detection in database systems," in *Proceedings of the 13th International Conference on Recent Advances in Intrusion Detection*, RAID'10, (Berlin, Heidelberg), pp. 382–401, Springer-Verlag, 2010.

[12] E. Bertino, E. Terzi, A. Kamra, and A. Vakali, "Intrusion detection in rbac-administered databases," in *21st Annual Computer Security Applications Conference (ACSAC'05)*, pp. 10 pp.–182, Dec 2005.

[13] E. Costante, J. den Hartog, M. Petković, S. Etalle, and M. Pechenizkiy, "Hunting the unknown," in *Data and Applications Security and Privacy XXVIII* (V. Atluri

and G. Pernul, eds.), (Berlin, Heidelberg), pp. 243–259, Springer Berlin Heidelberg, 2014.

[14] J. S. Oakland, *Statistical Process Control (Sixth Edition)*. Butterworth-Heinemann, 2008.

[15] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," *Official Journal of the European Union*, vol. L119, pp. 1–88, May 2016.

[16] "Anonymisation and pseudonymisation," tech. rep., Data Protection Commission, Ireland, 2019. Online at: https://www.dataprotection.ie/en/guidance-landing/anonymisation-and-pseudonymisation.

[17] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (SP' 08)*, (Los Alamitos, CA, USA), pp. 111–125, IEEE Computer Society, may 2008.

[18] M. Barbaro and T. Z. Jr., "A face is exposed for aol searcher no. 4417749." The New York Times. Online at: http://www.nytimes.com/2006/08/09/technology/09aol.html?mcubz=2.

[19] A. Hern, "New york taxi details can be extracted from anonymised data, researchers say," 2014. The Guardian. Online at: https://www.theguardian.com/technology/2014/jun/27/new-york-taxi-details-anonymised-data-researchers-warn.

[20] L. Sweeney, "Simple demographics often identify people uniquely," working paper, 2000. Working paper. Online at: http://dataprivacylab.org/projects/identifiability/.

[21] V. Torra and G. Navarro-Arribas, "Big data privacy and anonymization," in *Privacy and Identity Management. Facing up to Next Steps - 11th IFIP WG 9.2, 9.5, 9.6/11.7, 11.4, 11.6/SIG 9.2.2 International Summer School, Karlstad, Sweden, August 21-26, 2016, Revised Selected Papers* (A. Lehmann, D. Whitehouse, S. Fischer-Hübner, L. Fritsch, and C. D. Raab, eds.), vol. 498 of *IFIP Advances in Information and Communication Technology*, pp. 15–26, 2016.

[22] V. Torra and G. Navarro-Arribas, "Data privacy: A survey of results," in *Advanced Research in Data Privacy* (G. Navarro-Arribas and V. Torra, eds.), vol. 567 of *Studies in Computational Intelligence*, pp. 27–37, Springer, 2015.

[23] V. Torra and G. Navarro-Arribas, "Data privacy," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 4, no. 4, pp. 269–280, 2014.

[24] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[25] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*, pp. 106–115, April 2007.

[26] C. Clifton and T. Tassa, "On syntactic anonymity and differential privacy," in *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, pp. 88–93, April 2013.

[27] J. Soria-Comas, "Improving data utility in differential privacy and k-anonymity," *CoRR*, vol. abs/1307.0966, 2013.

[28] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *Proceedings 1996 IEEE Symposium on Security and Privacy*, pp. 120–128, May 1996.

[29] A. Somayaji and S. Forrest, "Automated response using system-call delays," in *Proceedings of the 9th Conference on USENIX Security Symposium - Volume 9*, SSYM'00, (Berkeley, CA, USA), pp. 14–14, USENIX Association, 2000.

[30] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *J. Comput. Secur.*, vol. 6, pp. 151–180, Aug. 1998.

[31] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 303–336, First 2014.

[32] Q. Zhao, Y. Zhang, Y. Shi, and J. Li, "Analyzing and visualizing anomalies and events in time series of network traffic," in *Recent Advances in Information and Communication Technology 2019* (P. Boonyopakorn, P. Meesad, S. Sodsee, and H. Unger, eds.), (Cham), pp. 15–25, Springer International Publishing, 2020.

[33] Z. Li, A. L. G. Rios, G. Xu, and L. Trajkovic, "Machine learning techniques for classifying network anomalies and intrusions," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, May 2019.

[34] C. Y. Chung, M. Gertz, and K. N. Levitt, "DEMIDS: A misuse detection system for database systems," in *Integrity and Internal Control in Information Systems, IFIP TC11 Working Group 11.5, Third Working Conference on Integrity and Internal Control in Information Systems: Strategic Views on the Need for Control, Amsterdam, The Netherlands, November 18-19, 1999* (M. E. van Biene-Hershey and L. Strous, eds.), vol. 165 of *IFIP Conference Proceedings*, pp. 159–178, Kluwer, 1999.

[35] M. Alizadeh, S. Peters, S. Etalle, and N. Zannone, "Behavior analysis in the medical sector: Theory and practice," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, SAC '18, (New York, NY, USA), pp. 1637–1646, ACM, 2018.

[36] M. Gafny, A. Shabtai, L. Rokach, and Y. Elovici, "Poster: Applying unsupervised context-based analysis for detecting unauthorized data disclosure," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, (New York, NY, USA), pp. 765–768, ACM, 2011.

[37] Y. S. Koh and N. Rountree, "Finding sporadic rules using apriori-inverse," in *Advances in Knowledge Discovery and Data Mining* (T. B. Ho, D. Cheung, and H. Liu, eds.), (Berlin, Heidelberg), pp. 97–106, Springer Berlin Heidelberg, 2005.

[38] L. Szathmary, A. Napoli, and P. Valtchev, "Towards rare itemset mining," in *19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)*, vol. 1, pp. 305–312, Oct 2007.

[39] Z.-H. Deng and S.-L. Lv, "Prepost+: An efficient n-lists-based algorithm for mining frequent itemsets via children-parent equivalence pruning," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5424 – 5432, 2015.

[40] L. P. Sondeck, M. Laurent, and V. Frey, "Discrimination rate: an attribute-centric metric to measure privacy," *Annals of Telecommunications*, vol. 72, pp. 755–766, Dec 2017.

[41] S. N. Chari and P.-C. Cheng, "Bluebox: A policy-driven, host-based intrusion detection system," *ACM Trans. Inf. Syst. Secur.*, vol. 6, pp. 173–200, May 2003.

[42] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian, "Flexible support for multiple access control policies," *ACM Trans. Database Syst.*, vol. 26, pp. 214–260, June 2001.

[43] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati, "A temporal access control mechanism for database systems," *IEEE Trans. on Knowl. and Data Eng.*, vol. 8, pp. 67–80, Feb. 1996.

[44] M. Bishop and C. Gates, "Defining the insider threat," in *Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research: Developing Strategies to Meet the Cyber Security and Information Intelligence Challenges Ahead*, CSIIRW '08, (New York, NY, USA), pp. 15:1–15:3, ACM, 2008.

[45] M. Bishop, D. Gollmann, J. Hunker, and C. W. Probst, eds., *Countering Insider Threats, 20.07. - 25.07.2008*, vol. 08302 of *Dagstuhl Seminar Proceedings*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2008.

[46] J. R. C. Nurse, O. Buckley, P. A. Legg, M. Goldsmith, S. Creese, G. R. T. Wright, and M. Whitty, "Understanding insider threat: A framework for characterising attacks," in *2014 IEEE Security and Privacy Workshops*, pp. 214–228, May 2014.

[47] J. Hunker and C. W. Probst, "Insiders and insider threats - an overview of definitions and mitigation techniques," *Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications*, pp. 4–27, 2011.

[48] R. C. Brackney and R. H. Anderson, *Understanding the insider threat: Proceedings of a march 2004 workshop*, vol. 196. Rand Corporation, 2004.

[49] J. Patzakis, "New incident response best practices: Patch and proceed is no longer acceptable incident response procedure," tech. rep., Guidance Software, Pasadena, CA.

[50] "2018 insider threat report," tech. rep., ca Technologies, 2018. Online at: `https://crowdresearchpartners.com/wp-content/uploads/2017/07/Insider-Threat-Report-2018.pdf`.

[51] S. L. Pfleeger, J. B. Predd, J. Hunker, and C. Bulford, "Insiders behaving badly: Addressing bad actors and their actions," *Trans. Info. For. Sec.*, vol. 5, pp. 169–179, Mar. 2010.

[52] "2015 cost of cyber crime: Global," tech. rep., Ponemon Institute, 2015. Online at: http://www.cnmeonline.com/myresources/hpe/docs/HPE_SIEM_Analyst_Report_-_2015_Cost_of_Cyber_Crime_Study_-_Global.pdf.

[53] "Cybersecurity snapshot global results," tech. rep., ISACA, 2016.

[54] B. Brenner, "Healthcare data breaches mostly caused by insiders," 2017. Naked Security by Sophos. Online at: https://nakedsecurity.sophos.com/2017/02/23/healthcare-data-breaches-mostly-caused-by-insiders/.

[55] "Security trends in the healthcare industry data theft and ransomware plague healthcare organizations," tech. rep., IBM Security, IBM, 2016. Online at: https://www.ibm.com/downloads/cas/PLWZ76MM.

[56] "2014 US state of cybercrime survey," tech. rep., CERT, Software Engineering Institute, Carnegie Mellon University, 2014. Online at: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=298318.

[57] "Privacy amendment (notifiable data breaches) act 2017," 2017. Online at: https://www.legislation.gov.au/Details/C2017A00012.

[58] R. A. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," *Computer*, vol. 35, pp. 27–30, Apr 2002.

[59] S. Kumar and E. H. Spafford, "A pattern matching model for misuse intrusion detection," in *In Proceedings of the 17th National Computer Security Conference*, pp. 11–21, 1994.

[60] P. Ning and S. Jajodia, *Intrusion Detection Techniques*. American Cancer Society, 2004.

[61] F. Anjum, D. Subhadrabandhu, and S. Sarkar, "Signature based intrusion detection for wireless ad-hoc networks: a comparative study of various routing protocols," in *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484)*, vol. 3, pp. 2152–2156 Vol.3, Oct 2003.

[62] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *IEEE Wireless Communications*, vol. 11, pp. 48–60, Feb 2004.

[63] T. F. Lunt, R. Jagannathan, R. Lee, A. Whitehurst, and S. Listgarten, "Knowledge-based intrusion detection," in *[1989] Proceedings. The Annual AI Systems in Government Conference*, pp. 102–107, March 1989.

[64] Trustwave, "DbProtect." Online at: `https://www.trustwave.com/en-us/services/security-testing/dbprotect/`.

[65] BeyondTrust, "PowerBroker for Databases." Online at: `https://www.beyondtrust.com/resources/brochures/powerbroker-for-databases`.

[66] IBM, "Guardium." Online at: `http://www-01.ibm.com/software/data/guardium/`.

[67] A. Lazarevic, V. Kumar, and J. Srivastava, *Intrusion Detection: A Survey*, pp. 19–78. Boston, MA: Springer US, 2005.

[68] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," *IEEE Trans. Comput.*, vol. 63, pp. 807–819, Apr. 2014.

[69] C. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: network intrusion detection and countermeasure selection in virtual network systems," *IEEE Trans. Dependable Sec. Comput.*, vol. 10, no. 4, pp. 198–211, 2013.

[70] A. S. Abed, T. C. Clancy, and D. S. Levy, "Applying bag of system calls for anomalous behavior detection of applications in linux containers," in *2015 IEEE Globecom Workshops, San Diego, CA, USA, December 6-10, 2015*, pp. 1–5, 2015.

[71] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: A new approach for detecting network intrusions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, (New York, NY, USA), pp. 265–274, ACM, 2002.

[72] C. Ko, M. Ruschitzka, and K. Levitt, "Execution monitoring of security-critical programs in distributed systems: a specification-based approach," in *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097)*, pp. 175–187, May 1997.

[73] D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, (New York, NY, USA), pp. 255–264, ACM, 2002.

[74] A. Tang, S. Sethumadhavan, and S. J. Stolfo, "Unsupervised anomaly-based malware detection using hardware features," in *Research in Attacks, Intrusions and Defenses* (A. Stavrou, H. Bos, and G. Portokalidis, eds.), (Cham), pp. 109–129, Springer International Publishing, 2014.

[75] J. E. Tapiador and J. A. Clark, "Masquerade mimicry attack detection: A randomised approach," *Computers & Security*, vol. 30, no. 5, pp. 297 – 310, 2011. Advances in network and system security.

[76] J. Bouche, D. Hock, and M. Kappes, "On the performance of anomaly detection systems uncovering traffic mimicking covert channels," in *Eleventh International Network Conference, INC 2016, Frankfurt, Germany, July 19-21, 2016.*

Proceedings, pp. 19–24, 2016.

[77] C. Kruegel, E. Kirda, D. Mutz, W. Robertson, and G. Vigna, "Automating mimicry attacks using static binary analysis," in *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14*, SSYM'05, (Berkeley, CA, USA), pp. 11–11, USENIX Association, 2005.

[78] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, (New York, NY, USA), pp. 275–283, ACM, 2000.

[79] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 266–282, First 2014.

[80] D. W. Parter, ed., *Proceedings of the 13th Conference on Systems Administration (LISA-99), Seattle, WA, USA, November 7-12, 1999*, USENIX, 1999.

[81] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1, pp. 18 – 28, 2009.

[82] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," *IEEE Transactions on Computers*, vol. 63, pp. 807–819, April 2014.

[83] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: Behavior-based malware detection system for android," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '11, (New York, NY, USA), pp. 15–26, ACM, 2011.

[84] M. Caselli, E. Zambon, and F. Kargl, "Sequence-aware intrusion detection in industrial control systems," in *Proceedings of the 1st ACM Workshop on Cyber-*

Physical System Security, CPSS '15, (New York, NY, USA), pp. 13–24, ACM, 2015.

[85] Chi-Ho Tsang and S. Kwong, "Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction," in *2005 IEEE International Conference on Industrial Technology*, pp. 51–56, Dec 2005.

[86] R. R. R. Barbosa and A. Pras, "Intrusion detection in scada networks," in *Mechanisms for Autonomous Management of Networks and Services* (B. Stiller and F. De Turck, eds.), (Berlin, Heidelberg), pp. 163–166, Springer Berlin Heidelberg, 2010.

[87] Wei Gao, T. Morris, B. Reaves, and D. Richey, "On scada control system command and response injection and intrusion detection," in *2010 eCrime Researchers Summit*, pp. 1–9, Oct 2010.

[88] G. Z. Wu, S. L. Osborn, and X. Jin, *Database Intrusion Detection Using Role Profiling with Role Hierarchy*, pp. 33–48. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[89] W. L. Low, J. Lee, and P. Teoh, "DIDAFIT: detecting intrusions in databases through fingerprinting transactions," in *ICEIS 2002, Proceedings of the 4st International Conference on Enterprise Information Systems, Ciudad Real, Spain, April 2-6, 2002*, pp. 121–128, 2002.

[90] S. Y. Lee, W. L. Low, and P. Y. Wong, "Learning fingerprints for a database intrusion detection system," in *Proceedings of the 7th European Symposium on Research in Computer Security*, ESORICS '02, (London, UK, UK), pp. 264–280, Springer-Verlag, 2002.

[91] A. Kamra, E. Terzi, and E. Bertino, "Detecting anomalous access patterns in relational databases," *The VLDB Journal*, vol. 17, pp. 1063–1077, Aug. 2008.

[92] A. Sallam, E. Bertino, S. R. Hussain, D. Landers, R. M. Lefler, and D. Steiner, "Dbsafe:an anomaly detection system to protect databases from exfiltration attempts," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–11, 2015.

[93] G. Kul, D. T. A. Luong, T. Xie, V. Chandola, O. Kennedy, and S. Upadhyaya, "Similarity metrics for sql query clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, pp. 2408–2420, Dec 2018.

[94] R. Majumdar and K. Sen, "Hybrid concolic testing," in *Proceedings of the 29th International Conference on Software Engineering*, ICSE '07, (Washington, DC, USA), pp. 416–426, IEEE Computer Society, 2007.

[95] K. Sen, "Concolic testing," in *Proceedings of the Twenty-second IEEE/ACM International Conference on Automated Software Engineering*, ASE '07, (New York, NY, USA), pp. 571–572, ACM, 2007.

[96] K. Sen, D. Marinov, and G. Agha, "Cute: A concolic unit testing engine for c," in *Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ESEC/FSE-13, (New York, NY, USA), pp. 263–272, ACM, 2005.

[97] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning*, vol. 2, pp. 139–172, Sep 1987.

[98] J. H. Gennari, P. Langley, and D. Fisher, "Models of incremental concept formation," *Artificial Intelligence*, vol. 40, no. 1, pp. 11 – 61, 1989.

[99] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, pp. 207–216, June 1993.

[100] F. Schoeman, "Privacy: Philosophical dimensions," *American Philosophical Quarterly*, vol. 21, no. 3, pp. 199–213, 1984.

[101] M. Zuckerman, "Privacy in colonial new england. by david h. flaherty.," *Journal of American History*, vol. 59, no. 3, pp. 679–681, 1972.

[102] O. M. Reynolds *Administrative Law Review*, vol. 22, no. 1, pp. 101–106, 1969.

[103] P. G. Polyviou, *Search & seizure : constitutional and common law Polyvios G. Polyviou*. Duckworth London, 1982.

[104] W. A. Parent, "Privacy, morality, and the law," *Philosophy & Public Affairs*, vol. 12, no. 4, pp. 269–288, 1983.

[105] C. Fried, "Privacy," *The Yale Law Journal*, vol. 77, no. 3, pp. 475–493, 1968.

[106] L. Henkin, "Privacy and autonomy," *Columbia Law Review*, vol. 74, no. 8, pp. 1410–1433, 1974.

[107] R. Gavison, "Privacy and the limits of law," *The Yale Law Journal*, vol. 89, no. 3, pp. 421–471, 1980.

[108] C. H. Pyle, "Privacy, law, and public policy. by david m. obrien. (new york: Praeger publishers, 1979.)," *American Political Science Review*, vol. 75, no. 1, pp. 206–207, 1981.

[109] *Universal Declaration of Human Rights*. December 1948. Online at: `https://www.ohchr.org/EN/UDHR/Documents/UDHR_Translations/eng.pdf`.

[110] Centers for Medicare & Medicaid Services, "The Health Insurance Portability and Accountability Act of 1996 (HIPAA)." Online at http://www.cms.hhs.gov/hipaa/, 1996.

[111] "2018 reform of EU data protection rules," 2018. Online at: `https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf`.

[112] OECD, "The OECD privacy framework," tech. rep., OECD Publishing, 2013. Online at: https://www.oecd.org/sti/ieconomy/oecd_privacy_framework.pdf.

[113] X. Yang, C. M. Procopiuc, and D. Srivastava, "Recommending join queries via query log analysis," in *2009 IEEE 25th International Conference on Data Engineering*, pp. 964–975, March 2009.

[114] T. Dalenius, "Towards a methodology for statistical disclosure control," *Statistik Tidskrift*, vol. 15, no. 429-444, pp. 2–1, 1977.

[115] J. Fogel and E. Nehmad, "Internet social network communities: Risk taking, trust, and privacy concerns," *Computers in Human Behavior*, vol. 25, no. 1, pp. 153 – 160, 2009.

[116] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: An online social network with user-defined privacy," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, SIGCOMM '09, (New York, NY, USA), pp. 135–146, ACM, 2009.

[117] L. A. Cutillo, R. Molva, and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust," *IEEE Communications Magazine*, vol. 47, pp. 94–101, Dec 2009.

[118] S. Sicari, A. Rizzardi, L. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146 – 164, 2015.

[119] A. Ukil, S. Bandyopadhyay, and A. Pal, "Iot-privacy: To be private or not to be private," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 123–124, April 2014.

[120] P. Porambage, M. Ylianttila, C. Schmitt, P. Kumar, A. Gurtov, and A. V. Vasilakos, "The quest for privacy in the internet of things," *IEEE Cloud Computing*,

vol. 3, pp. 36–45, Mar.-Apr. 2016.

[121] A. Gurung and M. Raja, "Online privacy and security concerns of consumers," *Information and Computer Security*, vol. 24, no. 4, pp. 348–371, 2016.

[122] G. R. Milne, A. J. Rohm, and S. Bahl, "Consumers' protection of online privacy and identity," *The Journal of Consumer Affairs*, vol. 38, no. 2, pp. 217–232, 2004.

[123] G. Antoniou and L. Batten, "E-commerce: protecting purchaser privacy to enforce trust," *Electronic Commerce Research*, vol. 11, p. 421, Aug 2011.

[124] E. F. Codd, "A relational model of data for large shared data banks," *Commun. ACM*, vol. 13, no. 6, pp. 377–387, 1970.

[125] A. Watt and N. Eng, *Database Design*. BCcampus, 2014. Online at: `https://opentextbc.ca/dbdesign01/`.

[126] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation* (M. Agrawal, D. Du, Z. Duan, and A. Li, eds.), (Berlin, Heidelberg), pp. 1–19, Springer Berlin Heidelberg, 2008.

[127] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discov. Data*, vol. 1, Mar. 2007.

[128] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang, "(α, k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, (New York, NY, USA), pp. 754–759, ACM, 2006.

[129] P. Zhao, J. Li, F. Zeng, F. Xiao, C. Wang, and H. Jiang, "Illia: Enabling*k*-anonymity-based privacy preserving against location injection attacks in con-

tinuous lbs queries," *IEEE Internet of Things Journal*, vol. 5, pp. 1033–1042, April 2018.

[130] Y.-M. Ye, C.-C. Pan, and G.-K. Yang, "An improved location-based service authentication algorithm with personalized k-anonymity," in *China Satellite Navigation Conference (CSNC) 2016 Proceedings: Volume I* (J. Sun, J. Liu, S. Fan, and F. Wang, eds.), (Singapore), pp. 257–266, Springer Singapore, 2016.

[131] Y. Zhang, W. Tong, and S. Zhong, "On designing satisfaction-ratio-aware truthful incentive mechanisms for*k*-anonymity location privacy," *IEEE Transactions on Information Forensics and Security*, vol. 11, pp. 2528–2541, Nov 2016.

[132] Y. Wang, Z. Cai, Z. Chi, X. Tong, and L. Li, "A differentially k-anonymity-based location privacy-preserving for mobile crowdsourcing systems," in *2017 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2017, Shandong, China, October 19-21, 2017* (R. Bie, Y. Sun, and J. Yu, eds.), vol. 129 of *Procedia Computer Science*, pp. 28–34, Elsevier, 2017.

[133] S. Zhong, H. Zhong, X. Huang, P. Yang, J. Shi, L. Xie, and K. Wang, *Connecting Things to Things in Physical-World: Security and Privacy Issues in Vehicular Ad-hoc Networks*, pp. 101–134. Cham: Springer International Publishing, 2019.

[134] Y. Khazbak, J. Fan, S. Zhu, and G. Cao, "Preserving location privacy in ride-hailing service," in *2018 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9, May 2018.

[135] D. Di Castro, L. Lewin-Eytan, Y. Maarek, R. Wolff, and E. Zohar, "Enforcing k-anonymity in web mail auditing," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, (New York, NY, USA), pp. 327–336, ACM, 2016.

[136] J. Ali, "Mechanism for the prevention of password reuse through anonymized hashes," *PeerJ PrePrints*, vol. 5, p. e3322, 2017.

[137] J. Domingo-Ferrer and V. Torra, "A critique of k-anonymity and some of its enhancements," in *2008 Third International Conference on Availability, Reliability and Security*, pp. 990–993, March 2008.

[138] H. Liang and H. Yuan, "On the complexity of t-closeness anonymization and related problems," in *Database Systems for Advanced Applications* (W. Meng, L. Feng, S. Bressan, W. Winiwarter, and W. Song, eds.), (Berlin, Heidelberg), pp. 331–345, Springer Berlin Heidelberg, 2013.

[139] B. C. Fung, K. Wang, A. W.-C. Fu, and P. S. Yu, *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Chapman & Hall/CRC, 1st ed., 2010.

[140] X. Xiao and Y. Tao, "M-invariance: Towards privacy preserving re-publication of dynamic datasets," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, (New York, NY, USA), pp. 689–700, ACM, 2007.

[141] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu, "Aggregate query answering on anonymized tables," in *2007 IEEE 23rd International Conference on Data Engineering*, pp. 116–125, April 2007.

[142] J. Li, Y. Tao, and X. Xiao, "Preservation of proximity privacy in publishing numerical sensitive data," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, (New York, NY, USA), pp. 473–486, ACM, 2008.

[143] M. E. Nergiz, C. Clifton, and A. E. Nergiz, "Multirelational k-anonymity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1104–1117, Aug 2009.

[144] J. Brickell and V. Shmatikov, "The cost of privacy: destruction of data-mining utility in anonymized data publishing," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pp. 70–78, 2008.

[145] Q. Geng and P. Viswanath, "The optimal mechanism in differential privacy," in *2014 IEEE International Symposium on Information Theory*, pp. 2371–2375, June 2014.

[146] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103, Oct 2007.

[147] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor, "Optimizing linear counting queries under differential privacy," in *Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '10, (New York, NY, USA), pp. 123–134, ACM, 2010.

[148] M. Hardt, K. Ligett, and F. McSherry, "A simple and practical algorithm for differentially private data release," *CoRR*, vol. abs/1012.4763, 2010.

[149] J. He and L. Cai, "Differential private noise adding mechanism: Fundamental theory and its application," *CoRR*, vol. abs/1611.08936, 2016.

[150] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, pp. 211–407, Aug. 2014.

[151] S. L. Garfinkel, J. M. Abowd, and S. Powazek, "Issues encountered deploying differential privacy," in *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, WPES'18, (New York, NY, USA), pp. 133–137, ACM, 2018.

[152] J. Aligon, M. Golfarelli, P. Marcel, S. Rizzi, and E. Turricchia, "Similarity measures for olap sessions," *Knowledge and Information Systems*, vol. 39, pp. 463–

489, May 2014.

[153] K. Wang and B. C. M. Fung, "Anonymizing sequential releases," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, (New York, NY, USA), pp. 414–423, ACM, 2006.

[154] J. Cao and P. Karras, "Publishing microdata with a robust privacy guarantee," *PVLDB*, vol. 5, no. 11, pp. 1388–1399, 2012.

[155] M. E. Nergiz, M. Atzori, and C. Clifton, "Hiding the presence of individuals from shared databases," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, (New York, NY, USA), pp. 665–676, ACM, 2007.

[156] C. Wressnegger, G. Schwenk, D. Arp, and K. Rieck, "A close look on n-grams in intrusion detection: Anomaly detection vs. classification," in *Proceedings of the 2013 ACM Workshop on Artificial Intelligence and Security*, AISec '13, (New York, NY, USA), pp. 67–76, ACM, 2013.

[157] M. Damashek, "Gauging similarity with n-grams: Language-independent categorization of text," *Science*, vol. 267, no. 5199, pp. 843–848, 1995.

[158] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[159] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernandez, "Syntactic n-grams as machine learning features for natural language processing," *Expert Systems with Applications*, vol. 41, no. 3, pp. 853 – 860, 2014. Methods and Applications of Artificial and Computational Intelligence.

[160] J. Y. Kim and J. Shawe-Taylor, "Fast string matching using an n-gram algorithm," *Software: Practice and Experience*, vol. 24, no. 1, pp. 79–88, 1994.

[161] N. Stakhanova, S. Basu, and J. Wong, "A taxonomy of intrusion response systems," *Int. J. Inf. Comput. Secur.*, vol. 1, pp. 169–184, Jan. 2007.

[162] C. Kruegel, E. Kirda, D. Mutz, W. Robertson, and G. Vigna, "Automating mimicry attacks using static binary analysis," in *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14*, SSYM'05, (Berkeley, CA, USA), pp. 11–11, USENIX Association, 2005.

[163] D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, (New York, NY, USA), pp. 255–264, ACM, 2002.

[164] R. V. Yampolskiy, "Mimicry attack on strategy-based behavioral biometric," in *Fifth International Conference on Information Technology: New Generations (itng 2008)*, pp. 916–921, April 2008.

[165] C. Parampalli, R. Sekar, and R. Johnson, "A practical mimicry attack against powerful system-call monitors," in *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, ASIACCS '08, (New York, NY, USA), pp. 156–167, ACM, 2008.

[166] G. Kul, D. T. A. Luong, T. Xie, V. Chandola, O. Kennedy, and S. J. Upadhyaya, "Towards effective log summarization," 2016. Unpublished. Online at: `https://odin.cse.buffalo.edu/papers/2017/EDBT-SummarizingSQL-submitted.pdf`.

[167] "27 suspended for clooney file peek," 2007. CNN Report, Online at: `http://edition.cnn.com/2007/SHOWBIZ/10/10/clooney.records/index.html?eref=ew`.

[168] J. Carr, "Breach of britney spears patient data reported, sc magazine for it security professionals," 2008. Online at: `https:`

//www.scmagazine.com/breach-of-britney-spears-patient-data-reported/article/554340/.

[169] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764 – 766, 2013.

[170] I. Pollak, "Statistics and data analysis for financial engineering (ruppert, d.; 2011) [book reviews]," *IEEE Signal Process. Mag.*, vol. 28, no. 5, pp. 146–147, 2011.

[171] V. Mavroeidis, K. Vishi, and A. Jøsang, "A framework for data-driven physical security and insider threat detection," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018.

[172] G. Kul, S. J. Upadhyaya, and A. Hughes, "Complexity of insider attacks to databases," in *Proceedings of the 2017 International Workshop on Managing Insider Security Threats, Dallas, TX, USA, October 30 - November 03, 2017*, pp. 25–32, ACM, 2017.

[173] L. Genga and N. Zannone, "Towards a systematic process-aware behavioral analysis for security," in *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications, ICETE 2018 - Volume 1: DCNET, ICE-B, OPTICS, SIGMAP and WINSYS, Porto, Portugal, July 26-28, 2018.*, pp. 626–635, 2018.

[174] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, (San Francisco, CA, USA), pp. 487–499, Morgan Kaufmann Publishers Inc., 1994.

[175] T. Uno, M. Kiyomi, and H. Arimura, "Lcm ver.3: Collaboration of array, bitmap

and prefix tree for frequent itemset mining," in *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*, OSDM '05, (New York, NY, USA), pp. 77–86, ACM, 2005.

[176] T. Uno and Others, "LCM ver.2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets," in *Proc. 1st Int'l workshop on open source data mining: frequent pattern mining implementations*, 2004.

[177] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Trans. on Knowl. and Data Eng.*, vol. 12, pp. 372–390, May 2000.

[178] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, (New York, NY, USA), pp. 1–12, ACM, 2000.

[179] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang, "H-mine: Fast and space-preserving frequent pattern mining in large databases," *IIE Transactions*, vol. 39, no. 6, pp. 593–605, 2007.

[180] S. Tsang, Y. S. Koh, and G. Dobbie, "Rp-tree: Rare pattern tree mining," in *Data Warehousing and Knowledge Discovery* (A. Cuzzocrea and U. Dayal, eds.), (Berlin, Heidelberg), pp. 277–288, Springer Berlin Heidelberg, 2011.

[181] S. Bouasker and S. Ben Yahia, "Key correlation mining by simultaneous monotone and anti-monotone constraints checking," in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, (New York, NY, USA), pp. 851–856, ACM, 2015.

[182] O. Pieczul and S. N. Foley, "Discovering emergent norms in security logs," in *2013 IEEE Conference on Communications and Network Security (CNS)*, pp. 438–445, Oct 2013.

[183] P. Fournier-Viger, J. C.-W. Lin, B. Vo, T. T. Chi, J. Zhang, and H. B. Le, "A survey of itemset mining," *WIREs Data Mining and Knowledge Discovery*, vol. 7, no. 4, p. e1207, 2017.

[184] Z. Deng and S. Lv, "Prepost$^+$: An efficient n-lists-based algorithm for mining frequent itemsets via children-parent equivalence pruning," *Expert Syst. Appl.*, vol. 42, no. 13, pp. 5424–5432, 2015.

[185] S. Moens, E. Aksehirli, and B. Goethals, "Frequent itemset mining for big data," in *2013 IEEE International Conference on Big Data*, pp. 111–118, Oct 2013.

[186] D. Apiletti, E. Baralis, T. Cerquitelli, P. Garza, F. Pulvirenti, and L. Venturini, "Frequent itemsets mining for big data: A comparative analysis," *Big Data Research*, vol. 9, pp. 67 – 83, 2017.

[187] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Advances in Database Technology — EDBT '96* (P. Apers, M. Bouzeghoub, and G. Gardarin, eds.), (Berlin, Heidelberg), pp. 1–17, Springer Berlin Heidelberg, 1996.

[188] Jian Pei, Jiawei Han, B. Mortazavi-Asl, Jianyong Wang, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu, "Mining sequential patterns by pattern-growth: the prefixspan approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 1424–1440, Nov 2004.

[189] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Advances in Database Technology — EDBT '96* (P. Apers, M. Bouzeghoub, and G. Gardarin, eds.), (Berlin, Heidelberg), pp. 1–17, Springer Berlin Heidelberg, 1996.

[190] M. J. Zaki, "Spade: an efficient algorithm for mining frequent sequences," in *Machine Learning Journal, special issue on Unsupervised Learning*, pp. 31–60, 2001.

[191] Z. Yang, Y. Wang, and M. Kitsuregawa, "LAPIN: effective sequential pattern mining algorithms by last position induction for dense databases," in *Advances in Databases: Concepts, Systems and Applications, 12th International Conference on Database Systems for Advanced Applications, DASFAA 2007, Bangkok, Thailand, April 9-12, 2007, Proceedings* (K. Ramamohanarao, P. R. Krishna, M. K. Mohania, and E. Nantajeewarawat, eds.), vol. 4443 of *Lecture Notes in Computer Science*, pp. 1020–1023, Springer, 2007.

[192] E. Salvemini, F. Fumarola, D. Malerba, and J. Han, "Fast sequence mining based on sparse id-lists," in *Foundations of Intelligent Systems* (M. Kryszkiewicz, H. Rybinski, A. Skowron, and Z. W. Raś, eds.), (Berlin, Heidelberg), pp. 316–325, Springer Berlin Heidelberg, 2011.

[193] A. Gomariz, M. Campos, R. Marin, and B. Goethals, "Clasp: An efficient algorithm for mining frequent closed sequences," in *Advances in Knowledge Discovery and Data Mining* (J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, eds.), (Berlin, Heidelberg), pp. 50–61, Springer Berlin Heidelberg, 2013.

[194] J. Wang and J. Han, "Bide: efficient mining of frequent closed sequences," in *Proceedings. 20th International Conference on Data Engineering*, pp. 79–90, April 2004.

[195] P. Fournier-Viger, C.-W. Wu, and V. S. Tseng, "Mining maximal sequential patterns without candidate maintenance," in *Advanced Data Mining and Applications* (H. Motoda, Z. Wu, L. Cao, O. Zaiane, M. Yao, and W. Wang, eds.), (Berlin, Heidelberg), pp. 169–180, Springer Berlin Heidelberg, 2013.

[196] A. Samet, T. Guyet, and B. Négrevergne, "Mining rare sequential patterns with ASP," in *Late Breaking Papers of the 27th International Conference on Inductive Logic Programming, Orléans, France, September 4-6, 2017*, pp. 51–60, 2017.

[197] A. Rahman, Y. Xu, K. Radke, and E. Foo, "Finding anomalies in scada logs using rare sequential pattern mining," in *Network and System Security* (J. Chen, V. Piuri, C. Su, and M. Yung, eds.), (Cham), pp. 499–506, Springer International Publishing, 2016.

[198] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: A survey," 2018. Online at: `https://arxiv.org/abs/1812.02282`.

[199] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. Online at: `http://archive.ics.uci.edu/ml`.

[200] F. Esteva, L. Godo, and C. Noguera, "First-order t-norm based fuzzy logics with truth-constants: Distinguished semantics and completeness properties," *Annals of Pure and Applied Logic*, vol. 161, no. 2, pp. 185 – 202, 2009. Festschrift on the occasion of Franco Montagna's 60th birthday.

[201] G. Metcalfe, N. Olivetti, and D. Gabbay, *Proof Theory for Fuzzy Logics*. Springer Publishing Company, Incorporated, 1st ed., 2008.

[202] J. Aligon, M. Golfarelli, P. Marcel, S. Rizzi, and E. Turricchia, "Similarity measures for olap sessions," *Knowledge and Information Systems*, vol. 39, pp. 463–489, May 2014.

[203] G. Kul, D. T. A. Luong, T. Xie, V. Chandola, O. Kennedy, and S. Upadhyaya, "Similarity measures for sql query clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[204] K. Stefanidis, M. Drosou, and E. Pitoura, "You may also like results in relational databases," in *Proceedings international workshop on personalized access. Profile management and context awareness Databases (PersDB 2009),in conjunction with VLDB 2009 Lyon, France*, 2009.

[205] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[206] M. B. Salem, S. Hershkop, and S. J. Stolfo, *A Survey of Insider Attack Detection Research*, pp. 69–90. Boston, MA: Springer US, 2008.

[207] I. Žliobaitė, M. Pechenizkiy, and J. Gama, *An Overview of Concept Drift Applications*, pp. 91–114. Cham: Springer International Publishing, 2016.

[208] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen, "Dynamic integration of classifiers for handling concept drift," *Inf. Fusion*, vol. 9, pp. 56–68, Jan. 2008.

[209] A. Tsymbal, "The Problem of Concept Drift: Definitions and Related Work," tech. rep., 2004. Online at: `https://www.scss.tcd.ie/publications/tech-reports/reports.04/TCD-CS-2004-15.pdf`.

[210] J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, pp. 44:1–44:37, Mar. 2014.

[211] R. Elmasri and S. Navathe, *Fundamentals of Database Systems*. USA: Addison-Wesley Publishing Company, 6th ed., 2010.

[212] F. Turkmen, S. Foley, B. O'Sullivan, W. Fitzgerald, T. Hadzic, S. Basagiannis, and M. Boubekeur, "Explanations and relaxations for policy conflicts in physical access control," in *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pp. 330–336, 2013.

[213] B. O'Callaghan, E. C. Freuder, and B. O'Sullivan, "Useful explanations," in *Principles and Practice of Constraint Programming – CP 2003* (F. Rossi, ed.), (Berlin, Heidelberg), pp. 988–988, Springer Berlin Heidelberg, 2003.

[214] A. Ferguson and B. O'Sullivan, "Relaxations and explanations for quantified constraint satisfaction problems," in *Principles and Practice of Constraint Programming - CP 2006* (F. Benhamou, ed.), (Berlin, Heidelberg), pp. 690–694, Springer Berlin Heidelberg, 2006.