

“GEORGE EMIL PALADE” UNIVERSITY OF MEDICINE,
PHARMACY, SCIENCE, AND TECHNOLOGY OF TÂRGU MUREŞ
DOCTORAL SCHOOL OF LETTERS, HUMANITIES AND APPLIED SCIENCES
SCIENTIFIC FIELD: INFORMATICS

PHD THESIS

BALANCING PRIVACY AND UTILITY IN MULTIVARIATE TIME
SERIES DATA

Adrian-Silviu ROMAN

Scientific Supervisor:

Prof. Eng. Béla GENGE, PhD

TÂRGU MUREŞ

2025



GEORGE EMIL PALADE
UNIVERSITY OF MEDICINE,
PHARMACY, SCIENCE, AND
TECHNOLOGY OF TARGU MURES

“GEORGE EMIL PALADE” UNIVERSITY OF MEDICINE,
PHARMACY, SCIENCE, AND TECHNOLOGY OF TÂRGU MUREŞ
DOCTORAL SCHOOL OF LETTERS, HUMANITIES AND APPLIED SCIENCES
SCIENTIFIC FIELD: INFORMATICS

BALANCING PRIVACY AND UTILITY IN MULTIVARIATE TIME SERIES DATA

Adrian-Silviu ROMAN

Scientific Supervisor:

Prof. Eng. Béla GENGE, PhD



TÂRGU MUREŞ
2025

“In the end, it’s all a question of balance.”

Rohinton Mistry

“Life is like riding a bicycle. To keep your balance, you must keep moving.”

Albert Einstein

Abstract

In the contemporary digital era, the continuous generation and collection of vast amounts of data from sensors embedded in devices such as biometric monitors, connected vehicles, and geolocation systems offer important opportunities for analysis and innovation. At the same time, there are significant challenges in ensuring data privacy while maintaining its utility. Achieving the privacy-utility balance becomes particularly complex in multivariate time series data, characterized by high-dimensional and correlated features. Such data requires advanced techniques for privacy preservation without compromising the data utility.

Privacy research has become increasingly necessary in light of evolving data protection regulations such as the General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), and Personal Data Protection Act (PDPA). These regulations establish standards for protecting personal data and impose the adoption of robust privacy-preserving mechanisms. The ability to balance privacy and utility ensures compliance with these laws and sustains the usability of data for various applications.

This thesis addresses the challenges of achieving an optimal balance between data privacy and utility, focusing on privacy-preserving mechanisms tailored for multivariate time series data. The research presents three novel approaches: Feature Selection for Balancing Utility and Privacy in Multivariate Time Series (FS-BUP), Data Perturbation for Balancing Utility and Privacy in Multivariate Time Series (DP-BUP), and Adaptive Privacy and Utility Preserving Real-Time Synthetic Trajectory Generation (APU-TrajGen). Each method targets specific aspects of the privacy-utility trade-off in the context of multivariate time series data, offering innovative solutions that extend beyond existing methodologies.

FS-BUP and DP-BUP address the problem data privacy in the context of Time Series Classification (TSC). FS-BUP provides a framework for optimizing feature selection for privacy protection by using a dual-classifier model comprising a Utility-Focused Classifier (UFC) and a Privacy-Breaking Classifier (PBC). This approach identifies and selects features that maximize utility while minimizing privacy risks, using a novel Feature Importance Gap (FIG) score. DP-BUP presents a perturbation mechanism that distributes noise dynamically based on feature importance, ensuring that the privacy-utility balance is maintained even in scenarios where feature reduction is not feasible. It employs the Classification Utility-Privacy Balance (BUP) score to determine optimal perturbation levels, and it ensures the effectiveness of the method against adversarial attacks. DP-BUP is designed to be implemented locally, at the device level, and to support infinite data streams.

APU-TrajGen focuses on location-based time series data. This real-time approach protects privacy while maintaining utility and adjusts parameters dynamically according to specific privacy and utility objectives. It solves the problem of balancing privacy and utility in continuous trajectory data release by using a Long Short-Term Memory (LSTM)-based model for generating synthetic trajectories that resemble real-world patterns. APU-TrajGen presents a point-to-point and trajectory-to-trajectory evaluation mechanism, ensuring that the synthetic data retains utility for applications such as traffic analysis and urban planning while protecting user privacy. Unlike traditional location data protection methods, which rely on centralized aggregators, APU-TrajGen operates locally, reducing the risk of data breaches and providing real-time adaptability. This solution is particularly valuable for autonomous driving, where real-time data transmission and privacy are critical.

The proposed methods were validated through extensive experiments on diverse datasets and adversarial scenarios. These evaluations demonstrate their effectiveness in preserving data utility while protecting privacy, even against robust adversarial models. By addressing privacy concerns in continuous data streams, these methods support applications such as autonomous driving, real-time monitoring, and health analytics.

This thesis significantly contributes to the field of privacy-preserving data processing by presenting solutions that address the complex privacy-utility trade-offs in multivariate time series data. The methodologies developed provide practical tools for data analysts, processors, and manufacturers aiming to utilize sensor-generated data while adhering to existing privacy standards.

Abstract

În era digitală, generarea și colectarea continuă a unor volume mari de date ce provin de la senzori încorporați în diverse dispozitive, precum cele biometrice, vehicule conectate și sisteme cu geolocalizare, oferă oportunități importante pentru analiză și inovație. În același timp, există provocări semnificative în asigurarea confidențialității datelor, menținând totodată utilitatea acestora. Obținerea unui echilibru optim între confidențialitate și utilitate este deosebit de complexă în cazul seriilor de timp multivariate, datorită numărului mare de atribute și a corelațiilor existente între acestea. Astfel de date necesită tehnici avansate pentru păstrarea confidențialității fără a compromite utilitatea datelor.

Cercetarea în domeniul confidențialității datelor a devenit tot mai necesară în contextul reglementărilor în continuă evoluție privind protecția datelor, cum ar fi General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), și Personal Data Protection Act (PDPA). Aceste reglementări stabilesc standarde pentru protejarea datelor cu caracter personal și impun adoptarea unor mecanisme solide de protecție a confidențialității. Echilibrul dintre confidențialitate și utilitate poate garanta respectarea acestor reglementări și contribuie la păstrarea utilității datelor pentru o gamă variată de aplicații.

Prezenta teză abordează provocările legate de obținerea unui echilibru optim între confidențialitatea și utilitatea datelor, concentrându-se pe mecanisme de protecție a confidențialității adaptate pentru seriile de timp multivariate. Cercetarea prezintă trei abordări inovatoare: Feature Selection for Balancing Utility and Privacy in Multivariate Time Series (FS-BUP), Data Perturbation for Balancing Utility and Privacy in Multivariate Time Series (DP-BUP) și Adaptive Privacy and Utility Preserving Real-Time Synthetic Trajectory Generation (APU-TrajGen). Fiecare metodă vizează aspecte specifice ale compromisului dintre confidențialitate și utilitate în contextul seriilor de timp multivariate, oferind soluții care depășesc metodologiile existente.

FS-BUP și DP-BUP abordează problema confidențialității datelor în contextul clasificării seriilor de timp, Time Series Classification (TSC). FS-BUP oferă un cadru pentru optimizarea selecției atributelor în vederea protejării confidențialității, utilizând un model cu format dintr-un clasificator orientat către utilitate, Utility-Focused Classifier (UFC), și un clasificator orientat către confidențialitate, Privacy-Breaking Classifier (PBC). Această abordare identifică și selectează trăsături care maximizează utilitatea și minimizează riscurile de confidențialitate, utilizând un nou tip de scor, Feature Importance Gap (FIG). DP-BUP prezintă un mecanism de perturbare care distribuie zgomotul pe baza importanței atributelor, asigurând menținerea

echilibrului confidențialitate-utilitate chiar și în scenarii în care reducerea numărului de atribute nu este fezabilă. Acesta utilizează scorul Classification Utility-Privacy Balance (BUP) pentru a determina nivelurile optime de perturbare și asigură eficiența metodei împotriva atacurilor de diverse tipuri. Metoda DP-BUP este concepută pentru a fi implementată local, la nivel de dispozitiv, și pentru a putea procesa fluxuri de date infinite.

APU-TrajGen se concentrează pe datele de tip serii de timp care includ date de geolocație. Abordarea propusă procesează datele în timp real și protejează confidențialitatea, menținând totodată utilitatea și ajustând parametrii în mod dinamic, în funcție de obiectivele specifice stabilite. De asemenea, rezolvă problema echilibrului confidențialitate-utilitate în publicarea continuă a datelor de localizare sub formă de traiectorii. Metoda utilizează un model bazat pe rețele neuronale de tip Long Short-Term Memory (LSTM) pentru generarea de traiectorii sintetice care reflectă tipare din lumea reală. APU-TrajGen prezintă un mecanism de evaluare punct-la-punct și traiectorie-la-traieorie, asigurând faptul că datele sintetice păstrează utilitatea pentru aplicații precum analiza traficului și planificarea urbană, protejând în același timp confidențialitatea utilizatorilor. Spre deosebire de metodele tradiționale de protecție a datelor de locație, care se bazează pe agregatori centralizați, APU-TrajGen funcționează local, la nivelul sistemelor de colectare a datelor, reducând riscul de breșe de securitate și oferind posibilitatea de adaptare în timp real. Această soluție este deosebit de valoroasă pentru scenariile ce implică conducerea autonomă, unde transmiterea datelor în timp real și confidențialitatea sunt critice.

Metodele propuse au fost validate prin numeroase experimente pe seturi de date diverse și în scenarii cu diverse tipuri de adversari. Aceste evaluări demonstrează eficiența lor în păstrarea utilității datelor, protejând în același timp confidențialitatea. Abordând problemele de confidențialitate în fluxuri continue de date, aceste metode sprijină aplicații precum conducerea autonomă, monitorizarea în timp real și analiza datelor din domeniul sănătății.

Această teză aduce o contribuție semnificativă în domeniul prelucrării datelor cu protecția confidențialității, prezentând soluții care abordează compromisurile complexe dintre confidențialitate și utilitate în cazul seriilor de timp multivariate. Metodologiile dezvoltate oferă instrumente practice pentru analiștii de date, procesatorii de date și producătorii de aplicații software, care doresc să utilizeze datele generate de senzori respectând în același timp standardele existente de protecție a datelor.

Contents

List of Publications	xi
List of Figures	xiii
List of Tables	xvii
List of Acronyms	xix
1 Introduction: The Need for Utility Preserving Data Privacy	1
1.1 Research motivation	3
1.2 Research objectives	6
1.3 Main contributions	7
1.4 Thesis structure	8
2 Related Work: Privacy and Utility Preservation in Time Series Data	11
2.1 Background	12
2.2 Utility-preserving time series data privacy techniques	16
2.2.1 Data privacy approaches	16
2.2.2 Privacy and utility metrics	30
2.3 Data privacy and utility in the context of multivariate time series classification	32
2.3.1 Time series classification	33
2.3.2 Feature selection approaches	34
2.3.3 Data perturbation approaches	35
2.4 Data privacy and utility of location data	36
2.4.1 Trajectory data privacy	36
2.4.2 Privacy-preserving synthetic trajectory generation	37
2.5 Summary	37
3 Feature Selection for Balancing Privacy and Utility in Multivariate Time Series Data	39
3.1 Problem statement and research gaps	40
3.2 The FS-BUP method	42
3.2.1 Computation of feature importance coefficients	44

3.2.2	Importance-based feature clustering	44
3.3	Experimental evaluation	47
3.3.1	Datasets and data preprocessing	48
3.3.2	Experiments, results, and discussion	49
3.4	Comparison to the state-of-the-art privacy-oriented feature selection approaches	55
3.5	Summary	58
4	Utility-Preserving Data Perturbation for Multivariate Time Series Data	
	Privacy	59
4.1	Problem statement and research gaps	61
4.2	The DP-BUP method	62
4.2.1	Feature clustering for data perturbation	64
4.2.2	Data privacy using data perturbation	65
4.2.3	Privacy and utility analysis	70
4.2.4	Parameter optimisation	71
4.3	Experimental evaluation	72
4.3.1	Data perturbation using w-event level LDP	73
4.3.2	Data perturbation using the DFT-based approach	83
4.3.3	Testing the adversarial models	90
4.3.4	Computing the optimum perturbation parameters	93
4.4	Comparison to the state-of-the-art time series protection approaches	97
4.5	Summary	100
5	Balancing Privacy and Utility in Real-Time Location Data Release	103
5.1	Problem statement and research gaps	104
5.2	The APU-TrajGen method	105
5.2.1	LSTM model-based trajectory generation	106
5.2.2	Adaptive trajectory data privacy	107
5.2.3	Privacy and utility evaluation criteria	108
5.3	Experimental evaluation	110
5.3.1	The location dataset	110
5.3.2	Experiments, results, and discussion	111
5.4	Comparison to the state-of-the-art synthetic trajectory approaches	116
5.5	Summary	117
6	Conclusions	121
6.1	Addressed research questions and objectives	121
6.2	Scientific contributions	123
6.3	Challenges and limitations of the proposed methods	125
6.4	Future research	126

List of Publications

The evaluation of the author's publications complies with the standards set by CNATDCU (National Council for the Recognition of University Degrees, Diplomas and Certificates), applicable to doctoral students enrolled after October 1, 2018. Rankings are presented based on the classification of conferences¹ and journals² in the Computer Science field.

1. **Adrian-Silviu Roman**, Roland Bolboacă, Sophie Cerf, Piroška Haller, Bogdan Robu (2025). *APU-TrajGen: Adaptive Privacy and Utility Preserving Real-Time Synthetic Trajectory Generation*. IEEE Access. [Under Review]
Journal Paper, Rank: B, 0 points, source: UEFISCDI 2023.
2. **Adrian-Silviu Roman**, Bèla Genge, Piroška Haller (2024, September). *Balancing Privacy and Utility in Multivariate Time-Series Classification*. In The 19th International Workshop on Data Privacy Management (DPM) part of The 29th European Symposium on Research in Computer Security (ESORICS).
Conference Paper, Rank: B (Workshop within an A-ranked conference), 4 points, source: CORE2023¹.
3. **Adrian-Silviu Roman**, Bèla Genge, Bolboacă Roland (2024, September). *Privacy-Oriented Feature Selection for Multivariate Time-Series Classification*. In The 28th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES), Volume 246, 2024, (pp. 500-509), <https://doi.org/10.1016/j.procs.2024.09.430>.
Conference Paper, Rank: B, 4 points, source: CORE2023¹.
4. **Adrian-Silviu Roman** (2023, October). *Dynamic Utility-Improving Time Series Data Perturbation Framework*. In The 16th International Conference Interdisciplinarity in Engineering (Inter-Eng) (pp. 850-861), Lecture Notes in Networks and Systems, vol 605. Springer, Cham, https://doi.org/10.1007/978-3-031-22375-4_69.
Conference Paper, Rank: National, 1 point, source: CORE2023¹.
5. Bèla Genge, Piroška Haller, and **Adrian-Silviu Roman** (2023, March). *E-APTDetect: Early Advanced Persistent Threat Detection in Critical Infrastructures with Dynamic Attestation*. Applied Sciences, 13, no. 6 (2023): 3409,

<https://doi.org/10.3390/app13063409>.

Journal Paper, Rank: C, 2 points, source: UEFISCDI 2023 (October)²

6. **Adrian-Silviu Roman**, and Bèla Genge, (2023, October). *A Differential Privacy-Based Approach for Protecting Crowdsourced WiFi RSS Data*. In 2023 IEEE 19th International Conference on Intelligent Computer Communication and Processing (ICCP) (pp. 255-262). IEEE, <https://doi.org/10.1109/ICCP60212.2023.10398725>.

Conference Paper, Rank: National, 1 point, source: CORE2023¹.

7. **Adrian-Silviu Roman** (2023, March). *Evaluating the Privacy and Utility of Time-Series Data Perturbation Algorithms*. Mathematics, 11(5), 1260, <https://doi.org/10.3390/math11051260>.

Journal Paper, Rank: C, 2 points, source: UEFISCDI 2023 (October)²

8. **Adrian-Silviu Roman**, Béla Genge, Adrian-Vasile Duka, and Piroška Haller (2021, December). *Privacy-Preserving Tampering Detection in Automotive Systems*. Electronics, 10, no. 24 (2021): 3161, <https://doi.org/10.3390/electronics10243161>.

Journal Paper, Rank: C, 1 point, source: UEFISCDI 2022 (October)²

Publication scores per rank:

- Rank A: 0 points
- Rank B: 8 points
- Rank C: 5 points
- Rank D: 2 points

Total publication score: 15 points.

¹CORE Conference Ranking Portal. <http://portal.core.edu.au/conf-ranks/>

²UEFISCDI. <https://uefiscdi.gov.ro/premierea-rezultatelor-cercetarii-articole>

List of Figures

1.1	Multivariate time series data protection. Data collected from sensors on several devices is protected at the device level and transmitted to a centralized data warehouse for classification and decision-making, with utility and privacy constraints.	4
1.2	The methodology utilized for achieving the privacy-utility balance in the case of multivariate time series data.	7
2.1	Batch vs. streaming data publishing in continuous data releasing. [19]	13
2.2	CDP vs. LDP protection schemes in case of continuous data releasing.	24
2.3	Example of DFT filtered speed data values using various cut-off frequencies. .	29
2.4	Overview of the most relevant state-of-the-art works, including the context, methods, data processing approaches, metrics, and identified research gaps. .	37
3.1	Problem description. Multivariate time series data collected from several sensor-based devices is transmitted to a centralized database. The selected features should decrease the accuracy of PBC, while maintaining the accuracy of the UFC.	41
3.2	The adversarial model. The attacker gets access to the necessary data for building a capable PBC.	42
3.3	Steps of the FS-BUP method.	43
3.4	Examples of feature clustering based on FIG computation for two classifications (UFC and PBC).	45
3.5	Examples of K-means-based feature clustering for two classifications (UFC and PBC). d_{max} is computed for selecting $\mathcal{F}_u$	46
3.6	Examples of K-means-based feature clustering for multiple UFCs. In this case, $\mathcal{F}^* = \mathcal{F}_{u1} \cup \mathcal{F}_{u2}$	47
3.7	The FCN-LSTM architecture [154].	49
3.8	The research gaps targeted by the FS-BUP method (highlighted in green). . .	58
4.1	Steps of the DP-BUP method.	64
4.2	Feature clustering objective.	64

4.3	Example of feature clustering based on feature importance coefficients for two classifications (UFC and PBC).	65
4.4	Example of w-event LDP perturbed speed data using various ϵ values.	67
4.5	Example of DFT perturbed speed data using various k and σ values.	69
4.6	Feature clustering based on feature importance coefficients for two classifications (UFC and PBC), conducted using Random Forest with Gini importance ($\rho_I = 0.01$).	75
4.7	The number of features in each cluster ($\rho_I = 0.01$).	76
4.8	Classification accuracy with equally distributed perturbation for all features using the w -event Local Differential Privacy (LDP) approach.	77
4.9	Driver-vs-road-type classification accuracy on the LDP perturbed data (UAH dataset).	78
4.10	Driver-vs-road-type classification accuracy on the LDP perturbed data (HCRL dataset).	78
4.11	Driver-vs-behavior classification accuracy on the LDP perturbed data (UAH dataset).	78
4.12	User-vs-activity classification accuracy on the LDP perturbed data (MotionSense dataset).	79
4.13	User-vs-activity classification accuracy on the LDP perturbed data (PAMAP2 dataset with 31 features).	79
4.14	User-vs-activity classification accuracy on the LDP perturbed data (PAMAP2 dataset with 22 features).	79
4.15	Classification accuracy with equally distributed perturbation for all features using the Discrete Fourier Transform (DFT) approach.	84
4.16	Driver-vs-road-type classification accuracy on the DFT perturbed data (UAH dataset).	85
4.17	Driver-vs-road-type classification accuracy on the DFT perturbed data (HCRL dataset).	85
4.18	Driver-vs-behavior classification accuracy on the DFT perturbed data (UAH dataset).	85
4.19	User-vs-activity classification accuracy on the DFT perturbed data (MotionSense dataset).	86
4.20	User-vs-activity classification accuracy on the DFT perturbed data (PAMAP2 dataset with 31 features).	86
4.21	User-vs-activity classification accuracy on the DFT perturbed data (PAMAP2 dataset with 22 features).	86
4.22	Parameter optimisation for the UAH dataset (<i>driver-vs-road-type</i> classification).	95
4.23	Parameter optimisation for the HCRL dataset (<i>driver-vs-road-type</i> classification).	95
4.24	Parameter optimisation for the UAH dataset (<i>driver-vs-behavior</i> classification).	95

4.25	Parameter optimisation for the Motionsense dataset (<i>user-vs-behavior</i> classification).	96
4.26	Parameter optimisation for the PAMAP2 dataset with 31 features (<i>user-vs-behavior</i> classification).	96
4.27	Parameter optimisation for the PAMAP2 dataset with 22 features (<i>user-vs-behavior</i> classification).	96
4.28	The research gaps targeted by the DP-BUP method (highlighted in green).	101
5.1	The usage of the LSTM-based model to generate protected points.	107
5.2	The APU-TrajGen trajectory generation approach.	109
5.3	Examples of trajectory protection for trajectory id = 1372859766620000467.	112
5.4	Examples of adaptive trajectory protection for trajectory id = 1372859766620000467.	114
5.5	The distribution of the distances between $\mathcal{T}_r$ and $\mathcal{T}_p$ for various targeted $[\mu_{min}, \mu_{max}]$.	115
5.6	Examples of attack results for trajectory id = 1372859766620000467.	116
5.7	The research gaps targeted by the APU-TrajGen method (highlighted in green).	118

List of Tables

2.1	Error measures for the error e_i (α and β are positive constants) [88]	30
2.2	The definitions of the most common mathematical norms [88].	31
3.3	Classification results for the selected set of features using FIG-based clustering and K-means clustering.	52
3.4	Comparison to the state-of-the-art privacy-oriented feature selection approaches	57
4.2	Classification accuracy for perturbed data using DP-BUP with w-event LDP.	80
4.3	Classification accuracy for perturbed data using DP-BUP with DFT-based perturbation ($k = 25$).	87
4.4	Classification accuracy for the advanced adversarial model.	92
4.6	Comparison to the state-of-the-art data perturbation approaches for multivariate TSC.	99

List of Acronyms

APU-TrajGen Adaptive Privacy and Utility Preserving
Real-Time Synthetic Trajectory Generation.

BA Budget Absorption.
BD Budget Distribution.
BUP Classification Utility-Privacy Balance.

CCPA California Consumer Privacy Act.
CDP Centralized Differential Privacy.
CRA Cyber Resilience Act.

DFT Discrete Fourier Transform.
DP Differential Privacy.
DP-BUP Data Perturbation for Balancing Utility and
Privacy in Multivariate Time Series.
DTW Dynamic Time Warping.
DWT Discrete Wavelet Transform.

ECU Electronic Control Units.
EDPB European Data Protection Board.
EDR Edit Distance on Real Sequences.

FFT Fast Fourier Transform.
FIG Feature Importance Gap.
FS-BUP Feature Selection for Balancing Utility and
Privacy in Multivariate Time Series.

GAN Generative Adversarial Network.
GDPR General Data Protection Regulation.
GNN Graph Neural Network.

GRU	Gated Recurrent Unit.
HAR	Human Activity Recognition.
IL	Information Loss.
IMU	Inertial Measurement Units.
IoT	Internet-of-Things.
kNN	k-Nearest Neighbors.
LBA	LDP Budget Absorption.
LBD	LDP Budget Distribution.
LBS	Location-Based Services.
LBU	LDP Budget Uniform Method.
LDP	Local Differential Privacy.
LDPS	Local Differential Privacy Streaming.
LPA	Laplace Perturbation Algorithm.
LPPM	Location Privacy Protection Mechanisms.
LSM	LDP Sampling Method.
LSTM	Long Short-Term Memory.
MAE	Mean Average Error.
MRE	Mean Relative Error.
MSE	Mean Squared Error.
NIS-2	NIS-2 Directive.
PBC	Privacy-Breaking Classifier.
PBI	Privacy Breach Increase.
PDPA	Personal Data Protection Act.
PII	Personal Identifying Information.
PL	Privacy Loss.
POI	Point of Interest.
PPCDP	Privacy-Preserving Continuous Data Publishing.
PPDP	Privacy-Preserving Data Publishing.
QI	Quasi-Identifier.

RAPPOR Randomized Aggregatable Privacy-Preserving
Ordinal Response.

RF Random Forest.

RNN Recurrent Neural Network.

RO Research Objective.

RQ Research Question.

SU Symmetrical Uncertainty.

SUP Utility-Privacy Score.

SVM Support Vector Machines.

TLDP Local Differential Privacy in the Temporal
Setting.

TLVG Time Labeled Visibility Graph.

TSC Time Series Classification.

UFC Utility-Focused Classifier.

UPA Utility Preserved Anonymization.

WCSS Within Cluster Sum of Squares.

Chapter 1

Introduction: The Need for Utility Preserving Data Privacy

In today's digital age, extensive quantities of data are collected from a wide variety of sources, including biometric sensors, connected vehicles, and geolocation devices. In this context, protecting the privacy of the sensor-generated data before it is shared or released is essential. Techniques such as anonymization, perturbation, and encryption are important tools for safeguarding this data. Equally important is the objective to ensure that the data remains accurate and usable after privacy measures are applied. Achieving the balance between privacy and utility is especially challenging in the era of Big Data [1], where integrating diverse datasets presents significant potential but also introduces new complexities, particularly for multivariate time series data.

Modern devices continuously produce large volumes of user-related and geo-tagged data. As they are more and more integrated with external systems, the resulting data is transmitted over the Internet to centralized databases for third-party analysis and decision-making (e.g., traffic management, anomaly detection, or remote diagnostics). Moreover, in certain circumstances, authorities may require [2] the transmission of sensor data for purposes such as emergency response or environmental monitoring. The high level of integration introduces privacy and security vulnerabilities, exposing devices to data breaches, cyberattacks [3, 4], and tampering [5]. In this context, safeguarding user privacy remains critical.

Sharing sensor data with external systems is not straightforward, particularly when privacy is a concern. Sensor data is typically structured as *multivariate time series* [6], representing sequences of sequences of real-world observations collected over time. The resulting datasets are often high-dimensional, with numerous correlated attributes [7]. Even when this data does not contain explicit personal identifiers, it can still reveal user-specific details or enable identification through advanced classification methods [8, 9, 10]. Thus, protecting this data from both *honest-but-curious* data processors [11] and malicious adversaries [12] is vital. These characteristics of the sensor-collected data must be considered when selecting suitable privacy-preserving methods.

The problem of data protection has also been acknowledged by policymakers and regulators. For instance, through the European Data Protection Board (EDPB), the European Commission has adopted policies for processing the personal data in the context of connected vehicles [13]. These policies classify various types of personal data collected from devices and propose suitable processing methods. Broad regulations and guidelines, such as the EU's General Data Protection Regulation (GDPR) [14], California Consumer Privacy Act (CCPA) [15], and the Personal Data Protection Act (PDPA) [16], emphasize the importance of protecting user data and highlight the necessity for effective data privacy solutions. In addition, the European Union introduced NIS-2 Directive (NIS-2) [17] and Cyber Resilience Act (CRA) [18] to strengthen cybersecurity and data protection across critical infrastructures and miscellaneous digital systems. The NIS-2 Directive requires organizations to adopt measures for managing risks and protecting their networks and data. This includes mandatory reporting of incidents and compliance with strict cybersecurity protocols, thus indirectly reinforcing data privacy objectives. Similarly, the CRA focuses on the security of digital products and services. Both highlight the importance of integrating privacy-preserving mechanisms into data processing and analysis. To comply with these evolving regulations and guidelines, collaboration between policymakers, researchers, and industry stakeholders is essential.

Data privacy refers to the safeguarding of personal and sensitive information from unauthorized access and exposure [19]. Generally, data privacy techniques are intended to allow data analysts to extract useful insights from sensitive datasets without revealing information specific to individuals [20]. Multiple mechanisms are required to ensure privacy when potentially sensitive data is transmitted over the Internet. Since many forms of data analysis do not require the actual values, the collected data can often be distorted or aggregated while providing valuable input for analysis and other types of data processing. Various techniques have been developed to protect user privacy during data sharing. Nevertheless, the generation of data in a continuous manner and the abundance of external information sources introduce additional threats and complexities to the privacy protection problem.

Implementing privacy measures on multivariate time series data is challenging due to the need to balance privacy and data utility. Protecting data while ensuring it remains valid for analysis is demanding. Many data protection methods are designed to release helpful information, distinguishing privacy from security. Security mechanisms such as encryption have the objective to obscure data completely, while privacy-preserving methods aim to release some information and hide others. At the same time, the goal is to minimize the risks of re-identification or exposure. To address privacy concerns associated with releasing time series data, various techniques have been proposed, with a significant focus on approaches involving Differential Privacy (DP) [21, 22, 23, 24]. These methods apply perturbation methods to the data, making it difficult to accurately reconstruct specific values. Despite the effectiveness of these approaches in preserving privacy, introducing noise to achieve this can significantly distort the original values, thus reducing data utility.

Ensuring *data utility* after applying a privacy mechanism is important, especially for applications such as data analysis or *data classification* [25], which are often used in decision-making processes. The utility is frequently measured using metrics based on aggregate statistical information (such as mean, variance, and distribution), which provide the necessary information for data analysts and privacy mechanism developers in order to compare the effectiveness of different privacy-preserving approaches [26]. However, in classification scenarios, using aggregated data may not be appropriate [27, 28], as classifiers need explicit time series data to capture trends and patterns in the data.

Despite the growing focus on Privacy-Preserving Data Publishing (PPDP) and Privacy-Preserving Continuous Data Publishing (PPCDP), existing research focuses on datasets that contain snapshots of data at particular moments or over a specific period, while real-world applications involve real-time, crowd-sourced data streams. These data streams require privacy-preserving measures to protect the user's data continuously during *real-time data releasing*.

Data privacy techniques can be enforced either at the source, referred to as *local data privacy* protection, or by aggregators of personal/private data, termed *global data privacy* protection. Local data privacy protection involves implementing privacy measures where the data is generated or collected. However, in many instances, local data protection may not be implemented due to additional costs involved, decreased service quality, or other technical issues (e.g., low computational resources, energy consumption, latency). In the case of global data privacy protection, the responsibility for maintaining privacy lies with aggregators of personal/private data. These entities provide technical solutions that ensure data privacy for all users, regardless of where the data originates or is accessed.

This research presents effective privacy-preserving mechanisms for protecting sensor data in the form of multivariate time series, focusing on achieving a balance between privacy and utility. The approaches are evaluated in the context of sensor data collected and processed locally with a focus on efficient implementation strategies for data privacy and utility preservation. This study aims to contribute to state-of-the-art research by developing new methods that provide privacy protection without sacrificing the value of the collected data. The research is necessary for all manufacturers interested in complying with the European Union privacy regulations or similar regulations (GDPR, CCPA, PDPA), and it aspires to have positive implications for those that rely on the utility of the protected data (e.g., data analysts, data processors).

1.1 Research motivation

While methods exist to safeguard data collected from sensors during transmission to third-party systems [10, 29, 30, 31], ensuring data privacy poses additional challenges. When applying data protection mechanisms, monitoring and retaining data usefulness is essential to ensure that enough information is preserved for effective analysis. Typically, evaluating data utility

(i.e., the usefulness of data) involves measuring how well a selected data privacy method preserves aggregate statistical information [32]. However, when classification of time series data, including location data, is concerned, relying exclusively on aggregate statistical information derived from protected data might not adequately capture the extent to which the data achieves the desired balance between privacy and utility.

In a recent review study [33] on the protection of time series data, the privacy of high dimensional data and the protection of data against malicious users are highlighted as open research questions. Additionally, the state-of-the-art literature does not cover specific scenarios for protecting time series data.

Consider the data collected continuously from sensor-based devices and sent to a central data warehouse for classification. Frequently, these sensor-based devices utilize a distributed client/server structure in which a vast number of user devices consistently generate data. Meanwhile, the backend server performs real-time analysis of the collected data streams. However, the collection of streaming data raises privacy concerns. Within this setting, consider two types of data classifiers: (i) a Utility-Focused Classifier (UFC), which identifies patterns unrelated to user identities, and (ii) a Privacy-Breaking Classifier (PBC), which aims to identify users from the data. From the data protection perspective, the balance between privacy and utility involves the implementation of a mechanism (Figure 1.1) that protects the data such that the UFC provides suitable detection accuracy, while the accuracy of the PBC is upper bounded by a designated value.

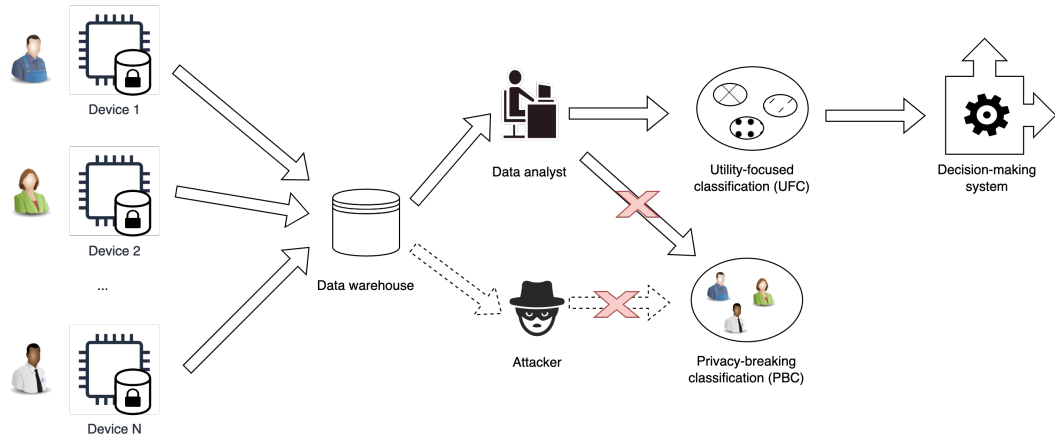


Figure 1.1 Multivariate time series data protection. Data collected from sensors on several devices is protected at the device level and transmitted to a centralized data warehouse for classification and decision-making, with utility and privacy constraints.

Example. Consider a fleet management company that collects data from sensors in their vehicles with the objective of optimizing their operations and to improve driver safety. The data includes information on vehicle speed, engine diagnostics, and fuel consumption. Further, data is transmitted to the company’s servers for analysis and decision-making. The company has two primary objectives: (i) to classify behavior into normal or aggressive driving, and (ii) to monitor vehicle health and identify potential maintenance issues. These objectives do

not involve personal information or identification. Nevertheless, the company must prevent potential threats such as unauthorized access to vehicle data or the development of PBC models capable of identifying drivers. The goal is to utilize the collected data to optimize fleet management while minimizing the risk of privacy breaches.

Balancing privacy and utility is also critical in the context of protecting location data during real-time trajectory release. Generating synthetic trajectories is useful in preserving user privacy while enabling the analysis of mobility data. Synthetic data enables location data processing without revealing actual user paths, which is necessary for applications like traffic analysis, urban planning, and mobility studies [34, 35, 36]. However, achieving an effective balance between privacy and the utility of synthetic trajectory data remains an open research challenge.

The following prerequisites regarding the data to be protected are considered in the current thesis:

- The collected data does not explicitly contain Personal Identifying Information (PII), but it contains information that can lead to user identification or re-identification.
- Data is released continuously as infinite data streams.
- The protection is performed locally, at the user level.
- Data may be multivariate, exposing multiple attributes.
- Data is not aggregated as a request to maintain data utility and the possibility of further processing, such as Time Series Classification (TSC).

Taking into account the identified research gaps – such as ensuring the privacy of multivariate data in time series classification, providing resistance against robust attacks, addressing challenges regarding scalability on low-computational devices, and the lack of effective privacy-utility balancing – along with the presented data requirements, the primary aim of this study is to explore and answer the following question: *How can multivariate time series sensor data be published/released while preserving privacy and maintaining data utility?*. The following Research Questions (RQs) were formulated to support the stated objective:

- **RQ1:** *What are the most effective methods for protecting multivariate time series data in the context of data classification involving a Utility-Focused Classifier (UFC) and a Privacy-Breaking Classifier (PBC)?*
- **RQ2:** *How can an optimal balance between data privacy and utility be defined and measured?*
- **RQ3:** *What approaches can achieve an effective privacy-utility balance across various scenarios, such as protecting multivariate time series data or ensuring location privacy?*

1.2 Research objectives

Achieving an effective balance between privacy and utility remains a fundamental challenge in privacy-preserving data releasing. The overall objective is to develop privacy-preserving data protection methods that effectively balance privacy and utility while ensuring computational efficiency for real-world applications with limited resources. By calibrating privacy parameters and selecting appropriate privacy-preserving techniques, the goal is to maintain high data utility without compromising individual privacy. Moreover, the presented methods prioritize computational efficiency, making them suitable for real-world applications with limited resources. Specific Research Objectives (ROs) have been proposed for obtaining the desired privacy-utility balance for multivariate time series data.

- **RO1:** *Formulate the problem of ensuring data privacy in the context of TSC by presenting a dual-model framework incorporating two competing classifiers: an UFC and a PBC.*
- **RO2:** *Develop an efficient feature selection method designed for multivariate time series within the context of the presented competing classifiers framework.*
- **RO3:** *Present a comprehensive time series data privacy approach that achieves a balance between privacy and utility by selecting suitable perturbation levels for real-world applications.*
- **RO4:** *Extend the concept of privacy-utility balance to location data by presenting a novel privacy-preserving real-time trajectory generation method.*
- **RO5:** *Validate the effectiveness of these methods through experiments, ensuring their correctness and practical applicability.*

Consequently, the research methodology consisted of the following steps:

- Identification of the necessary requirements for data privacy on sensor devices (e.g., automotive vehicles, wearable devices, industrial equipment) in the context of TSC.
- Evaluation of existing data protection methods suitable for time series data.
- Presentation of the methods to achieve data privacy while maintaining utility in the considered context.
- Definition of appropriate measures for evaluating the balance between privacy and utility and present methods to optimize protection parameters.
- Evaluation of the applicability of the presented strategies to location data.

To establish the initial requirements for this research, two fundamental questions were addressed, *What data needs to be hidden to preserve privacy?* and *What specific information should be retained to maintain utility?* (Figure 1.2). The answers shaped the solutions presented in this thesis.

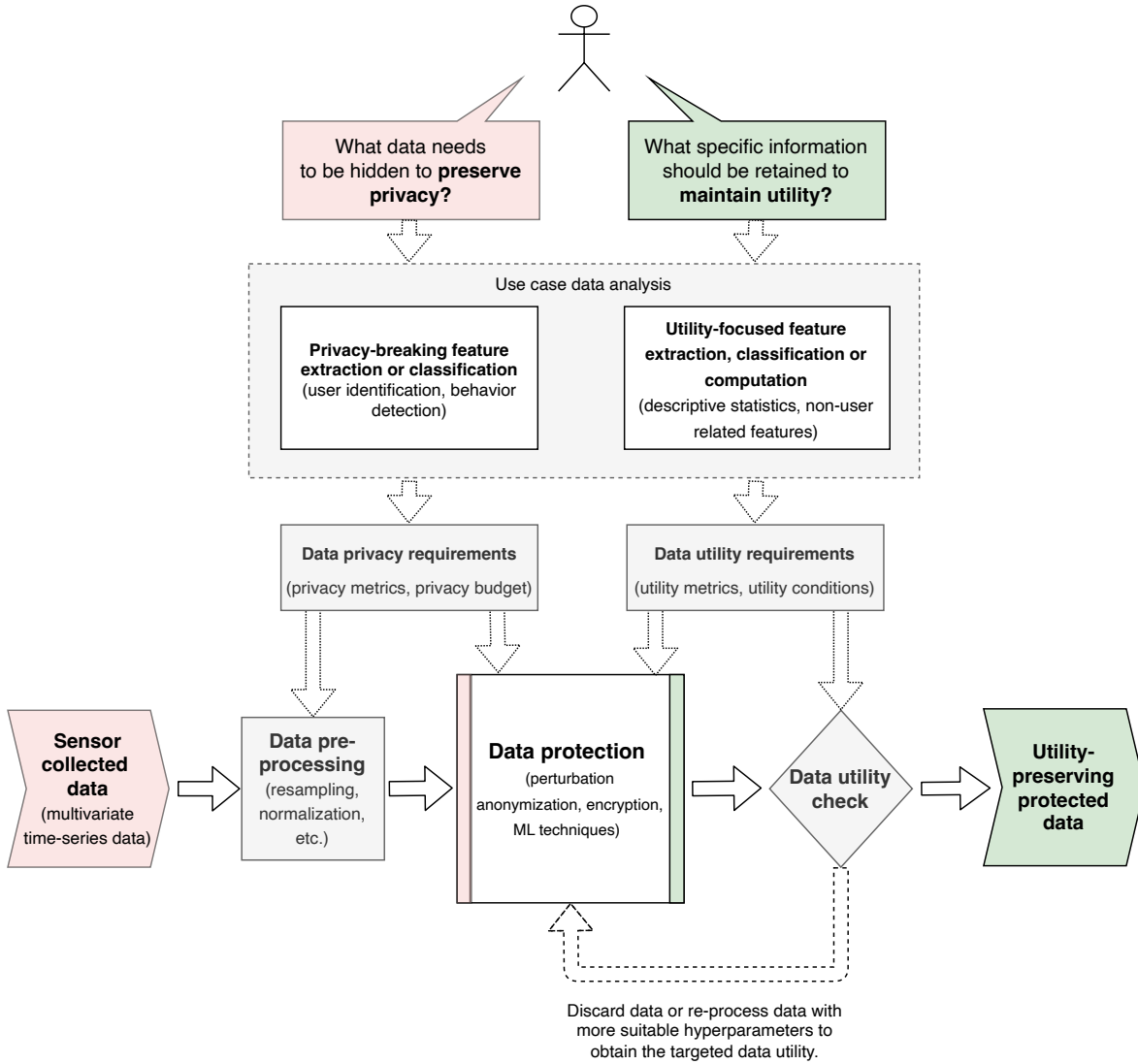


Figure 1.2 The methodology utilized for achieving the privacy-utility balance in the case of multivariate time series data.

1.3 Main contributions

This thesis presents significant advancements in privacy-preserving techniques for multivariate time series and location data, focusing on feature selection, perturbation-based mechanisms, and synthetic trajectory generation. The main contributions are summarized below:

- **Feature Selection for Balancing Privacy and Utility in Multivariate Time Series Data (Chapter 3)**
 - The development of the FS-BUP method, a novel feature selection method that optimizes the balance between privacy and utility in multivariate time series data.

- The design of the dual-classifier model with two competing classifiers: a UFC to retain data usability and a PBC to measure privacy risks.
- The definition of the Feature Importance Gap (FIG) score for ranking feature importance, enabling the selection of features that minimize privacy risks while maintaining classification accuracy.
- **Utility-Preserving Data Perturbation for Multivariate Time Series Data Privacy (Chapter 4)**
 - The development of the DP-BUP method, a perturbation-based mechanism that dynamically adjusts distributed noise on privacy-utility requirements, using the same dual-classifier model introduced in Chapter 3.
 - The definition the Classification Utility-Privacy Balance (BUP) score to quantify and fine-tune the trade-off between privacy and utility.
- **Balancing Privacy and Utility in Location Data Release (Chapter 5)**
 - The development of the APU-TrajGen, a low-computation trajectory synthesis method that generates privacy-preserving trajectories while supporting real-time point-to-point data release.
 - The design of the dynamic privacy-utility balancing mechanism for location data, capable of adjusting privacy protections at both point-to-point and trajectory-to-trajectory levels.
 - The definition the Utility-Privacy Score (SUP) to enable real-time adjustments of privacy mechanisms.

The thesis advances the field of privacy-preserving data processing by introducing novel techniques built for multivariate time series data. These contributions address key challenges such as privacy-utility balancing, resilience against adversarial inference, and device-level privacy protection, making them applicable to various systems and applications.

1.4 Thesis structure

Beyond the current introductory chapter, the thesis is organized into the following chapters:

- **Chapter 2:** This chapter presents fundamental concepts of privacy and utility, which are essential for understanding the methods introduced for publicly releasing protected time series data. The focus is on infinite time series data protected by local mechanisms. Additionally, this chapter identifies the research gaps to be addressed to advance the field. These include the need for privacy-preserving techniques that are resilient to robust adversarial attacks, the development of methods compatible with low-computational devices, and the ability to achieve an adjustable balance between privacy and utility.

- **Chapter 3:** The third chapter presents feature selection as an essential step for improving data privacy in multivariate time series data, particularly within TSC contexts. The chapter details a method that effectively balances privacy and utility by presenting Feature Selection for Balancing Utility and Privacy in Multivariate Time Series (FS-BUP), a novel privacy-oriented feature selection approach. The approach allows the owner of the data to modify the balance according to specific privacy and utility requirements. The methodology is tested on driver and Human Activity Recognition (HAR) datasets, and it demonstrates the effectiveness across various scenarios. Moreover, it also identifies cases where additional protection methods are required.
- **Chapter 4:** The fourth chapter presents Data Perturbation for Balancing Utility and Privacy in Multivariate Time Series (DP-BUP), a complete data protection mechanism through perturbation for multivariate time series data. Based on two supervised classification models – the PBC and the UFC - the mechanism measures the effectiveness of each model in compromising privacy or preserving utility. Unlike feature selection, this protection method can be utilized for cases where feature reduction is not feasible. The chapter also considers specific challenges and demonstrates the feasibility of the proposed approach.
- **Chapter 5:** This fifth chapter presents Adaptive Privacy and Utility Preserving Real-Time Synthetic Trajectory Generation (APU-TrajGen), a novel methodology for protecting location-based time series data through real-time synthetic trajectory generation. Compared to existing synthetic data methods, which lack adaptive privacy-utility balancing and point-to-point release capabilities, this method utilizes an Long Short-Term Memory (LSTM)-based model for generating synthetic data. By dynamically adjusting parameters, the model fulfills specific privacy and utility objectives. Experimental results show that the presented method maintains data utility while ensuring privacy, making it ideal for dynamic applications such as autonomous driving and real-time monitoring.
- **Chapter 6:** The last chapter presents the conclusions of the thesis, highlights the original contributions of the thesis, discusses limitations, and proposes possible directions for future research.

Chapter 2

Related Work: Privacy and Utility Preservation in Time Series Data

This chapter includes excerpts from the following articles:

- **Adrian-Silviu Roman**, Béla Genge, and Roland Bolboacă. “Privacy-Oriented Feature Selection for Multivariate Time Series Classification”. In: *Procedia Computer Science* 246 (2024). 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024), 2024. [37]
- **Adrian-Silviu Roman**, Béla Genge, and Piroska Haller. “Balancing Privacy and Utility in Multivariate Time-Series Classification”. In: *Computer Security. ESORICS 2024 International Workshops. Springer* (2025). 19th International Workshop on Data Privacy Management (DPM 2024), 2024. [25]
- **Adrian-Silviu Roman**, Béla Genge, Adrian-Vasile Duka, and Piroska Haller. “Privacy-preserving tampering detection in automotive systems”. In: *Electronics* 10.24, 2021. [38]
- **Adrian-Silviu Roman**. “Evaluating the Privacy and Utility of Time-Series Data Perturbation Algorithms”. In: *Mathematics* 11.5, p. 1260, 2023. [39]
- **Adrian-Silviu Roman**. “Dynamic Utility-Improving Time Series Data Perturbation Framework”. In: *International Conference Interdisciplinarity in Engineering. Springer*. 2022. [40]
- **Adrian-Silviu Roman** and Béla Genge. “A Differential Privacy-Based Approach for Protecting Crowdsourced WiFi RSS Data”. In: *2023 IEEE 19th International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE*. 2023. [41]

This chapter introduces the concepts necessary to explain the proposed approaches for publicly releasing protected sensor-collected data. We consider the case of infinite streamed time series data with local protection mechanisms. Detailed proofs for the theorems discussed are provided in the cited references.

2.1 Background

The objective of *data privacy* is to protect personal and sensitive information from unauthorized access. In time series data, privacy becomes critical due to the evolving, continuous nature of the data, making it highly susceptible to adversarial attacks and unintended disclosures [19]. Privacy protection ensures that data is processed in compliance with legal and ethical standards, permitting analysis while preserving other security properties such as confidentiality.

The importance of *data utility* lies in ensuring that data remains useful and accurate even after privacy-preserving measures are applied. Strict privacy methods can degrade data quality, making it ineffective for data analysis [26]. This trade-off between privacy and utility is challenging in time series data due to the strong temporal correlations between data points. Maintaining utility without compromising privacy is a fundamental concern for continuous data collection.

Continuous data release presents a unique set of challenges when the data collected is both sensitive and collected in real-time. The available approaches for data release are particularly important for sensor-collected data from Internet-of-Things (IoT) devices, whose values evolve over time and are observable at regular intervals. Depending on the observation span, such data is classified as [19]:

- *Finite data* – Data observed within a predetermined timeframe.
- *Infinite data* – Data is observed and released continuously. This increases the risk of privacy breaches, making adopting suitable data protection frameworks critical.

Both finite and infinite datasets pose significant privacy risks, especially when released continuously. The releasing methods for such datasets are categorized as [19]:

- *Batch processing* – Data is collected, processed, and released in intervals or batches. This method is typically used in offline scenarios for both finite and infinite datasets.
- *Streaming processing* – Data is processed and released in real-time. Streaming is particularly fitted for infinite datasets and is used in scenarios requiring immediate data release.

Batch data processing (Figure 2.1) and releasing commonly occur offline and can handle finite and infinite datasets. In contrast, streaming processing and releasing are inherently connected to infinite datasets, often in real-time.

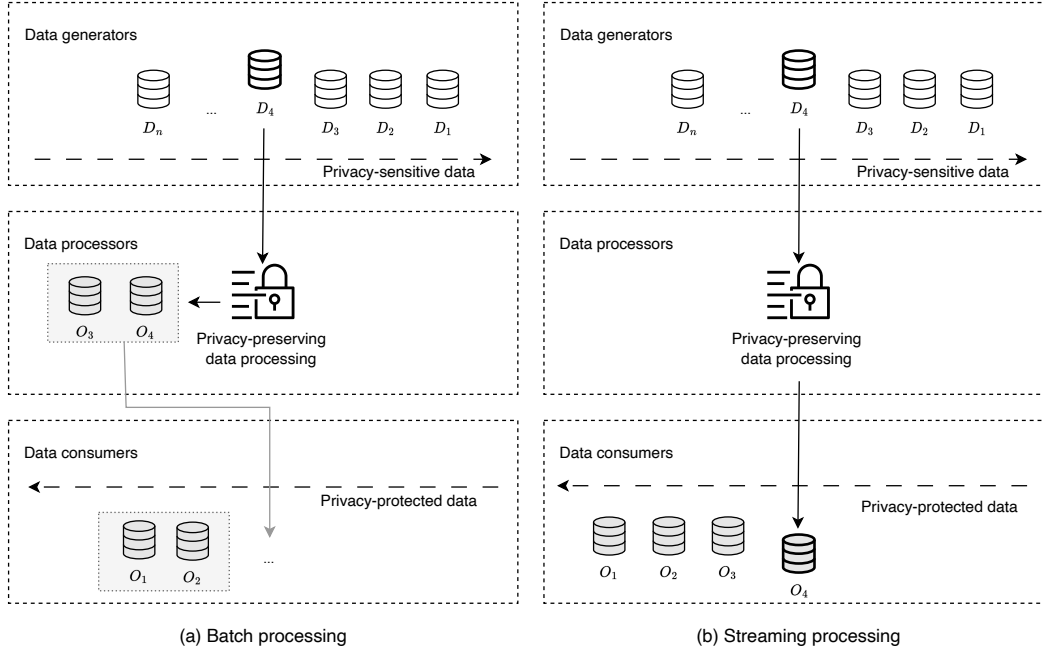


Figure 2.1 Batch vs. streaming data publishing in continuous data releasing. [19]

Two primary frameworks are commonly used for processing and releasing time series data [19]:

- *Global* – In this approach, data is collected by a centralized system that processes it and applies privacy protections before release. Although this framework offers better control over general protection, it risks privacy breaches if the central entity is compromised. However, it tends to distort data less because of the availability of larger datasets, maintaining higher utility.
- *Local* – In the local privacy framework, data is processed and protected by the data generators (e.g., users or IoT devices) before transmitting it to any third party. This framework gives individuals greater control over their data but often results in higher distortion and lower utility in the aggregated data. In this approach, the overall accuracy of the released data may be lower.

Both frameworks have their advantages and trade-offs. In the global framework, data aggregators have the ability to enforce privacy policies but may introduce significant risks if compromised. In contrast, local frameworks allow users to protect their data at the source, though this often results in lower utility of the aggregated data.

Continuous data releasing, especially for sensitive information, increases the risk of *information disclosure*, where adversaries extract sensitive details from the data. There are three primary types of information disclosure [19]:

- *Presence disclosure* – Reveals whether an individual is present in a dataset.

- *Identity disclosure* – Associates an individual with specific data points.
- *Attribute disclosure* – Exposes additional information about an individual, such as personal attributes.

Several privacy-preserving techniques [42, 43, 29, 30, 31, 44, 45, 46] have been proposed to counter information disclosure risks, each offering different levels of protection. These include:

- *Aggregation* – Combines multiple data points into a single value to hide individual contributions.
- *Generalization* – Replaces specific data values with more general categories.
- *Suppression* – Removes sensitive data entirely.
- *Perturbation* – Adds random noise to data points.
- *Encryption* – Secures data using cryptographic techniques to prevent unauthorized access.
- *Transformation* – Alters data through techniques like Fourier or wavelet transforms to reduce sensitivity.
- *Synthesis* – Generates synthetic data similar to the original without exposing sensitive information.

Each technique offers different trade-offs between privacy and utility, with some methods (such as perturbation and transformation) being more suited for real-time, continuous data processing.

Various attacks [47] can compromise sensitive information, even in systems designed for privacy protection. *Re-identification attacks* or *linkage attacks* occur when an adversary links anonymized data to external information, exposing individual users. *Inference attacks* allow attackers to predict sensitive attributes by analyzing patterns in the data, while membership inference attacks determine whether an individual's data is part of a dataset. Additionally, *differential attacks* use small changes in data queries to reveal private details. These attacks emphasize the importance of privacy protection methods, particularly in the context of continuous data collection, where the risks of exposure are higher.

Time series data consists of ordered sequences of values collected at specific time intervals. Due to their structured nature, time series datasets often capture trends, periodic patterns, and correlations between observations over time [48]. This makes them essential for predictive analysis in various domains, such as finance, healthcare, and engineering. This data may include highly sensitive personal information, particularly when linked to user activities, health metrics, or geolocation. For instance, time series generated from wearable devices or geotagged social media apps can reveal movement patterns, routines, and behavioral information. This

raises significant privacy concerns, as adversaries can exploit temporal dependencies between observations to identify individuals or infer private information even if direct identifiers are absent. Techniques such as re-identification and linkage attacks are particularly effective in this context, where the data correlations can be used to expose protected data.

Time series data is typically represented as an ordered set of values: $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$, where each data point in a time series is paired with a corresponding timestamp, forming a set of tuples (x_i, t_i) for $i = 1, \dots, n$, where x_i represents the observed value and t_i denotes the associated timestamp. When data points are collected at regular intervals, time series can be represented as a *univariate* sequence (single attribute) or a *multivariate* sequence with multiple attributes. This distinction is essential for analytical models, as multivariate time series require specialized methods to account for interdependencies between variables.

Definition 1. [49, 50] *A time series $\mathbf{X}$ is defined as an ordered sequence of values, expressed as $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$. If each element x_i is a real number, $x_i \in \mathbb{R}$, then the time series is called univariate. In contrast, if each element x_i is a vector in $\mathbb{R}^d$, the time series is called a multivariate time series, where d dimensions characterize each data point.*

Multivariate time series datasets can be structured into matrix form with n rows (*records*) and d columns (*attributes*), where each row corresponds to an individual data point. Attributes in a time series dataset can be categorized into [19]:

- *Identifiers* – Single attributes that uniquely identify an individual (e.g., card ID number).
- *Quasi-identifiers* – Attributes that, when combined, can uniquely identify an individual (e.g., age, zip code).
- *Sensitive attributes* – Confidential information linked to individuals (e.g., medical conditions, financial data, marital status).
- *Non-sensitive attributes* – Non-confidential attributes that might still contribute to re-identification risks when combined with other information.

An important characteristic of time series data is the presence of temporal dependencies. These dependencies illustrate the relationship between data values at different time points. Thus, modeling these dependencies is important for predictive tasks, as failing to account for temporal correlations can lead to misleading results. Additionally, the continuous and sequential character of time series introduces unique challenges in privacy protection, as adversaries can exploit these correlations to conduct inference attacks.

In the context of protecting multivariate time series data, particularly in the case of *infinite datasets* with *streaming processing*, it is crucial to address the increasing risks of data breaches [51]. We advocate for *local data protection* as a vital measure to preserve privacy. The introduced approaches focus on utilizing techniques such as *perturbation*, *transformation*, and *data synthesis* to protect sensitive information. Further, we provide background details

relevant to this context and present how these methods can effectively protect privacy while maintaining data utility.

2.2 Utility-preserving time series data privacy techniques

Protecting time series data poses a significant challenge because it requires finding a balance between hiding sensitive information and preserving the utility of data for various applications. This section discusses state-of-the-art approaches to data privacy that maintain utility, including differential privacy, transformation-based perturbation, and feature selection. It also introduces key privacy and utility metrics for evaluating the effectiveness of the resulting data.

2.2.1 Data privacy approaches

Time series data collected from various sources, such as automotive or wearable sensors, are commonly treated as time-domain signals. Their characteristics—including amplitude, average, trends, peaks and troughs, and periodicity—can reveal user-specific patterns, making them sensitive and requiring privacy protection. Furthermore, effective protection mechanisms must account for both the time and frequency domains to ensure comprehensive data security.

Various methodologies and techniques [24] have been developed to safeguard the privacy of sensitive sensor data streams when transmitted over the Internet for processing and release by third-party systems. The primary approaches for protecting private information in time series sensor data include *perturbation*, *encryption*, and *anonymization* [52].

Encryption is fundamental privacy-preserving technique designed to protect data by converting it into a format that can only be deciphered with the appropriate decryption key. While throughout the scientific literature we find many types of cryptographic operations (e.g., homomorphic encryption [45]) that provide privacy protection, they often have important limitations. Implementing encryption in sensor-based systems can be challenging due to processing power constraints. Moreover, encryption methods inherently rely on the assumption that the entities handling personal information are trustworthy [53]. In contrast, the focus of this research is to explore approaches that offer a satisfactory balance between ensuring the privacy of individuals and preserving the utility of the data, making the solutions more applicable to real-world scenarios.

Anonymization (also called sanitation [54] or de-identification) involves the removal, replacement, or hashing of personally identifiable information (PII) within a dataset. This process preserves data quality while enabling secure transmission over the Internet. Common anonymization techniques include *k-anonymity* [42], *l-diversity* [55], and *t-closeness* [44]. However, enforcing anonymization in time series data remains challenging, particularly when explicit personal identifiers are absent.

Perturbation, also known as *random perturbation*, is a popular technique for safeguarding private time series data. It hides sensitive information while preserving the potential for meaningful data analysis through data mining techniques. Perturbation does not require prior

knowledge of the data being protected, making it adaptable to various contexts. Additionally, it is computationally efficient, making it particularly suitable for resource-constrained environments [54]. Data perturbation techniques include *randomization-based methods* (additive perturbation [46], geometric perturbation [56], multiplicative perturbation [56, 57], nonlinear perturbation [58], differential privacy (DP) [59]) and *transformation-based methods* [43, 29, 30, 31, 10]. In transformation-based methods, collected values are first mapped from their original feature space to a lower-dimensional representation, where noise is then introduced to enhance privacy protection. Whether the perturbation preserves the essential characteristics of the data depends on (i) how much perturbation is added (i.e., ϵ in differential privacy) and (ii) how the perturbation is added. Generally, the perturbation amount is pre-defined by the user.

Differential privacy (DP) (also named ϵ -differential privacy or ϵ -DP) [60] is a widely recognized technique for implementing perturbation-based privacy protection. Traditional approaches, such as anonymization or aggregation, have been susceptible to various attacks, and even advanced techniques like k -anonymity [42]) have failed to protect data from certain types of attacks. Differential privacy mitigates these risks by introducing carefully calibrated noise to the original data using various mathematical algorithms (e.g., Laplace, Gaussian). Initially designed for securing database queries, Differential Privacy (DP) ensures that modifying a single record has a minimal impact on the overall statistical output [61]. However, achieving an optimal balance between privacy and accuracy remains particularly challenging for time series data, as the introduction of noise can significantly degrade data utility [52].

Another practical approach for data anonymization is *data transformation*: the collected values are firstly translated from the original feature space into a lower-dimensional feature space. Furthermore, high energy coefficients are considered [43], or noise is added to the resulting data matrix. The processed data is then transformed back to the original space. Examples of such transformations that preserve Euclidean distances are the Discrete Fourier Transform (DFT) and the Discrete Wavelet Transform (DWT) [62]. Various algorithms that implement data transformations have been described in the literature, such as the Fourier perturbation algorithm (FPA) [30], clamping Fourier perturbation algorithm (CFPA) [10], wavelet perturbation algorithm (WPA) [31], and clamping wavelet perturbation algorithm (CWPA) [10]. These approaches are helpful as they preserve Euclidean distances [43, 29], make reconstructing original values difficult, and provide significant data reduction. The drawback of time series data transformation is related to the usefulness of the perturbed data, as a trade-off between privacy and utility needs to be established.

2.2.1.1 Data perturbation

Time series perturbation is a privacy-preserving technique that safeguards sensitive data by introducing noise into the original values. This process obscures the true data, making it challenging for adversaries to reconstruct or infer the original information while preserving the overall statistical properties necessary for analysis. One of the fundamental principles

[29] in time series perturbation is that *each time point must be perturbed independently*. This approach ensures that a data leak at a specific timestamp does not reveal the pattern or nature of the perturbation across the entire series. If adjacent data points are perturbed in a correlated manner, the perturbation pattern can be inferred by analyzing these temporal dependencies, and, thus, compromising the privacy of the series.

When a time series shows specific trends, the independent perturbation may not be sufficient to protect against advanced filtering techniques. This is due to the fact that filtering methods can exploit the inherent patterns within the data to separate the added noise from the actual values. For example, if a time series displays regular periodic patterns, even independently perturbed values may be reconstructed using smoothing techniques or signal decomposition. To solve this issue, perturbation techniques must be designed to be *indistinguishable from the original data*. This ensures that the perturbed data follows the same distribution and behavior as the actual data.

For a perturbation to be effective, it must meet the criterion of *indistinguishability* [29]. It should be difficult for an adversary to distinguish between the perturbed series and a series with similar characteristics to the original. This concept is essential for time series data, as traditional perturbation methods may still allow for detecting the actual values if temporal dependencies are strong. Indistinguishability ensures that even if some data points are known, the noise pattern cannot be used to predict or infer the values at other timestamps.

Let $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ represent the original time series, where x_i is the observed value at time t_i . The perturbed time series, denoted as $\mathbf{X}' = \{x'_1, x'_2, \dots, x'_n\}$, is generated by adding noise η_i to each time point independently, such that:

$$x'_i = x_i + \eta_i. \quad (2.1)$$

The noise η_i is typically extracted from a zero-mean distribution (e.g., Gaussian or Laplace), calibrated to balance privacy and utility. The challenge lies in selecting a noise distribution that aligns with the patterns and distribution of $\mathbf{X}$ so that $\mathbf{X}'$ retains the same statistical properties as $\mathbf{X}$.

Compared to other privacy-preserving techniques, such as k-anonymity or generalization, *perturbation offers a higher degree of flexibility* in controlling privacy levels through adjustable noise parameters. Perturbation, especially when combined with ϵ -DP, provides a robust way to obscure data without completely masking significant trends, allowing for more granular control over the trade-off between privacy and utility.

While several perturbation algorithms have been developed, DP has emerged as a standard in privacy research due to its strong guarantees. DP introduces calibrated noise to the dataset and it guarantees that the inclusion or exclusion of any individual data point has a minimal impact on the output of a query. When applied to time series data, DP can be implemented either at individual timestamps or over larger data windows, providing varying levels of privacy depending on the application. The advantage of using DP in time series perturbation is that

it not only obscures individual data points but also mitigates the risk of cumulative leakage over time, a common problem in continuous data releasing.

Several techniques integrate data perturbation with data transformation. In these methods, the collected time series data is mapped from the high-dimensional space into a lower-dimensional feature space. Noise is introduced to the resulted features, thereby preserving data privacy while minimizing information loss. In this study, in addition to utilizing a DP-based approach, we also experimented with a perturbation method based on the DFT, which obfuscates data by modifying its frequency components before applying noise.

2.2.1.2 Differential privacy

The state-of-the-art research proposes various perturbation algorithms capable of protecting sensitive data as presented in prior work [39]. Nevertheless, DP has gained considerable recognition as a prominent technique. Differential privacy (DP) or ϵ -differential privacy [60] is a widely recognized framework for perturbation-based privacy protection. It has been extensively applied across various domains, including deep learning-based approaches [63], to ensure data privacy while maintaining analytical utility.

The initial DP approach proposed by Dwork et al. [60] assumes that a trustworthy entity collects potentially sensitive information from multiple participants collected in a database D . The objective is to share statistical analyses computed from D without compromising participant privacy. This is accomplished by introducing random perturbation to the query results, ensuring that (i) the existence of any single participant in D remains private and (ii) the utility of the released statistics is maintained.

Consider $\mathcal{D}$ as a set of databases, each represented as a series of records. D and D' in $\mathcal{D}$ are considered *neighboring* if D' can be obtained from D by either adding or removing a single record. Let $\mathcal{M}$ be a perturbation mechanism that takes a database D from $\mathcal{D}$ as input and generates an output O in the domain $\mathcal{O}$. Then, the principle of ϵ -differential privacy can be formulated as follows.

Definition 2. (ϵ -differential privacy [60]) A perturbation mechanism, $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{O}$, satisfies ϵ -differential privacy if for any output $O \subseteq \mathcal{O}$ and any pair of neighboring databases D and D' , the following condition holds for $\epsilon \geq 0$:

$$Pr[\mathcal{M}(D) \in O] \leq e^\epsilon \cdot Pr[\mathcal{M}(D') \in O]. \quad (2.2)$$

The non-negative real number ϵ is called *privacy budget* and it controls the magnitude of the perturbation. Definition 2 states that the probability of $\mathcal{M}$ generating a specific output O should be closely the same whether or not a particular data record is included in D . Consequently, an adversary would be unable to discern the presence or absence of a specific participant in D from a participant in D' . The smaller the ϵ , the higher the privacy.

The perturbation added by a mechanism $\mathcal{M}$ that satisfies the ϵ -differential privacy is commonly calculated using the Laplace Perturbation Algorithm (LPA) [60, 64]. Other

approaches include the exponential mechanism [65] and the randomized response [66]. LPA operates by adding noise that follows a Laplace distribution to the output of a query or statistic derived from a database D . The Laplace distribution is characterized by its mean (usually set to zero) and scale parameter, which is directly related to the *sensitivity* of the query and the desired level of ϵ (privacy budget).

In the context of differential privacy, *sensitivity* refers to the maximum amount that the output of a query or function can change due to adding or removing a single individual's data. It measures how much privacy can be compromised by the inclusion or exclusion of a single record.

Definition 3. (*Sensitivity [60]*) Given a function $f : D \rightarrow \mathbb{R}^d$, the sensitivity, Δf , is defined as:

$$\Delta f = \max_{D, D'} (\| f(D) - f(D') \|), \quad (2.3)$$

for all D, D' that differ on one record, where d is the number of values in the output, and $\| \cdot \|$ denotes the norm used to measure the difference between the function outputs $f(D)$ and $f(D')$.

The scale parameter of the Laplace distribution is set in such a way that it compensates for the maximum change any single individual's data can have on the output, scaled by the privacy budget ϵ . LPA satisfies ϵ -differential privacy as stated by the following theorem.

Theorem 1. (*Laplace Mechanism [60]*). Let f be a function, $f : D \rightarrow \mathbb{R}^d$, where D is the domain and $\mathbb{R}$ is the range of f . The mechanism $\mathcal{M}$ satisfies the ϵ -differential privacy if

$$\mathcal{M}(D) = f(D) + \text{Lap}(\frac{\Delta f}{\epsilon})^d, \quad (2.4)$$

where Δf is the sensitivity of the function f , and $\text{Lap}^d(\Delta f/\epsilon)$ draws randomly d values from Laplace distribution with center 0 and scale $\Delta f/\epsilon$.

The LPA is the general method for introducing noise to data to achieve differential privacy. Nonetheless, in specific contexts, the magnitude of noise introduced by the LPA can be excessively high, significantly diminishing the data's utility.

DP exhibits the following essential properties: composability and post-processing. DP composability [67, 68] refers to maintaining privacy guarantees when combining multiple queries or computations on a dataset. It ensures that the overall privacy protection remains intact even as additional queries are made or computations are performed. Composability is crucial in scenarios where multiple analyses or applications must access the same dataset while preserving privacy. There are two main aspects to consider.

Sequential composition ensures that differential privacy remains preserved when multiple computations are made sequentially on the same dataset. In other words, the privacy guarantee holds even if an adversary observes the responses to all computations. Sequential composition typically involves adjusting the privacy parameters of individual computation to account for

the cumulative privacy loss over multiple computations. Sequential composition is expressed as follows:

$$\epsilon_{\text{total}} = \sum_{i=1}^n \epsilon_i, \quad (2.5)$$

where ϵ_{total} is the total privacy parameter for the sequence of n computations, and ϵ_i represents the privacy parameter for the i^{th} computation.

Parallel composition addresses scenarios where multiple independent analyses or computations are performed simultaneously on disjoint subsets of the dataset. It guarantees that the overall privacy protection is at least as strong as the individual protections provided for each analysis. Parallel composition is expressed as follows:

$$\epsilon_{\text{total}} = \max_{i=1}^n \epsilon_i, \quad (2.6)$$

where ϵ_{total} is the total privacy parameter for the parallel computations, and ϵ_i represents the privacy parameter for the i^{th} computation.

By ensuring both sequential and parallel composition, DP offers strong guarantees of privacy preservation in various scenarios, including data analysis, machine learning, and statistical inference. This property allows the deployment of privacy-preserving techniques confidently in complex data environments while safeguarding sensitive information.

Additionally, the *post-processing* property ensures that after the data has been protected with a differentially private mechanism, any further computation on the output does not compromise the privacy guarantee. In other words, applying further transformations or analyses to the output of a differentially private computation does not weaken the overall privacy protection provided by the original mechanism. This property is significant as it allows for flexibility in data analysis pipelines without sacrificing privacy.

The post-processing property can be formally described as follows:

Theorem 2. (*Post-processing [20]*) *If $\mathcal{M}(D)$ satisfies ϵ -differential privacy, then for any function g , $g(\mathcal{M}(D))$ satisfies ϵ -differential privacy.*

By ensuring the post-processing property alongside sequential and parallel composition, differential privacy offers robust guarantees of privacy preservation across various data analysis scenarios. This allows the deployment of privacy-preserving techniques in diverse settings while maintaining individuals' confidentiality and trust.

Differential privacy for time series data

DP is generally applied to time series data or a stream of events for computing statistics that are released periodically. In this case, commonly used approaches involve *event-level perturbation* [21], where each event or value is perturbed individually, and *user-level perturbation* [21], where the privacy budget is allocated on a per-user basis and hides all events of any user throughout

the entire stream. In the case of continuously collected time series data, sensitive information is distributed over multiple events that should be protected. *Event-level* DP often does not provide adequate privacy protection, whereas *user-level* DP is typically feasible only within the bounds of a finite data stream. The *user-level privacy* is unsuitable for infinite streams as the perturbation should be applied to all values to achieve the privacy goal. Consequently, the perturbation on each value should be substantial, which drastically decreases data utility in the long run. The *event-level privacy* aims to protect single events, and it does not protect multiple events.

Time series data, continuously collected from sensors, is perceived as infinite data streams, with each distinct value representing an event generated by the user (such as an event indicating “the vehicle v is traveling at speed s at time t ”).

The *w-event-level* perturbation [22] provides privacy protection to any sequence of w consecutive events and represents a compromise between the *user* and the *event-level privacy*. The *event-level* privacy, which hides individual events in streams, fails to protect user privacy adequately. Conversely, the *user-level* privacy method, aimed to hide a user’s presence across infinite streams, proves unrealistic for many practical situations [69].

The *w-event-level* approach distributes the privacy budget into sub-mechanisms (one submechanism for each timestamp), and it uses the constraint that the total privacy budget consumed in a window is less than or equal to ϵ .

Dwork [21] introduced the idea of neighboring among stream prefixes, similar to the notion of neighboring databases, thereby enabling the application of DP to streaming data. Consider a stream denoted by S , representing a sequential series of values, with each individual value $S[i]$ representing an event. Over time, new events are continuously added to this stream. When the stream is infinite, let S_t be the prefix of S after adding t events, $S_t = (S[1], S[2], \dots, S[t])$. Two streams S_t, S'_t are considered *w-neighboring* if their constituent elements are pairwise identical or neighboring, while all their neighboring elements are in a window of up to w values [22].

Definition 4. (*w-event privacy* [22]) Let $\mathcal{M}$ be a perturbation mechanism that processes an input stream S_t . Define $\mathcal{O}$ as the set of all possible outputs of $\mathcal{M}$. The mechanism $\mathcal{M}$ satisfies *w-event differential privacy* (or *w-event privacy*) if, for all possible outputs $O \subseteq \mathcal{O}$, all *w-neighboring* stream prefixes S_t and S'_t , and for any time t , the following condition holds:

$$Pr[\mathcal{M}(S_t) \in O] \leq e^\epsilon \cdot Pr[\mathcal{M}(S'_t) \in O]. \quad (2.7)$$

This definition shows that a *w-event* privacy mechanism provides an ϵ -differential privacy for any sliding window of size w . It considers scenarios in which sensitive data is revealed through a sequence of events spanning a fixed duration, w , and it refers to situations where all events happen one after another within a span of w consecutive timestamps.

For instance, in the case of sensor data collected by a vehicle, the privacy goal may require concealing the speeding pattern of the vehicle in a specific time interval (for example, one

minute), as such patterns may lead to driver identification. Therefore, the w -event (DP) ensures the obfuscation of all speed data occurring within a sequence of w consecutive timestamps.

For $w = 1$, the w -event privacy is equivalent to the *event-level privacy*, while setting w to infinity, the w -event privacy converges to the *user-event privacy*.

The mechanism $\mathcal{M}$ is regarded as a composition of t mechanisms $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_t$. Thus, each mechanism $\mathcal{M}_i$ perturbs $S_t[i]$, introducing an independent randomness.

Theorem 3. [22] *Let $\mathcal{M}$ be a mechanism that processes a prefix stream S_t as input, where each element $S_t[i] = v_i$, and produces an output transcript $(o_1, o_2, \dots, o_t) \in \mathcal{O}$. Suppose $\mathcal{M}$ is decomposed into t individual mechanisms $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_t$, such that each mechanism $\mathcal{M}_i$ independently processes v_i and generates an output o_i , ensuring ϵ_i -differential privacy. Then, $\mathcal{M}$ satisfies w -event privacy if, for any user and any timestamp $i \in [t]$, the cumulative privacy budget within a sliding window of size w adheres to the following condition:*

$$\sum_{k=i-w+1}^i \epsilon_k \leq \epsilon. \quad (2.8)$$

Theorem 3, derived from the composition property of ϵ -differential privacy, establishes that the total privacy budget ϵ applies within a sliding window of size w . Thus, the allocation of the privacy budget across the timestamps contained within the window is possible.

Local differential privacy

Differential privacy was initially introduced within a centralized framework, referred to as Centralized Differential Privacy (CDP) [60]. Under the CDP model, a trusted aggregator collects raw data from multiple sources and applies calibrated noise to the aggregated results before releasing them to data analysts. This approach requires the assumption of a trusted entity to ensure that sensitive information is adequately protected. In scenarios where such a trusted central party is unavailable or infeasible, LDP is an alternative.

Local Differential Privacy (LDP) was developed to eliminate the need for a trusted data aggregator by enabling each user to independently perturb their data before sharing it with any central entity [70] (Figure 2.2). LDP provides privacy protection for numerous end users at a local level. This approach has been effectively implemented by several leading companies, including Google [71], Microsoft [72], Apple [73], and Uber [74]. It ensures that even though data from multiple individuals is collected, the aggregator cannot accurately infer the actual values of individual inputs. Despite being a variant of DP, LDP inherits essential properties from Centralized Differential Privacy (CDP), such as sequential and parallel composition, as well as post-processing [65]. However, applying LDP to time series data introduces several challenges due to the sequential nature of time series, the high-dimensional structure, and the need for real-time privacy protection.

The application of LDP to time series data involves several challenges [75] that are distinct from those encountered in CDP-based approaches:

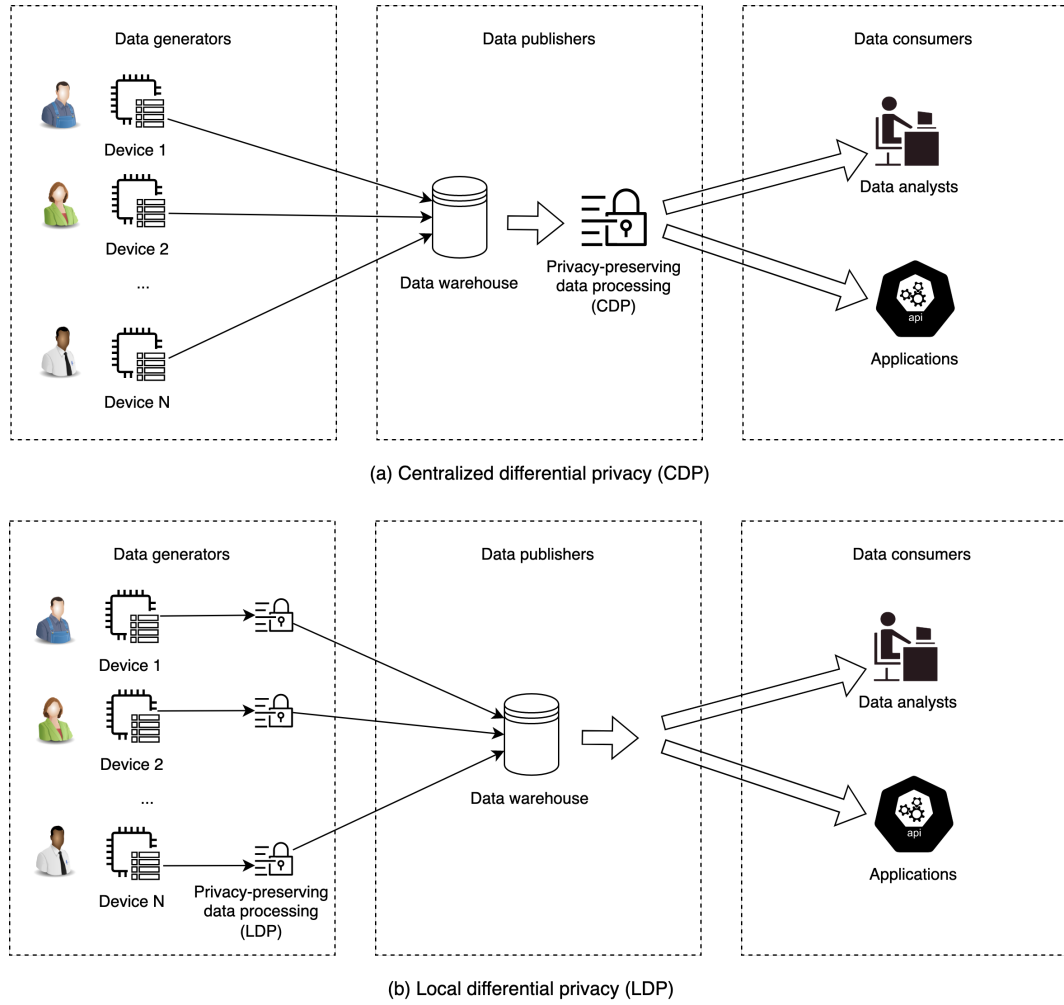


Figure 2.2 CDP vs. LDP protection schemes in case of continuous data releasing.

- *Limited access to raw data* – In CDP, privacy-preserving mechanisms can leverage global knowledge of raw data to strategically inject noise at key timestamps or adapt the privacy budget allocation based on data sparsity. However, in LDP, since raw data is not accessible to the central aggregator, individual users must add noise locally before the data is shared. This lack of direct access complicates the implementation of efficient noise reduction techniques commonly used in CDP.
- *Diminished utility due to budget allocation* – Budget allocation strategies that are effective in CDP contexts, such as varying privacy budgets across timestamps, lead to a reduction in utility for CDP. In LDP settings, applying similar budget allocation techniques may result in an exponential decrease in utility, especially when dealing with high-dimensional and multivariate time series data. Thus, optimizing the privacy-utility trade-off more carefully in LDP is necessary.

- *Increased communication costs* – Since users must continuously submit their perturbed data to the central aggregator in a streaming context, LDP can result in significant communication overhead. This is particularly problematic for resource-constrained devices like IoT sensors, which may struggle to handle the increased data transmission demands.

To better adapt to the unique requirements of time series data, various approaches to the standard LDP model [70] have been proposed. One such modification is the introduction of the w -event LDP framework, which protects sequences of events within a predefined window w . This approach provides a balance between protecting individual events and preserving data utility over longer observation windows. Furthermore, being a variant of differential privacy (DP), LDP inherits the key properties of CDP, including the principles of sequential/parallel composition and post-processing theorems [65].

Although event-level LDP is adaptable to infinite data streams, it falls short in safeguarding a user's data at a coarse-grained level [75]. In the case of local settings, Ren et al. [75] proposed a slightly modified definition for neighboring streams.

Definition 5. (w -neighbouring [75]) Consider two stream prefixes, S_t and S'_t , and let w be a positive integer. S_t and S'_t are w -neighbouring in the local setting if for each $S_t[i]$, $S_t[j]$, $S'_t[i]$, $S'_t[j]$ with $i \leq j$, $S_t[i] \neq S'_t[i]$, and $S_t[j] \neq S'_t[j]$, then $j - i + 1 \leq w$.

This definition means that two streams are considered w -neighboring if their differing elements span a window of no more than w timestamps. This interpretation differs slightly from the definition of neighboring in the centralized setting [60], reflecting the localized nature of privacy preservation in LDP.

The notion of w -event privacy is formalized by the following definition:

Definition 6. (w -event LDP [75]) Let $\mathcal{M}_{LDP}$ be a local perturbation mechanism, which takes as input a stream S_t . Also, let $\mathcal{O}$ be the set of all possible outputs of $\mathcal{M}_{LDP}$. Then $\mathcal{M}_{LDP}$ satisfies the w -event ϵ -Local Differential Privacy (LDP) (or, w -event LDP) if for all outputs $O \subseteq \mathcal{O}$, all w -neighboring stream prefixes S_t, S'_t , and all t , the following condition is satisfied:

$$\Pr[\mathcal{M}_{LDP}(S_t) \in O] \leq e^\epsilon \cdot \Pr[\mathcal{M}_{LDP}(S'_t) \in O]. \quad (2.9)$$

Further, theorem 3, based on the composition property of ϵ -differential privacy, can be adapted for LDP.

Theorem 4. [75]. Let $\mathcal{M}_{LDP}$ be a mechanism that processes a prefix stream S_t as input, where each element $S_t[i] = v_i$, and produces an output transcript $(o_1, o_2, \dots, o_t) \in \mathcal{O}$. Suppose that $\mathcal{M}_{LDP}$ is decomposed into t independent mechanisms $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_t$, where each mechanism $\mathcal{M}_i$ independently processes v_i and generates an output o_i , ensuring ϵ_i -Local Differential Privacy (LDP). Then, $\mathcal{M}_{LDP}$ satisfies w -event LDP if, for any user and any timestamp

$i \in [t]$, the cumulative privacy budget within a sliding window of size w satisfies the following condition:

$$\sum_{k=i-w+1}^i \epsilon_k \leq \epsilon. \quad (2.10)$$

Theorem 4 enables the usage of w -event privacy in local settings and the distribution of the ϵ privacy budget among the timestamps contained in the window w . For brevity, from now on, we will denote $\mathcal{M}_{LDP}$ as $\mathcal{M}$.

Privacy budget division

The approaches to determining the size of the privacy budget ϵ range from experimenting with various values to building a threat model and working backward to computing the necessary constraints for protection. However, many differential privacy practitioners admit that the choice of ϵ is entirely arbitrary [76], following the general rule that “smaller ϵ means better privacy”.

In the context of centralized CDP, Kellaris and Papadopoulos [77] propose two data-dependent budget allocation schemes, Budget Distribution (BD) and Budget Absorption (BA), tailored explicitly for privacy-preserving data releasing. These schemes consider the impact of previous data releases on the utility of the current release, making them highly sensitive to the properties of the original dataset. Kellaris and Papadopoulos [77] recommend conducting thorough experimental analyses on actual datasets to assess the effectiveness of these approaches.

Wang et al. [78] introduced a dynamic allocation strategy that adjusts the privacy budget based on the similarity of data streams. This approach improves overall utility by distributing more budget to highly variable data segments and less to stable segments. Later, Fioretto and Van Hentenryck [79] enhanced this by selecting representative data points for noise injection and using interpolation techniques to reconstruct the remaining points, thereby reducing the overall noise added to the dataset.

In the LDP context, Ren et al. [75] mentions the following privacy budget division techniques:

- LDP Budget Uniform Method (LBU) – Distributes the total privacy budget ϵ uniformly across all w timestamps within the sliding window. Each timestamp receives a budget of ϵ/w , which results in high noise levels for large w -values, compromising data utility.
- LDP Sampling Method (LSM) – Allocates the entire budget ϵ to a single timestamp within the window and approximates the values for the remaining $w - 1$ timestamps. This technique is effective for streams with minimal variation but can accumulate significant estimation errors if temporal changes are frequent.
- LDP Budget Distribution (LBD) – Follows an exponentially decreasing allocation pattern for publication timestamps. Half of the budget ($\epsilon/2$) is used for dissimilarity checks and

distributed evenly across timestamps, while the remaining half is reserved for releasing purposes.

- LDP Budget Absorption (LBA) – Initially allocates $\epsilon/2$ across all timestamps and reserves the other half for the timestamps selected for publication. This strategy balances evaluating the entire data stream with focusing on key timestamps.

The size of the noise is determined by the privacy budget ϵ and the *sensitivity* [60]. Introducing noise into the data by a mechanism $\mathcal{M}$ that satisfies ϵ -differential privacy is typically achieved using the Laplace mechanism as proposed by Dwork et al. [60].

The effectiveness of differential privacy mechanisms depends on both the privacy budget ϵ and the sensitivity of the data. *Sensitivity*, defined as the maximum change in the computation output when a single record is altered, determines the scale of noise required to achieve ϵ -differential privacy. The Laplace mechanism, proposed by Dwork et al. [60], is commonly employed to inject noise proportional to the sensitivity, ensuring robust privacy guarantees while preserving the overall data structure. The mechanism is mathematically defined as:

$$\mathcal{M}(x) = f(x) + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right), \quad (2.11)$$

where $\text{Lap}(\frac{\Delta f}{\epsilon})$ is the Laplace noise added to obscure the actual value, and Δf is the sensitivity of the function f .

Choosing an appropriate privacy budget is a complex task that requires balancing privacy and utility. While smaller values of ϵ provide more robust privacy, they can significantly reduce the utility of the data. Conversely, larger values offer better utility but may expose sensitive information. Thus, selecting ϵ often involves domain-specific considerations, empirical evaluation, and sometimes compromises based on the expected threat model.

2.2.1.3 Discrete Fourier Transform-based distortion

The Fourier transform is reversible, which describes a function as a sum of sine and cosine waves. By exploiting the fundamental properties of the Fourier transform, sensor data is decomposed into frequencies and reconstructed after a prior filtering phase. Fast Fourier Transform (FFT) is an algorithm for computing the DFT. It converts a data sequence from its original domain (time in our case) to the frequency domain. Several approaches to achieve time series protection using DFT have been proposed [30, 10, 38].

Consider $\mathbf{X}_i^t$ as the time series data to be protected. $\mathbf{X}_i^t$ is collected for a period of time and stored locally at time t on the sensor-based device i . $\mathbf{X}_i^t$ is a matrix of w rows and d columns, and each column denotes measurements associated with a particular attribute/feature. For simplicity, in the following, denote $\mathcal{X} = \mathbf{X}_i^t$.

For a function $f(x, y)$ of size $w \times d$, the DFT decomposition in the frequency domain is defined as:

$$F(u, v) = \sum_{x=0}^{w-1} \sum_{y=0}^{d-1} f(x, y) e^{-2\pi i (\frac{ux}{w} + \frac{vy}{d})}. \quad (2.12)$$

The exponential in Equation 2.12 can be expanded into sine and cosine components, and the variables u and v determine their frequencies.

The following equation describes the inverse of the above discrete Fourier transform:

$$f(x, y) = \frac{1}{w \cdot d} \sum_{u=0}^{w-1} \sum_{v=0}^{d-1} F(u, v) e^{2\pi i (\frac{ux}{w} + \frac{vy}{d})}. \quad (2.13)$$

The value of the Fourier transform at the origin $F(0, 0)$ of the frequency domain is called the Dominant Component and represents $w \cdot d$ times the average value of $f(x, y)$.

As it is a base-2 algorithm, FFT assumes that the number of points d to be processed satisfy the relationship $d = 2^\gamma$, where γ is an integer value [80]. This results in essential savings in computation time, hence reducing the complexity of DFT from $O(n^2)$ to $O(n \log n)$.

Thus, matrix $\mathcal{X}$ needs to be transformed such that its width and height are integers power of 2. The simplest way to achieve this is to zero-pad the matrix. The size of the new zero-padded matrix $\bar{\mathcal{X}}$ is $\bar{w} \times \bar{d}$, where both $\bar{w}$ and $\bar{d}$ are integers power of 2. The next step is to move the Dominant Component located at $F(0, 0)$ to the center of the transformed matrix, $G(\frac{w}{2}, \frac{d}{2})$, to perform the filtering more efficiently. This can be achieved by multiplying each entry (x, y) of the $\bar{\mathcal{X}}$ matrix with $e^{\pi i (x+y)}$ [81]. The new matrix is denoted by $\hat{\mathcal{X}}$, and its size remains the same $\bar{w} \times \bar{d}$.

The first transformation in the FFT-based data perturbation is filtering. Low-pass filtering, high-pass filtering, or other types of filtering can be applied in the frequency domain to modify the original sensor data. After applying the filter, additional transformations (e.g., adding a Gaussian white noise) can increase perturbation and decrease the chances of restoring the exact original signal.

Let the matrix $\bar{\mathcal{X}}_+$ be the result of all transformations performed in the frequency domain. Then, the perturbed/distorted data from the matrix $\bar{\mathcal{X}}_+$ is restored in the matrix $\bar{D}$ by applying the inverse fast Fourier transform. As the size of the distorted matrix $\bar{D}$ is larger than that of the original matrix $\mathcal{X}$, due to the zero-padding, only the data from the top-left corner of the matrix is considered. Therefore, $\bar{D}$ is cropped to matrix D of size $w \times d$.

In a nutshell, by applying the above-described FFT-based data distortion procedure the matrix D is obtained, which is similar to $\mathcal{X}$, but not identical.

As stated earlier, once the data is transformed to the frequency domain, a primary filter can be applied to cut off a selection of frequencies. Several filters, often used in image processing [82], can be applied, such as low-pass, high-pass, band-reject, band-pass, and threshold filters.

When applied to sensor measurements, the data restored from the frequency domain to the original time domain must maintain its main characteristics while hiding sensitive data. Our approach involves applying low-pass filters and cutting off the high frequencies from the

Fourier space, usually regarded as noise frequencies while keeping the Dominant Component unaltered. Thus, the filtered matrix $\bar{G}$ becomes:

$$\bar{F}(u, v) = H(u, v) \cdot F(u, v), \quad (2.14)$$

where H is a low-pass filter function that removes high frequencies, and $\bar{F}$ contains the remaining Fourier frequencies after applying the filter.

Further, the ideal 2D low-pass filter (ILPF) [82] is applied to the data. This filter keeps all frequencies inside a radius f_c and discards all the rest. The ILPF involves a simple computational process, and it can be obtained rapidly:

$$H(u, v) = \begin{cases} 1, & \text{if } \sqrt{u^2 + v^2} \leq f_c \\ 0, & \text{otherwise,} \end{cases} \quad (2.15)$$

where f_c is the cut-off frequency. Figure 2.3 illustrated the effects of applying a low-pass filter on data representing the speed of a vehicle. The data is distorted as more frequency components are filtered. The data distortion is utilized as a protection mechanism as real values are obscured.

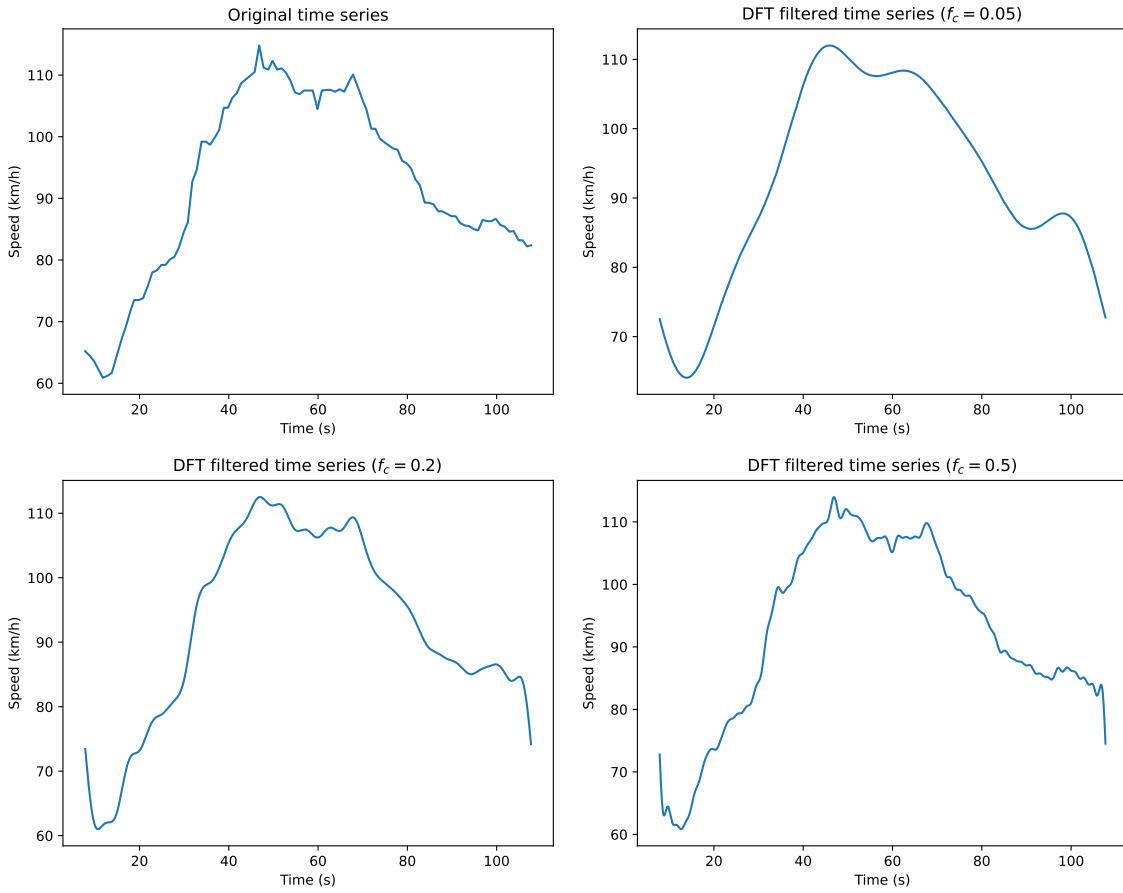


Figure 2.3 Example of DFT filtered speed data values using various cut-off frequencies.

2.2.2 Privacy and utility metrics

After applying a perturbation mechanism, it is essential to evaluate the utility of the resulting data to ensure that sufficient information is retained for further analysis and processing, such as anomaly or tampering detection [83, 38, 84]. Simultaneously, the effectiveness of the privacy protection must be assessed to determine the extent to which sensitive data remains protected.

When evaluating privacy and utility, two key aspects are considered [56]: the Privacy Loss (PL) (also called *disclosure risk* [85]) and the Information Loss (IL) (or *approximation error*). Privacy loss quantifies the extent to which an attacker can extract meaningful information from the perturbed data, while, information loss measures the reduction of the statistical utility of the perturbed data compared to the original one. Therefore, the ideal approaches minimize privacy loss (maximize perturbation) and the loss of information (maximize utility).

The PL is described as the difficulty level in estimating the original values from the perturbed ones [56]. Privacy loss refers to the capacity of the user to obtain meaningful and close-to-real information about the original data from the perturbed data. One approach to estimating the loss of privacy is by considering the variance of the random noise used by the perturbation [46]. For quantifying PL, three main approaches have been proposed [43]: measure how closely the original values of a perturbed attribute can be estimated [46, 61], use information theory [86], or, use the notion of privacy breach [87]. However, in practice, the PL is closely tied to the specific threat model, and various metrics can be used depending on the context and nature of the adversarial threat. Different scenarios may require different PL metrics to accurately illustrate the effectiveness of a privacy-preserving mechanism against potential attacks.

Data utility is equally essential when it comes to data privacy. After applying a randomization mechanism, it is essential to measure the retained information to ensure that sufficient data remains available for meaningful analysis [32]. Utility metrics have a crucial role for data analysts and data processing applications, serving as a benchmark for evaluating the effectiveness of different randomization techniques.

For time series data, measuring utility involves evaluating the *information loss* caused by the randomization process. Various error measures [88] can be computed (see Table 2.1) between two corresponding values (in our notation, x_i and x'_i).

Table 2.1 Error measures for the error e_i (α and β are positive constants) [88]

Error type	Notation	Definition
<i>absolute error</i>	e_i^{abs}	$ x_i - x'_i $
<i>relative error</i>	e_i^{rel}	$\frac{ x_i - x'_i }{\max\{1, x_i\}}$
<i>modified relative error</i>	$e_i^{m_rel}$	$\frac{ x_i - x'_i }{\max\{1, \min\{x_i, x'_i\}\}}$
<i>combined error</i>	e_i^{comb}	$\min\{\alpha \times e_i^{abs}, \beta \times e_i^{rel}\}$
<i>modified combined error</i>	$e_i^{m_comb}$	$\min\{\alpha \times e_i^{abs}, \beta \times e_i^{m_rel}\}$

To compute the approximation error for the entire time series, a suitable norm must be selected. Let $E = e_1, e_2, \dots, e_N$ represent the total error, where each e_i denotes the error between corresponding values. The p -norm average error for $p > 0$ is given by [88]:

$$\|E\|_p = \left(\frac{1}{N} \sum_{1 \leq i \leq N} e_i^p \right)^{1/p} \quad (2.16)$$

Thus, various norms can be calculated (for $p > 0$) for different types of errors. The most common are 1-norm and 2-norm (see Table 2.2).

Table 2.2 The definitions of the most common mathematical norms [88].

Norm	Notation	Definition
1-norm average error	$\ E\ _1$	$\frac{1}{N} \sum_{1 \leq i \leq N} e_i$
2-norm average error	$\ E\ _2$	$\sqrt{\left(\frac{1}{N} \sum_{1 \leq i \leq N} e_i^2 \right)}$

The state-of-the-art literature lists several common metrics for measuring information loss (utility), such as variance [89, 90], Mean Relative Error (MRE) [91, 92, 75], Mean Squared Error (MSE) [77] and Mean Average Error (MAE) [61, 93] (Table 2.3). For all mentioned metrics, the smaller the value, the better the utility. In this research, we mainly rely on MAE to measure the utility (Equation 2.17), a metric often utilized for comparing time series perturbation methods [61, 94]:

$$MAE = \frac{1}{N} \sum_{i \in N} |X_i - X'_i|, \quad (2.17)$$

where N is the length of the time series, X is the original data vector, and X' is the perturbed data.

For location data, the utility metrics are divided into three categories: global-level, trajectory-level, and semantic-level metrics. Global-level metrics (Table 2.3), such as *Density error*, *Relative query error*, *Hotspot query error*, and *Kendall-Tau* coefficient, assess the overall preservation of spatial and density patterns between real and protected datasets, as well as the maintenance of high-density areas (hotspots) [95].

Trajectory-level metrics evaluate individual trajectory characteristics and alignment. Metrics like *Trip error*, *Length error*, and *Diameter error* measure the fidelity of starting and ending points, trajectory lengths, and trajectory spread [96, 97, 98]. Additionally, trajectory similarity is captured through micro-level metrics such as *Hausdorff distance* [99], *DWT* [100], and *Edit Distance on Real Sequences (EDR)* [101], providing a detailed assessment of alignment between real and protected individual trajectories.

Semantic-level metrics focus on frequently observed movement patterns in the data. Metrics such as *Pattern F1* and *Pattern error* evaluate the consistency of common travel sequences, ensuring that patterns frequently occurring in real trajectories are similarly represented in the

protected data [95]. Together, these metrics offer a comprehensive view of data utility across different levels of trajectory analysis.

Table 2.3 The utility metric functions and application scenarios. [102]

Utility metric	Utility metric functions	Application scenario
Variance [89, 90]	$Var = \int_{x \in \mathbb{R}} (x - \mu)^2 f(x) dx$, where $f(x)$ is the probability density function and μ the mean.	Noise distribution optimization
MSE [61, 93]	$MSE(X, X') = \frac{1}{n} \sum_{i=1}^n (x_i - x'_i)^2$.	Data release/publication
MAE [77, 103]	$MAE(X, X') = \frac{1}{n} \sum_{i=1}^n x_i - x'_i $.	Data release/publication
MRE [91, 92, 75]	$MRE(X, X') = \frac{1}{n} \sum_{i=1}^n \left \frac{x_i - x'_i}{x_i} \right $.	Data release/publication
Euclidean Distance	$d_E(X, X') = \sqrt{\sum_{i=1}^n (x_i - x'_i)^2}$.	Location/trajectory privacy
Hausdorff Distance [104]	$d_H(X, X') = \max\{\sup_{x \in X} \inf_{x' \in X'} d(x, x'), \sup_{x' \in X'} \inf_{x \in X} d(x', x)\}$.	Location/trajectory privacy
Density Error [95, 105]	$DE = JSD(\mathcal{D}(\mathcal{T}_r), \mathcal{D}(\mathcal{T}_p))$, where $JDS(\cdot)$ is the Jensen-Shannon divergence.	Location/trajectory privacy
Relative Query Error [95, 105]	$RQE = \frac{ Q(\mathcal{T}_r) - Q(\mathcal{T}_p) }{\max\{Q(\mathcal{T}_r), s\}}$	Location/trajectory privacy
Hotspot Query Error [95, 105]	$HQE = 1 - \frac{\sum_{C_i \in \mathcal{H}_p} \text{rel}(C_i) / \log(\text{rank}_{\mathcal{H}_p}(C_i) + 1)}{\sum_{j=1}^{n_h} 1 / (j \cdot \log(j + 1))}$	Location/trajectory privacy
Kendall's Coefficient [95, 105]	$\text{KendallTau} = \frac{N_c - N_d}{ C (C - 1)/2}$	Location/trajectory privacy

2.3 Data privacy and utility in the context of multivariate time series classification

Ensuring privacy while maintaining data utility in multivariate Time Series Classification (TSC) poses additional challenges, as sensitive information can be inferred from temporal dependencies and feature correlations. To address this, privacy-preserving techniques are designed to reduce the risks without compromising classification accuracy (data utility). This section explores data privacy and utility in the context of TSC, covering key aspects such as classification methodologies (commonly used models and their privacy implications), feature selection strategies, and data perturbation techniques.

2.3.1 Time series classification

TSC refers to the task of assigning a label or category to a given time series data based on its features, patterns, or trends. A classifier model produces a function:

$$f : \mathbb{R}^d \longrightarrow \mathcal{C}, \mathcal{C} = \{1, \dots, n\}, \quad (2.18)$$

where d represents the number of features and n is the number of categories that is applied to any input $\mathbf{X}$. The function $f(\mathbf{X})$ is regarded as an estimate of the category that $\mathbf{X}$ belongs to [106]. Several approaches have been proposed for time series classification, such as distance-based models [107, 108], shapelet-based models [109, 110], ensemble-based models [111, 112, 113], and deep learning models [114, 115]. One of the most popular TSC approaches is the use of a nearest neighbor (NN) classifier combined with a distance function [111], particularly the Dynamic Time Warping (DTW) distance. Other approaches involve the use of traditional machine learning algorithms such as k-Nearest Neighborss (kNNs), Support Vector Machiness (SVMs), and Decision Trees, which extract discriminative features from time series data. Another widely adopted method is based on feature extraction followed by classification algorithms (e.g., Random Forests, Gradient Boosting). Additionally, deep learning techniques, such as Recurrent Neural Networks (RNNs), have shown their utility in time series classification tasks, as they capture temporal dependencies and learn complex data patterns. These models include variants like Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) models, which have demonstrated remarkable performances [116, 117].

In the domain of TSC with machine learning methodologies, data privacy is attained through data perturbation, data transformation, homomorphic encryption [118], privacy-preserving classification models, and federated learning. Data perturbation methods [27] involve the injection of noise into each data point within the time series dataset before starting the classification procedure. This action serves to obscure the original data, thereby maintaining data confidentiality. A recent approach [119] proposes the transformation of the time series data into a graph, called the Time Labeled Visibility Graph (TLVG), that protects privacy while retaining useful information for classification. In this case, the classification is performed by a Graph Neural Network (GNN). Additionally, the potential application of homomorphic encryption in time series data classification has been examined, and various methods have been proposed [120, 121]. The construction of privacy-preserving classification models is another possible approach. Such models (e.g., DP-SGD [122], PART-GAN [123]) integrate perturbations during the model construction phase, which acts as a safeguard against the potential recreation of sensitive information from the models once they are deployed. Federated learning is a paradigm that processes data on local devices, with the overarching classification model being an ensemble of several localized models. The proposed approaches [124, 125] ensure that personal data is processed locally, significantly mitigating the risks associated with central data storage and management. However, a considerable number of existing

methodologies [122, 120, 119] overlooks an adversarial model characterized by a powerful classifier.

2.3.2 Feature selection approaches

Feature selection is essential for improving classification accuracy, reducing computational complexity, and enhancing model interpretability [126]. Beyond its primary objectives, feature selection has also been explored as a privacy-enhancing tool [127, 128, 129]. Zhang et al. [127] proposed a feature selection technique utilizing feature importance in a differentially private manner. It aims to reduce data correlation. The approach is based on selecting a threshold, which is critical for determining the importance of features while implementing differential privacy. The utility of feature selection for privacy protection was further demonstrated by Jafer, Matwin, and Sokolova [130]. Their research showed that even without explicit privacy considerations, the process of eliminating irrelevant or redundant attributes could improve privacy. However, this method lacks explicit privacy-aware considerations during the feature selection process. Addressing the need for a privacy-aware feature selection, an automatic feature selection framework that considers both privacy and utility objectives simultaneously and utilizes a two-dimensional measure was introduced [131]. The adversarial model considered in their study is based on the *uninformative principle*, suggesting that attackers gain minimal information about sensitive data from the released dataset.

In the context of collaborative data classification, Sheikhalishahi and Martinelli [128] proposed a framework that highlights privacy protection for data providers willing to share their data. This framework introduces a privacy mechanism that eliminates irrelevant features, thereby balancing accuracy and privacy to maintain data utility for classification. Another approach [129] extended the concept of privacy-preserving in collaborative data sharing by proposing a model that incorporates clustering, anonymization, and feature selection techniques. The model identifies features using Symmetrical Uncertainty (SU) to determine attributes that contribute significantly to data utility. It then applies Spearman's Correlation Coefficient to eliminate redundant features and reduce data dimensionality. Thus, the approach increases privacy without compromising data utility for classification purposes.

Existing privacy-aware feature selection methods do not explicitly incorporate privacy constraints in the selection process. As a result, adversaries can still exploit correlations between retained features to infer private information. Thus, existing feature selection methods are insufficient for privacy protection. Developing feature selection methods specifically designed for data protection in the context of time series classification remains an open research challenge.

Another significant challenge is the lack of robustness against adversarial attacks. Most feature selection techniques are designed to optimize predictive performance but do not take into account potential attacks, where adversaries use powerful classifiers to identify or re-identify users. Additionally, existing feature selection use importance metrics (e.g., mutual information, entropy, and correlation), which do not quantify privacy risks. Furthermore,

privacy requirements vary across applications and users, making static feature selection strategies ineffective in dynamic contexts. To address these challenges, this thesis develops an adaptive, privacy-oriented selection method that considers adversarial threats (using a privacy-breaking classifier) and balances data privacy and utility.

2.3.3 Data perturbation approaches

The privacy guarantees provided by many perturbation algorithms are typically robust for single data collections but degrade significantly in the context of continuous, infinite streams, such as daily app usage statistics [72]. Traditional DP and LDP algorithms are often inadequate in these scenarios, as they are designed for one-time data collection and not for repeated reporting [75, 24, 132, 23].

The advantage of DP/LDP mechanisms lies in their simple trust model: even if a single report is compromised, the user remains protected due to differential privacy guarantees. This holds even when multiple parties conspire against the user. However, this model oversimplifies real-world data collection scenarios, which often involve continuous monitoring at regular intervals and multiple correlated attributes. In these cases, DP/LDP guarantees can degrade rapidly. As the number of correlated reports grows, privacy protections weaken exponentially. Mechanisms such as Google’s Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR) [71] attempt to solve this deterioration by employing memoization techniques, but at the cost of reduced data accuracy.

Sensor-generated data is often high-dimensional, meaning time series data may consist of multiple attributes. When applying w -event LDP to such data (presented in Section 2.2.1.2), each attribute receives a reduced privacy budget (e.g., ϵ/d , where d is the number of attributes) to maintain overall privacy guarantees. This allocation, however, introduces significantly higher noise, which diminishes data utility. As the number of attributes increases, the noise added to each one intensifies, leading to a substantial decline in data quality. Alternative perturbation methods, such as those based on DFT (discussed in Section 2.2.1.3), can be applied effectively but offer limited privacy at the value level.

In multivariate time series data and TSC, where attributes are often interdependent, applying traditional differential privacy mechanisms independently to each attribute may fall short of providing robust protection. Kifer and Machanavajjhala [133] show that privacy mechanisms that overlook attribute correlations can lead to privacy leaks. Due to these dependencies, even perturbed data can be vulnerable to sophisticated inference attacks, such as cross-checking, which exploit correlations to reveal sensitive information. Thus, achieving strong data privacy in multivariate time series data, particularly within TSC, remains an ongoing challenge.

2.4 Data privacy and utility of location data

Location data, particularly in the form of trajectory records, is highly sensitive as it can reveal personal movement patterns and behaviors. This section presents privacy-preserving strategies for trajectory data, which protect movement information while preserving analytical value. Privacy-preserving synthetic trajectory generation is introduced as a method for producing realistic trajectory data. These approaches use machine learning and generative models to balance privacy protection with the need for useful mobility data.

2.4.1 Trajectory data privacy

In time series data, observations are recorded sequentially over time. Similarly, trajectory data records the path or movement of an object, usually over time and space, with timestamps marking each position. This spatial-temporal nature of trajectory data makes it a specific type of multivariate time series data, where each entry typically includes both spatial (e.g., latitude, longitude) and temporal (timestamps) components.

Trajectory privacy protection has been extensively studied in recent years, driven by the increasing use of Location-Based Services (LBS) and the resulting rise in trajectory data collection. Traditional Location Privacy Protection Mechanisms (LPPM) approaches to trajectory privacy include removing explicit identifiers (e.g., user names or IDs) and applying generalization or perturbation techniques to the spatial and temporal characteristics of the data [134]. However, these approaches often fall short in preventing re-identification due to the inherent uniqueness of individual movement patterns [135]. This has led to the exploration of more sophisticated methods that focus on maintaining privacy while preserving the utility of the trajectory data.

Current research on trajectory privacy includes three directions: DP-based or LDP-based methods [105, 95], spatial perturbation techniques, and synthetic trajectory generation. DP has been applied to trajectory data by ensuring that each individual trajectory is indistinguishable from a group of similar trajectories, converting the problem into a k -anonymity problem [136]. Techniques such as spatial cloaking and mix-zones have been used to create anonymity groups, making it difficult to link trajectories to specific users [137]. However, these approaches often suffer from reduced data utility and limit their applicability for certain types of spatial analyses.

Another research direction focuses on spatial perturbation or geomasking techniques. They involve modifying the coordinates of trajectory points to obfuscate the original locations while preserving the spatial patterns. Methods such as random perturbation, Gaussian noise addition, and grid masking have been used to protect location privacy while retaining spatial utility for applications such as spatial clustering and hotspot detection [138, 139]. However, these techniques primarily focus on the spatial dimension and often overlook the temporal and semantic characteristics of the data, which can also serve as quasi-identifiers for re-identification [140, 141].

2.4.2 Privacy-preserving synthetic trajectory generation

Synthetic trajectory generation aims to produce artificial movement data that mimics real-world trajectories while protecting sensitive information. Various deep learning-based methods have earned popularity for trajectory privacy protection. Generative models produce synthetic trajectories that maintain the utility of the original data while offering stronger privacy guarantees [142, 143]. Generative Adversarial Networks (GANs) and LSTM networks have been employed to learn the patterns of real trajectories and generate synthetic ones that closely emulate the originals [144, 145]. This approach addresses the limitations of traditional methods by capturing the dependencies between spatial, temporal, and thematic attributes. They are able to generate realistic trajectories that are difficult to link back to specific users.

Several existing trajectory-generating methods can be used for real-time data release [95, 105, 144, 146, 145]. However, the process involves a delicate trade-off between the degree of privacy and system performance, as higher levels of privacy typically incur increased computational overhead and reduced data utility. Addressing these challenges remains a critical area of research, particularly in the context of dynamic, real-time data release.

2.5 Summary

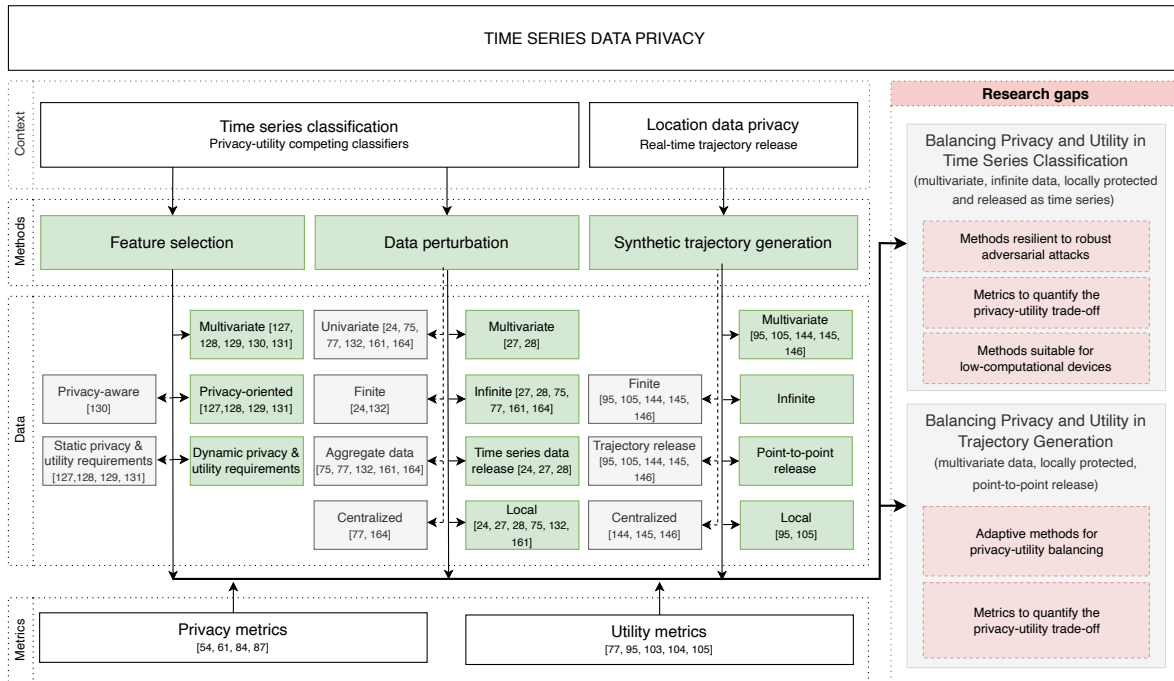


Figure 2.4 Overview of the most relevant state-of-the-art works, including the context, methods, data processing approaches, metrics, and identified research gaps.

This chapter presents an overview of the most relevant research areas related to utility and privacy-preserving techniques applied to time series data (Figure 2.4). It highlights the

need to protect sensitive information collected from sensors and processed as a multivariate time series. Various privacy-preserving techniques are examined, including perturbation, encryption, anonymization, and data transformation. Perturbation methods such as DP and LDP are highlighted for their effectiveness and simplicity. Additionally, a transformation-based method using Fourier Transform distortion is presented, which obfuscates data by modifying its frequency components. However, for specific scenarios, classical approaches do not ensure the required protection, and they do not achieve the needed trade-offs between privacy and utility.

In this thesis, we examine two contexts: the first refers to data privacy in TSC, and the second relates to data privacy concerning location data. In the first scenario, we acknowledge feature selection as a method for improving both classification accuracy and privacy, as it reduces data dimensionality and eliminates specific attributes. Further, we explore techniques such as perturbation and various machine learning models. Location privacy and trajectory data are examined as specific cases of time series data, where the spatial-temporal nature of the data introduces unique privacy challenges. Techniques such as spatial perturbation, synthetic trajectory generation, and DP-based approaches are discussed.

Despite the existence of various privacy-preserving techniques, several research gaps (Figure 2.4) have been identified in the considered contexts. Resistance to adversarial attacks is a critical area, as robust classifiers can exploit correlations in multivariate data. Furthermore, there are no universally accepted metrics for evaluating the trade-offs between privacy and utility. Scalability remains a significant challenge, and approaches that can be implemented locally and at the device level with minimal computational resources are required. Additionally, the development of adaptive methods (including feature selection) that adjust based on user-defined privacy preferences and data context is necessary. Finally, synthetic data generation methods need improvement to balance the privacy and utility of the resulting trajectory data.

This chapter highlighted the complexity of preserving privacy in time series data while ensuring its utility. It advocates for solutions that address scalability, robustness, and dynamic privacy-utility trade-offs, solutions that are effective and applicable in real-world scenarios.

Chapter 3

Feature Selection for Balancing Privacy and Utility in Multivariate Time Series Data

This chapter is based on the work presented in the following articles:

- **Adrian-Silviu Roman**, Béla Genge, and Roland Bolboacă. “Privacy-Oriented Feature Selection for Multivariate Time Series Classification”. In: *Procedia Computer Science* 246 (2024). 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024), 2024. [37]
- **Adrian-Silviu Roman**, Béla Genge, and Piroska Haller. “Balancing Privacy and Utility in Multivariate Time-Series Classification”. In: *Computer Security. ESORICS 2024 International Workshops. Springer (2025)*. 19th International Workshop on Data Privacy Management (DPM 2024), 2024. [25]

This chapter presents feature selection as an essential step in improving data privacy for multivariate time series data collected from sensors, particularly in the context of Time Series Classification (TSC). By selecting and managing features, the presented approach lowers privacy risks and maintains data utility, making it an important strategy for releasing sensitive sensor data. Feature selection serves multiple purposes, including improving classification accuracy, reducing computational complexity, and enhancing model interpretability [126]. Importantly, it has also been explored as a privacy-enhancing mechanism, as discussed in Section 2.3.2.

Addressing the research gaps, this chapter presents Feature Selection for Balancing Utility and Privacy in Multivariate Time Series (FS-BUP), a privacy-oriented feature selection approach that not only satisfies privacy objectives, even in the context of TSC, but also preserves data utility. This mechanism is designed to be configurable and to let the data owners achieve the desired balance between privacy and utility. To sum up, the chapter presents the following important contributions to the state of the art:

- It describes the problem of achieving privacy in the context of TSC, presenting a dual-model with two competing classifiers, an Utility-Focused Classifier (UFC) and a Privacy-Breaking Classifier (PBC).
- It defines the Feature Importance Gap (FIG), a score designed for feature ranking, to measure the significance of a feature for UFC compared to PBC.
- It presents a novel feature selection method for features that are more important for UFC than for PBC, with the objective of decreasing the classification accuracy of the PBC while maintaining the accuracy of UFC to suitable levels. The method relies on two feature selection techniques based on importance coefficients, the first one utilizing FIG, while the second one is based on the K-means clustering.
- It evaluates the effectiveness of the presented methodology on two driving [147, 148] and two Human Activity Recognition (HAR) datasets [28, 149].

3.1 Problem statement and research gaps

The identified research gaps (Figure 2.4) highlight the need for privacy-preserving techniques compatible with low-computation environments. As a pre-processing technique, feature selection requires no additional computational resources. It is particularly valuable in scenarios where data precision is crucial and perturbation-based methods are unsuitable (e.g., in the medical field). Moreover, feature selection can be resilient against adversaries by reducing the volume of sensitive data released, thereby limiting exposure. However, existing privacy-aware feature selection methods fail to explicitly integrate privacy constraints, leaving selected features vulnerable to inference attacks based on hidden correlations. Additionally, these

methods are vulnerable to adversarial attacks, as they do not consider adversaries who use robust classifiers for re-identification.

Consider the data collected from sensor-based devices as a multivariate time series, transmitted to a central database, and then shared with third-party processors for classification (Figure 3.1). Within this scenario, two types of classifiers are examined: (i) a UFC, which identifies patterns that are not related to user identities, and (ii) a PBC, designed to expose the identities of users within the data. The goal is to present a method for feature selection that reduces the classification accuracy of the PBCs in favor of increased privacy. At the same time, the classification accuracy of the UFC or UFCs should remain high enough to maintain the overall utility.

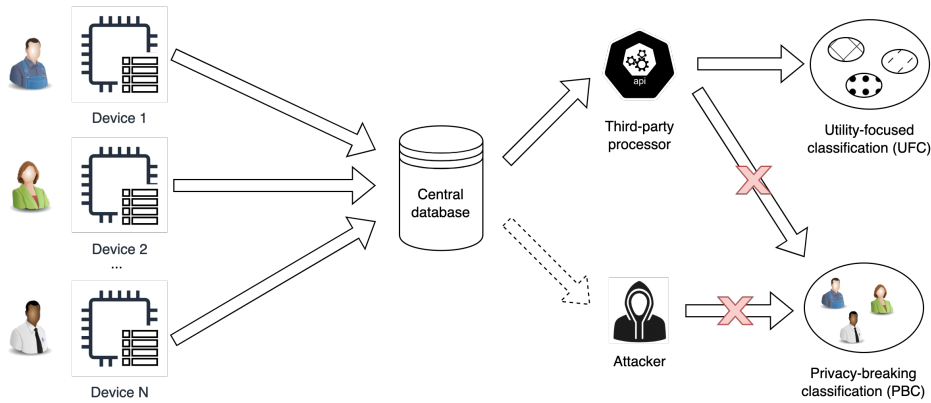


Figure 3.1 Problem description. Multivariate time series data collected from several sensor-based devices is transmitted to a centralized database. The selected features should decrease the accuracy of PBC, while maintaining the accuracy of the UFC.

Notably, the following assumptions are considered for the presented dual-model framework:

- No prior knowledge of the classification models is used by UFC or PBC. Thus, the framework ensures generality and that it can be used in various scenarios. Moreover, this assumption reflects real-world scenarios where attackers or defenders do not have detailed information about the opposing models.
- The process of training the UFC model is performed in a secure environment. In both the training and testing phases of constructing the UFC, data labels for each batch of records are stored in the data warehouse. Afterward, there is the possibility to remove the data labels from the database for privacy purposes. No user label information is shared with third-party systems.
- The scenarios involving breaches of privacy, as assessed in the adversarial or threat model, include: (i) the “honest-but-curious” entities, such as a data analyst who has access to all the data needed to construct a classifier that compromises privacy; (ii) an attacker (Figure 3.2) that is capable of constructing a PBC for identifying the users by

getting access to various data sources (training dataset with labels, the unlabeled data collected from the devices, and other observable data).

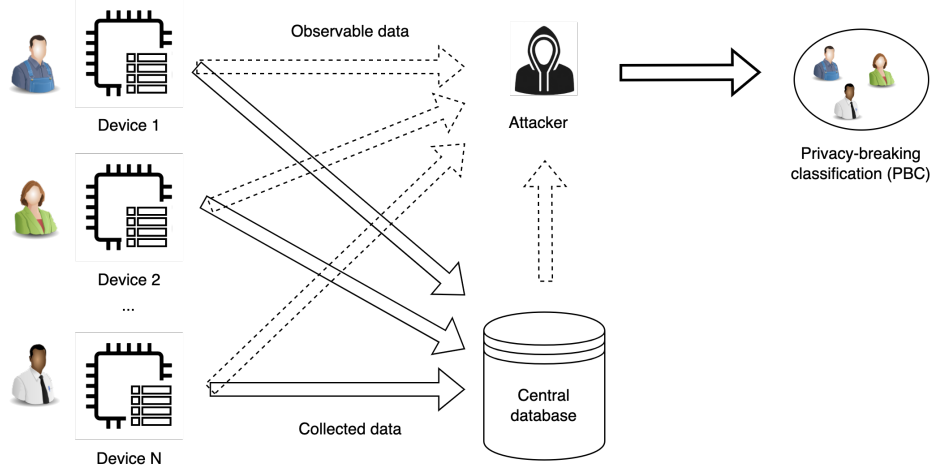


Figure 3.2 The adversarial model. The attacker gets access to the necessary data for building a capable PBC.

Use case: Human activity recognition in wearable fitness tracking. Consider wearable fitness trackers that collect data on human activity and send it to a health monitoring service through sensors in the device, which transmit the information to the provider's servers via a mobile app. The health monitoring service aims to classify different types of physical activities (e.g., walking, running, or cycling). Additionally, the service provider must protect the privacy of the users against potential threats, such as unauthorized access to the collected data. Moreover, the development of PBC models trained to identify individuals based on their activity patterns should not be possible. The presented method, FS-BUP, can be utilized at the design phase of the application or later, during testing, to identify the features to be collected such that resulting data could be utilized for activity recognition while minimizing the risk of user identification.

3.2 The FS-BUP method

Consider the continuously collected data consisting of records with d features transmitted to a central location for processing. Let $\mathbf{X}_i^t$ represent the time series data collected by sensor-based device i at time t . This data, $\mathbf{X}_i^t$, is structured as a matrix of w rows and d columns, where values on each column correspond to data collected from a specific sensor.

TSC involves assigning a label or category to time series data based on its patterns, trends, or features. Consider a classifier model $f : \mathbb{R}^{w \cdot d} \rightarrow \mathcal{C}, \mathcal{C} = \{1, \dots, n\}$, where w is the number of records in the considered window, and n the number of categories. The model is applied to any input $\mathbf{X}_i^t$, producing the corresponding classification output. The function $f(\mathbf{X}_i^t)$ provides an estimation of the category to which $\mathbf{X}_i^t$ belongs [106].

Further, let $f_p : \mathbb{R}^{w \cdot d} \rightarrow \mathcal{C}_p$ and $f_u : \mathbb{R}^{w \cdot d} \rightarrow \mathcal{C}_u$ represent two classification functions corresponding to the UFC and PBC models. These functions determine the class to which the input $\mathbf{X}_i^t$ belongs. f_p and f_u corresponds to classifications in $\mathcal{C}_p$ and $\mathcal{C}_u$, respectively. In the case of PBC, $f_p(\mathbf{X}_i^t)$ identifies the set of collected values as belonging to a user from a specified class $\mathcal{C}_p$, compromising user privacy. Conversely, for the UFC, the function $f_u(\mathbf{X}_i^t)$, categorizes non-sensitive information from a class $\mathcal{C}_u$.

Let $\mathbf{Y}_{pi}^t$ be the label information for user identification, and let $\mathbf{Y}_{ui}^t$ represent the label for classifying non-user-related data. The classification accuracy, which shows the number of correct classifications, for the two classifiers f_u and f_p is defined as follows:

$$\mathcal{A}_u = \frac{1}{N \cdot T} \sum_{i=1}^N \sum_{t=1}^T \mathbb{I}(\mathbf{Y}_{ui}^t = f_u(\mathbf{X}_i^t)), \mathcal{A}_p = \frac{1}{N \cdot T} \sum_{i=1}^N \sum_{t=1}^T \mathbb{I}(\mathbf{Y}_{pi}^t = f_p(\mathbf{X}_i^t)), \quad (3.1)$$

where $\mathcal{A}_u$ is the accuracy of f_u , $\mathcal{A}_p$ is the accuracy of f_p , T the number of data collected records, N the number of sensor-based devices, and $\mathbb{I}(c)$ the binary indicator function returning 1 iff the condition c is true, and returns 0 if the condition c is false.

Further, consider the collection of available features of the dataset, denoted as $\mathcal{F} = \{F_1, \dots, F_j, \dots, F_d\}$. The objective of the FS-BUP approach is to identify a subset $\mathcal{F}^*$ of $\mathcal{F}$ with d^* features, $d^* < d$, such that:

$$\mathcal{A}_p^* \ll \mathcal{A}_p \text{ and } \mathcal{A}_u^* \approx \mathcal{A}_u, \quad (3.2)$$

where $\mathcal{A}_p^*$ and $\mathcal{A}_u^*$ are the classification accuracy values calculated similarly to Equation 3.1, for the selected subset of features $\mathcal{F}^*$. Therefore, $\mathcal{A}_p^*$ should be significantly lower than $\mathcal{A}_p$, while $\mathcal{A}_u^*$ should remain approximately equal to $\mathcal{A}_u$.

The presented FS-BUP method for selecting the possible feature subsets $\mathcal{F}^*$ consists of the following steps (Figure 3.3): (i) compute feature importance for the two classifications (UFC and PBC) using all available features in $\mathcal{F}$; (ii) cluster features based on the computed importance coefficients; (iii) select the features in $\mathcal{F}^*$, features that are important for UFC while excluding the features that determine a high classification accuracy for PBC; (iv) validate the selected features, taking into account the desired privacy and utility objectives (i.e., values of $\mathcal{A}_p^*$ and $\mathcal{A}_u^*$).

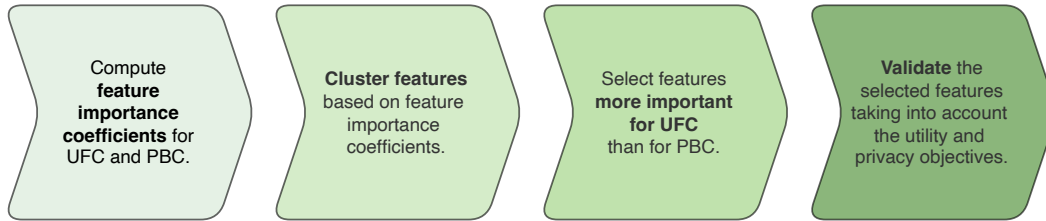


Figure 3.3 Steps of the FS-BUP method.

3.2.1 Computation of feature importance coefficients

The goal of computing feature importance [150] is to evaluate the significance of each feature in a dataset for a given classification task. A key advantage of this approach is the ability to assign feature scores without needing prior knowledge of the adversarial model. Instead, it relies solely on the available data labels and anticipated classifications.

Let $\mathbf{X}$ be the set of all $\mathbf{X}_i^t$ selected for the training phase with all available features in $\mathcal{F}$. By computing the feature importance for classifying $\mathbf{X}$ on class $\mathcal{C}_p$ using the labels $\mathbf{Y}_p$ and for classifying $\mathbf{X}$ on class $\mathcal{C}_u$ using labels $\mathbf{Y}_u$, I_u and I_p are obtained, the vectors that store the importance scores. Let $I_u = [i_1^u, i_2^u, \dots, i_d^u]$ and $I_p = [i_1^p, i_2^p, \dots, i_d^p]$, with $i_j^u, i_j^p \in [0, 1], \forall j \in \{1, \dots, d\}$.

To compute the feature importance coefficients, the Random Forest (RF) algorithm [150] was utilized, a popular method for feature ranking and importance evaluation. The goal was to identify the most relevant features for the considered classification tasks. RF was chosen for its ability to effectively handle high-dimensional, noisy data with minimal preprocessing. In this study, the impurity-based feature importance is utilized with Gini importance [150].

3.2.2 Importance-based feature clustering

The next step in identifying $\mathcal{F}^* \subset \mathcal{F}$ is to cluster $\mathcal{F}$ into two distinct subsets based on the calculated importance scores, I_p and I_u . These subsets are: $\mathcal{F}_u$, including features essential for UFC tasks, and $\mathcal{F}_p$, with features that play a significant role in resolving the PBC, such that $\mathcal{F} = \mathcal{F}_u \cup \mathcal{F}_p$. In the case of one UFC and one PBC, $\mathcal{F}^* = \mathcal{F}_u$.

The first clustering approach presented involves the computation of the FIG score, $\mathcal{S}_{F_j}$, defined as:

$$\mathcal{S}_{F_j}(i_j^u, i_j^p) = i_j^u - i_j^p, \quad (3.3)$$

where $i_j^u \in I_u$ and $i_j^p \in I_p$, for each feature F_j . I_u and I_p are the importance coefficients calculated for utility-focused and privacy-breaking classifications, respectively. The following rules for clustering are considered:

$$F_j \in \mathcal{F}_u \text{ if } \mathcal{S}_{F_j}(i_j^u, i_j^p) \geq \rho_I, \text{ and, } F_j \in \mathcal{F}_p \text{ if } \mathcal{S}_{F_j}(i_j^u, i_j^p) < \rho_I, \quad (3.4)$$

where ρ_I is the decision boundary. The score $\mathcal{S}_{F_i}$ determines when a feature shows a significantly greater significance in a classification context compared to another. Figure 3.4 exemplifies a group of ten features, their feature importance coefficients for an UFC and a PBC, and possible clustering of the features based on rules stated above, for $\rho_I = 0.0$ and $\rho_I = 0.05$.

The second presented clustering method involves using the K-means method [151], a widely utilized clustering algorithm. Consider $\mathcal{O}$ a set of d two-dimensional points o_j , where d is the number of features. Each point o_j is a pair of values (i_j^u, i_j^p) from vectors I_u and I_p , representing the importance coefficients of feature F_j . The objective is to determine k disjoint

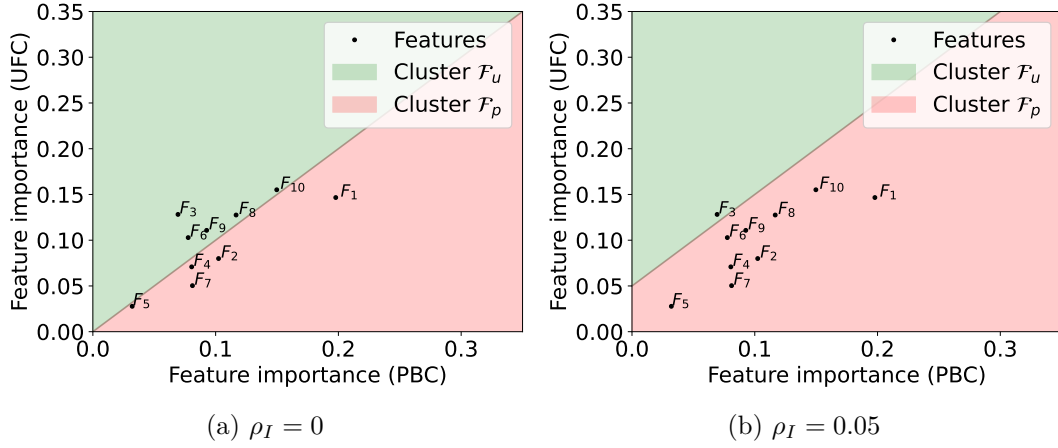


Figure 3.4 Examples of feature clustering based on FIG computation for two classifications (UFC and PBC).

clusters $\{C_1, \dots, C_k\}$, $k \geq 2$, with centroids $\{\mu_1, \dots, \mu_k\}$, such that the total Within Cluster Sum of Squares (WCSS) distances between each point and the mean of its cluster is minimized.

The features are clustered into two distinct groups: those that are important for UFC and those essential for PBC. The clustering approach is refined by selecting initial centroids to reflect the objective of separating representative features for UFC and PBC. Firstly, two initial centroids are selected for the clustering process, specifically from the set $\mathcal{O}$. These centroids are chosen as follows: one centroid is selected for its maximum value in I_u , indicating its high importance to UFC, and the other for its maximum value in I_p , highlighting its importance in PBC:

$$\mu^* = \arg \max_{(i_j^u, i_j^p) \in \mathcal{O}} i_j^u \text{ and } \mu^{**} = \arg \max_{(i_j^u, i_j^p) \in \mathcal{O}} i_j^p. \quad (3.5)$$

After the completion of the clustering process, the most significant cluster for UFC is selected. To identify this cluster, denoted as $\mathcal{F}_u$, the distances from the centroids to the decision boundary defined by condition $i_j^u = i_j^p$ are computed. The centroid of the cluster $\mathcal{F}_u$, denoted $\mu_u = (x_u, y_u)$, with $x_u \geq y_u$, lies at the maximum distance, d_{max} , defined as [152]:

$$d_{max} = \max \left(\frac{|x_i - y_i|}{\sqrt{2}} \right), \forall \mu_i = (x_i, y_i), i \in \{1, 2, \dots, k\}. \quad (3.6)$$

This procedure ensures that the chosen cluster $\mathcal{F}_u$, if it exists, includes the features favoring the UFC. Algorithm 1 summarizes the steps described above, and Figure 3.5 exemplifies a group of ten features, their feature importance coefficients for an UFC and a PBC, and a K-means clustering of the features, for $k = 2$ and $k = 3$.

The case of multiple utility objectives

The scenario involving multiple UFCs deserves particular consideration, as the data owner may be interested in multiple utility objectives. In such instances, the developed approach

Algorithm 1: Feature selection using importance-based K-means clustering for one UFC and one PBC

Input: $\mathcal{F}$ (vector of available features), I_u (vector of feature importance coefficients for UFC), I_p (vector of feature importance coefficients for PBC), k (number of clusters);

Output: The vector of selected features $\mathcal{F}^*$;

Function `kmeansFeatureSelection`($\mathcal{F}$, I_u , I_p , k):

```

// get the indexes of the largest values in  $I_u$  and  $I_p$ 
index_max_iu  $\leftarrow$  indexMaxValue( $I_u$ );
index_max_ip  $\leftarrow$  indexMaxValue( $I_p$ );

// select two initial clusters - Equation 3.5
 $\mu^* \leftarrow (I_u[\text{index\_max\_iu}], I_p[\text{index\_max\_iu}]);$ 
 $\mu^{**} \leftarrow (I_u[\text{index\_max\_ip}], I_p[\text{index\_max\_ip}]);$ 

// compute K-means clusters and centroids
 $C, \mu \leftarrow kMeans(\mathcal{F}, k, \mu^*, \mu^{**})$ 

// select the cluster with the centroid that lies at the maximum
// distance (Equation 3.6) above the decision boundary
 $\mathcal{F}^* \leftarrow C[\text{indexMaxDistance}(\mu)]$ 

return  $\mathcal{F}^*$ ;

```

End Function

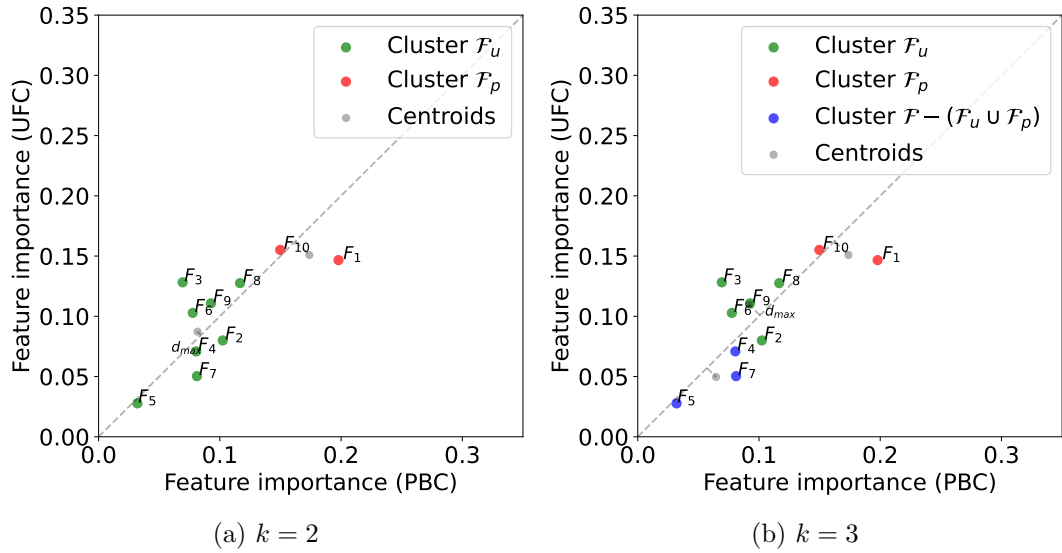


Figure 3.5 Examples of K-means-based feature clustering for two classifications (UFC and PBC). d_{max} is computed for selecting $\mathcal{F}_u$.

is adapted accordingly. Let U be the number of targeted UFCs. For the FIG-based feature clustering approach, consider $\mathcal{F}_{uj}$ the set of selected features for j -th UFC. $\mathcal{F}_{uj}$ is the set of

features important for the j -th UFC. The final set of selected features is the union of all $\mathcal{F}_{uj}$:

$$\mathcal{F}^* = \bigcup_{j=1}^U \mathcal{F}_{uj}. \quad (3.7)$$

In the case of K-means clustering, at least $U + 1$ clusters will be computed (U clusters for UFCs and one for PBC) with $U + 1$ initial centroids computed according to Equation 3.5. After the K-means algorithm completes the clustering, U clusters are selected, one for each UFC, by computing the maximum distance from the cluster to the decision boundary between the UFC cluster and the PBC cluster, according to Equation 3.6. Finally, the union of features from the U clusters forms $\mathcal{F}^*$. Figure 3.6 shows the clusters resulting in the case of two UFCs, for $k = 3$ and $k = 4$. In the case of two utility objectives with $k = 3$, the features are grouped into three clusters: one containing features important for the first utility objective, another for the second utility objective, and a third for the privacy-breaking classification. The selected feature set is then defined as $\mathcal{F}^* = \mathcal{F}_{u1} \cup \mathcal{F}_{u2}$. If four clusters are chosen ($k = 4$), the features in the cluster at the maximum distance (d_{max}) are discarded, and $\mathcal{F}^*$ is determined in the same manner as for $k = 3$. In real-world applications, the value of k is selected based on experimental evaluation.

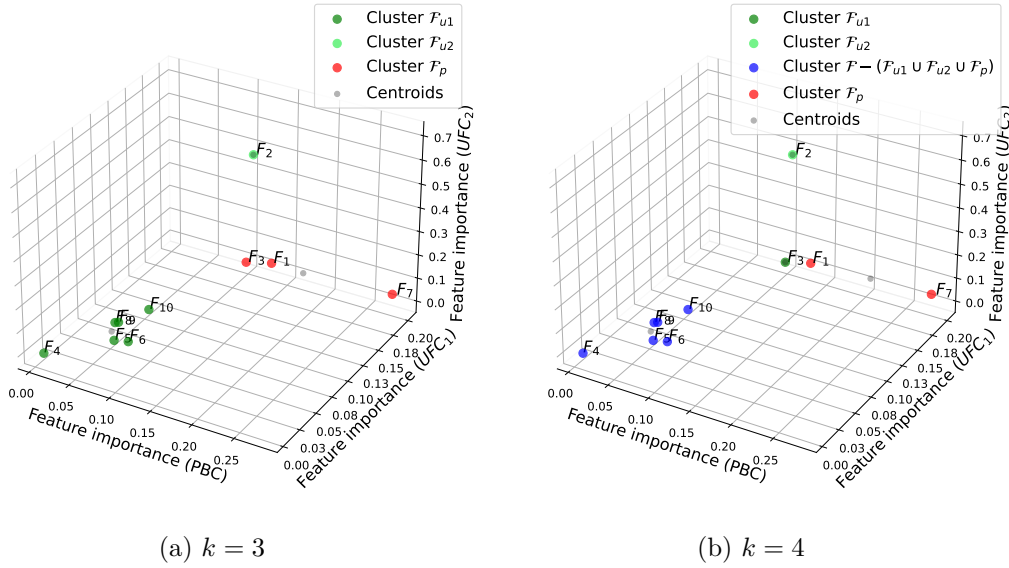


Figure 3.6 Examples of K-means-based feature clustering for multiple UFCs. In this case, $\mathcal{F}^* = \mathcal{F}_{u1} \cup \mathcal{F}_{u2}$.

3.3 Experimental evaluation

FS-BUP is evaluated using four datasets (Table 3.1), divided into two categories: first, two datasets with data collected from auto vehicles (UAH Dataset [147], HCRL Driving Dataset [148]), and second, two datasets focused on HAR (MotionSense [28], PAMAP2 [149]). The

experiments were conducted using TensorFlow Python libraries on a 2021 MacBook Pro with an Apple M1 Pro processor and 16 GB of RAM.

3.3.1 Datasets and data preprocessing

Consider the data generated by sensors embedded in modern automotive systems. In recent years, technological advances have fundamentally transformed the automotive industry. Modern vehicles are equipped with a large number of sensors that generate vast quantities of data processed by Electronic Control Units (ECU) [153]. ECUs are responsible for monitoring and controlling advanced vehicle features, such as automated parking or driverless functionality. As these vehicles become more and more integrated with external systems (e.g., cloud-assisted monitoring, remote diagnostics), data is processed both locally within the vehicle and externally by third-party systems to deliver functionality. However, this level of integration introduces privacy and security vulnerabilities [3, 4, 5, 38]. Other examples include the protection of data collected from wearable devices or the protection of localization data, which will also be addressed later in this work. In this context, the approach was tested on two driving and two HAR datasets.

The UAH dataset [147] was gathered from six drivers of various ages and vehicles, including a fully electric car. Each driver executed three distinct driving behaviors (normal, aggressive, and drowsy) across two different routes. The first route was a major highway with a speed limit of 120 km/h, covering a round trip of approximately 25 km. The second route was a secondary road with single lanes in each direction and a 90 km/h speed limit. Due to driving constraints, this route only simulated normal and drowsy behaviors with the electric vehicle. The study accumulated over 500 minutes of comprehensive driving data, encompassing real-world driving scenarios, fundamental data, supplementary information to contextualize the data, and video recordings of the drives. 17 features were extracted for TSC.

The HCRL dataset [148] includes driving data from ten drivers. Drivers have completed four separate trips on the same route on three different types of roads: urban, motorways, and parking lots. The dataset consists of 51 distinct features extracted from the Electronic Control Unit (ECU) of the vehicles involved. Data collection was carried out using the OBD-II protocol, with a 1Hz sampling frequency. Notably, the data was manually annotated to evaluate the UFC on the road type. After removing highly correlated features, a subset of 15 features was selected for this research.

The PAMAP2 (Physical Activity Monitoring) dataset [149] offers a large collection of sensor data for HAR research. Collected from nine subjects equipped with a heart rate monitor and three synchronized Inertial Measurement Units (IMUs), PAMAP2 includes 18 labeled activities from daily life (walking, cycling) and sports (soccer). Each subject's data is timestamped and contains 52 raw sensor readings (acceleration, gyroscope, magnetometer, orientation) from chest, ankle, and dominant hand IMUs, at a sampling frequency of 100 Hz. Additionally, heart rate data is included for each entry. 31 features were selected for the experiments.

The MotionSense dataset [28] collected physical activity data using smartphone sensors. A total of 15 trials were conducted using 24 participants in the same environment and conditions, performing actions such as walking upstairs, walking downstairs, walking, jogging, sitting, and standing. Measurements were collected with a sample rate of 50 Hz and included 12 features.

For data preprocessing, all datasets were normalized using standardization, which transforms the data to have a mean of 0 and a standard deviation of 1, as defined by the following formula:

$$x' = \frac{x - \mu}{\sigma}, \quad (3.8)$$

where σ is the standard deviation and μ is the mean of all considered values.

3.3.2 Experiments, results, and discussion

In the initial phase of the performed experiments, the classification accuracy was evaluated using the original set of features $\mathcal{F}$ to address both utility and privacy goals. The experiments were conducted using three distinct classification models: firstly, an LSTM-based technique, called FCN-LSTM (introduced by Karim *et al.* [154]) with the architecture illustrated in Figure 3.7, a state-of-the-art approach in the field of driver identification [155]; secondly, a SVM classifier, a classifier that maximizes the margin between data samples belonging to different classes; and thirdly, an XGBoost classifier, known for its efficiency and effectiveness in handling various types of data. During the data preprocessing phase, all datasets (Table 3.1) were normalized using a standard scaler to achieve a standard deviation of 1 and a mean of 0, for every feature. Additionally, the data was divided, allocating 70% for training purposes and the remaining 30% for testing. The results (Table 3.2) showed considerable accuracy levels across all employed classification techniques, even though the SVM classifier demonstrated relatively lower performance.

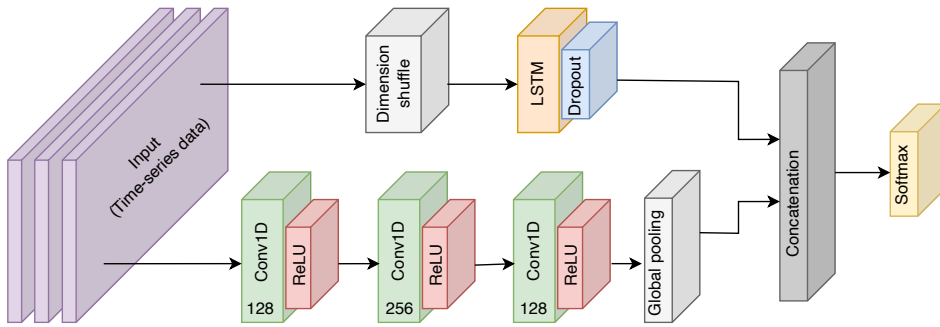


Figure 3.7 The FCN-LSTM architecture [154].

Further, the experiments involved the computation of feature importance for each classification objective, as explained in Section 3.2.1. Thus the relevance of each feature for user/driver detection and non-user-focused data classification (behavior, road type, activity type) was calculated. For instance, for the UAH dataset, the importance coefficients were

Table 3.1 Selected features for classification

Dataset	# of features	Selected features
UAH [147]	17	Speed (F1), Altitude (F2), Course (F3), Difcourse (F4), Acceleration in X (F5), Acceleration in Y (F6), Acceleration in Z (F7), Acceleration in X filtered by KF (F8), Acceleration in Y filtered by KF (F9), Acceleration in Z filtered by KF (F10), Roll angle (F11), Pitch angle (F12), Yaw angle (F13), Distance to ahead vehicle (F14), Time of impact (F15), X (F16), Phi (F17)
HCRL [148]	15	Long_Term_Fuel_Trim_Bank1 (F1), Intake_air_pressure (F2), Accelerator_Pedal_value (F3), Fuel_consumption (F4), Torque_of_friction (F5), Maximum_indicated_engine_torque (F6), Engine_torque (F7), Calculated_LOAD_value (F8), Activation_of_Air_compressor (F9), Engine_coolant_temperature (F10), Transmission_oil_temperature (F11), Wheel_velocity_front_left-hand (F12), Wheel_velocity_front_right-hand (F13), Wheel_velocity_rear_left-hand (F14), Torque_converter_speed (F15)
MotionSense [28]	12	attitude.roll (F1), attitude.pitch (F2), attitude.yaw (F3), userAcceleration.x (F4), userAcceleration.y (F5), userAcceleration.z (F6), gravity.x (F7), gravity.y (F8), gravity.z (F9), rotationRate.x (F10), rotationRate.y (F11), rotationRate.z (F12)
PAMAP2 [149]	31	heart rate (F1), hand temperature (F2), hand acceleration X (F3), hand acceleration Y (F4), hand acceleration Z (F5), hand gyroscope X (F6), hand gyroscope Y (F7), hand gyroscope Z (F8), hand magnetometer X (F9), hand magnetometer Y (F10), hand magnetometer Z (F11), chest temperature (F12), chest acceleration X (F13), chest acceleration Y (F14), chest acceleration Z (F15), chest gyroscope X (F16), chest gyroscope Y (F17), chest gyroscope Z (F18), chest magnetometer X (F19), chest magnetometer Y (F20), chest magnetometer Z (F21), ankle temperature (F22), ankle acceleration X (F23), ankle acceleration Y (F24), ankle acceleration Z (F25), ankle gyroscope X (F26), ankle gyroscope Y (F27), ankle gyroscope Z (F28), ankle magnetometer X (F29), ankle magnetometer Y (F30), ankle magnetometer Z (F31)

Table 3.2 Classification results for the initial set of features $\mathcal{F}$.

Dataset	# of features in $\mathcal{F}$ (initial set of features)	Classification objective	Classification accuracy		
			FCN-LSTM	SVM	XGBoost
UAH [147]	17	Driver (PBC)	0.9156	0.7435	0.9770
		Behavior (UFC)	0.9462	0.8457	0.9785
		Road (UFC)	0.9999	0.9993	0.9999
HCRL [148]	15	Driver (PBC)	0.9130	0.7733	0.9288
		Road (UFC)	0.9636	0.9281	0.9681
MotionSense [28]	12	User (PBC)	0.8592	0.7066	0.7782
		Activity (UFC)	0.9154	0.8359	0.8366
PAMAP2 [149]	31	User (PBC)	0.9980	0.9997	1.0000
		Activity (UFC)	0.9653	0.9678	0.9973

computed for the *driver* on the privacy side, and, for *behavior* and *road type*, for the utility side. Vectors containing the importance coefficients for each feature (I_u , I_p) were the output of this processing stage.

Table 3.3 Classification results for the selected set of features using FIG-based clustering and K-means clustering.

Clustering method	Dataset	$ \mathcal{F} $	Utility classification objective(s)	$ \mathcal{F}^* $	Classification	FCN-LSTM		SVM		XGBoost	
						$\mathcal{A}^*$	$\Delta\mathcal{A}$	$\mathcal{A}^*$	$\Delta\mathcal{A}$	$\mathcal{A}^*$	$\Delta\mathcal{A}$
FIG ($\rho_I = 0.0$)	UAH	17	Road	2	Driver (PBC)	0.4133	-0.5023	0.2388	-0.5047	0.4822	-0.4948
					Road (UFC)	0.9993	-0.0006	0.9974	-0.0019	0.9997	-0.0002
			Behavior	7	Driver (PBC)	0.6958	-0.2198	0.4632	-0.2803	0.7103	-0.2667
					Behavior (UFC)	0.8420	-0.1042	0.7280	-0.1177	0.8409	-0.1376
			Road and Behavior	9	Driver (PBC)	0.8756	-0.0400	0.5315	-0.2120	0.8382	-0.1388
					Behavior (UFC)	0.9171	-0.0291	0.7599	-0.0858	0.9023	-0.0762
					Road (UFC)	0.9996	-0.0003	0.9980	-0.0013	0.9998	-0.0001
	HCRL	15	Road	5	Driver (PBC)	0.2428	-0.6702	0.2029	-0.5704	0.2581	-0.6707
					Road (UFC)	0.9140	-0.0496	0.9038	-0.0243	0.9165	-0.0516
	MotionSense	12	Activity	6	User (PBC)	0.5987	-0.2605	0.4345	-0.2721	0.5588	-0.2194
					Activity (UFC)	0.7597	-0.1557	0.7056	-0.1303	0.7346	-0.1020
	PAMAP2	31	Activity	24	User (PBC)	0.9715	-0.0265	0.5187	-0.4810	0.9523	-0.0477
					Activity (UFC)	0.9651	-0.0002	0.5986	-0.3692	0.9513	-0.0460
K-means ($k = 2$)	UAH	17	Road	1	Driver (PBC)	0.2490	-0.6666	0.2321	-0.5114	0.2636	-0.7134
					Road (UFC)	0.9903	-0.0096	0.9900	-0.0093	0.9899	-0.0100
			Behavior	13	Driver (PBC)	0.6009	-0.3147	0.4594	-0.2841	0.5794	-0.3976
					Behavior (UFC)	0.7567	-0.1895	0.6640	-0.1817	0.7457	-0.2328
			Road and Behavior	14	Driver (PBC)	0.6566	-0.2590	0.5051	-0.2384	0.6990	-0.2780
					Behavior (UFC)	0.7809	-0.1653	0.6809	-0.1648	0.8038	-0.1747
					Road (UFC)	0.9978	-0.0021	0.9948	-0.0045	0.9994	-0.0005
	HCRL	15	Road	3	Driver (PBC)	0.1823	-0.7307	0.1687	-0.6046	0.1671	-0.7617
					Road (UFC)	0.8887	-0.0749	0.8850	-0.0431	0.8881	-0.0800

K-means ($k = 3$)	MotionSense	12	Activity	6	User (PBC)	0.5987	-0.2605	0.4345	-0.2721	0.5588	-0.2194
					Activity (UFC)	0.7597	-0.1557	0.7056	-0.1303	0.7346	-0.1020
	PAMAP2	31	Activity	28	User (PBC)	0.9701	-0.0271	0.9452	-0.0545	0.9523	-0.0477
					Activity (UFC)	0.9825	0.0172	0.9644	-0.0353	0.9513	-0.0460
	PAMAP2	31	Activity	3	User (PBC)	0.6560	-0.3420	0.5187	-0.4810	0.6462	-0.3538
					Activity (UFC)	0.7034	-0.2691	0.5986	-0.4011	0.6988	-0.2985

Further, features were clustered based on the rules outlined in Section 3.2.2, taking into account the computed feature importance coefficients determined in the preceding processing stage. Two clustering approaches were evaluated: one utilizing FIG values with $\rho_I = 0$, and the other employing K-means clustering with either $k = 2$ or $k = 3$. This selection process prioritized features most relevant for UFC, consequently omitting features important for PBC. The number of selected features is presented in Table 3.3, the fifth column ($|\mathcal{F}^*|$), for both methods. The classification is repeated with the same classifiers using the same testing conditions with the only difference being the number of features. Additionally, $\Delta\mathcal{A}$ was computed as the difference between the accuracy $\mathcal{A}$ computed with all initial features $\mathcal{F}$ and the accuracy $\mathcal{A}^*$ with the selected features $\mathcal{F}^*$. As observed in Table 3.3, the classification accuracy $\mathcal{A}^*$ is lower than $\mathcal{A}$ for most classification criteria.

According to the research goal (Equation 3.2), the objective was to discover a feature set $\mathcal{F}^*$ with significantly reduced classification accuracy for PBC compared to the original feature set, while ensuring that the accuracy of UFC remains at desired levels. These targeted accuracy levels are to be determined by the data owners.

The presented approach demonstrates excellent results on the UAH dataset by significantly reducing the number of features to be publicly released from 17 to 2 (for FIG-based clustering) and to 1 (for K-means clustering). Additionally, it led to a decrease in the accuracy of driver classification by approximately 0.50 for all tested classifiers (FCN-LSTM, SVM, and XGBoost) with FIG-based clustering, and by over 0.66 for K-means clustering. Despite these reductions, the accuracy of road-type identification remained consistently above 0.99. Therefore, it is sufficient to release only one feature to preserve user privacy while maintaining the utility of road-type detection. Comparable results were obtained for the HCRL dataset concerning driver/road-type classification.

Regarding the behavior detection utility objective, the approach successfully decreased the number of features from 17 to 7 (with FIG-based clustering) and 13 (with K-means clustering), while the accuracy of driver detection decreased by more than 0.21 across all tested scenarios. Notably, behavior detection was found to be higher for features selected by FIG-based clustering compared to those selected by K-means clustering.

For the MotionSense dataset, the features were reduced from 12 to 6 using both clustering methods, with the same set of selected features. This reduction led to a decrease in user identification classification accuracy to below 0.60. Concurrently, the utility accuracy also decreased to approximately 0.75.

The PAMAP2 dataset posed a challenge due to the remarkably high classification rate for the PBC (0.99), which is slightly higher than that of the UFC (0.96). By applying the FIG-based clustering method, there was a minimal reduction in the number of features. Similarly, the K-means method with $k = 2$ did not produce significant reductions. However, for $k = 3$, the number of selected features dramatically decreased to three, with the UFC achieving higher accuracy than the PBC. Nonetheless, both accuracies remained relatively low (0.65 – 0.70). Consequently, the developed method could offer limited applicability for

the PAMAP2 dataset, suggesting the need for alternative approaches for feature selection or introducing DP-based perturbation.

Furthermore, the case of multiple UFCs was tested for the UAH dataset with two utility classification objectives: behavior and road type detection simultaneously. In this case, the features were selected as described in Section 3.2.2. Significantly better results were obtained for the K-means clustering method, illustrating that in the case of multiple UFCs, the method of feature selection requires further research.

The experiments demonstrated the effectiveness of the methodology in significantly reducing the number of selected features while achieving the desired balance between privacy and utility in three of the four datasets. Despite inconclusive results for the PAMAP2 dataset, the methodology remains valuable for exploratory purposes.

3.4 Comparison to the state-of-the-art privacy-oriented feature selection approaches

In this chapter, feature selection has been examined as a mechanism to improve data privacy with the objective to complement classical methods such as data anonymization, transformation, and perturbation. Table 3.4 presents a comparison of related approaches and highlights the advantages of the developed method.

Zhang et al. [127] proposed a method that utilizes feature importance within a DP framework. This technique decreases data correlation and limits the sensitivity of feature relationships. It ensures that features contributing most to data correlation and sensitivity are minimized. However, the framework lacks a defined adversarial model, making it less effective against more sophisticated attacks. The absence of an explicit adversarial model limits its applicability in scenarios where high privacy goals are crucial, despite its utility for data with highly correlated features.

Jafer, Matwin, and Sokolova [131] introduced a method that includes an adversarial model to improve privacy during feature selection. This model applies the Privacy Breach Increase (PBI) score to assess potential breaches, ensuring that selected features do not increase data sensitivity. This method integrates privacy-aware considerations throughout the feature selection process and automatically identifies privacy-compromising features. Despite its strengths, the adversarial model assumes minimal access to sensitive data, which limits its robustness in cases of extensive adversarial knowledge.

Pattuk et al. [156] developed a privacy-aware dynamic feature selection technique that iteratively selects features based on the information needs of the targeted classifier. This approach requests minimal sensitive data during deployment. It aims to maximize privacy without sacrificing classification accuracy. However, its iterative nature may be computationally inefficient, particularly with large datasets or high-dimensional data.

The collaborative data privacy framework proposed by Sheikhalishahi and Martinelli [128] focuses on protecting data privacy in collaborative, distributed data-sharing environments.

Their framework proposes the elimination of irrelevant features to balance privacy and accuracy, specifically for collaborative classification tasks. The approach does not include a specific adversarial model, relying instead on the privacy-utility trade-off score to measure protection effectiveness. Its unique application to distributed architectures offers a novel contribution but may be limited in single-source or centralized data scenarios.

Srijayanthi and Sethukarasi [129] extend privacy-preserving feature selection by using clustering and anonymization techniques to identify and retain only the most relevant features. This method includes robust handling of Quasi-Identifiers (QIs) to mitigate identity, membership, attribute, and similarity attacks. By reducing data dimensionality and incorporating anonymization, it provides enhanced privacy while maintaining utility for classification. This approach, however, assumes the existence of QIs and may be less effective for data without clearly identifiable attributes.

Previous research examined privacy-focused methodologies for feature selection, ranging from differential privacy techniques to automated selection frameworks. This chapter presents a novel dual-model approach that incorporates adversarial threats capable of compromising user privacy. In contrast to previous works that either prioritize utility or apply privacy measures without considering the risk posed by privacy-compromising classifiers, the approach utilizes two opposing classifiers: one that optimizes utility by selecting features essential for classification accuracy and another that removes features that attackers could exploit. By integrating both privacy and utility as components of the feature selection process, the presented method offers a more balanced approach. A similar method was proposed by Rougé et al. [157], but the method lacks the flexibility to adjust the balance between privacy and utility according to specific needs.

Table 3.4 Comparison to the state-of-the-art privacy-oriented feature selection approaches

Research	Feature selection approach	Applicability	Adversarial model	Strengths
Zhang et al. [127]	DP feature selection based on feature importance	For highly correlated data	NA	Decreases data correlation
Sheikhalishahi et al. [128]	Selection based on the privacy-utility trade-off score	For collaborative data classification	NA	Applicable in a distributed data sharing architecture
Srijayanthi et al. [129]	Selection based on SU and the Utility Preserved Anonymization (UPA) algorithm	For data with QI attributes	Identity Disclosure (ID) attack, Membership Disclosure (MD) attack, Attribute Disclosure (AD) attack, Similarity Attack	Reduces data dimensionality and applies anonymization
Jafer et al. [131]	Selection based on the PBI score	For data with QI attributes	Privacy-breaking classifier with minimal information on sensitive data	Provides automatic feature selection, incorporates privacy considerations during the feature selection process
Pattuk et al. [156]	Dynamic iterative approach	For various data types	Information disclosure attacks	Dynamically selects the features based on the targeted classifier
Rougé et al. [157]	Feature selection based on feature importance	For various time series data	Privacy-breaking classifier	Balances privacy and utility
FS-BUP (presented in this chapter)	Feature clustering based on feature importance	For various time series data, when no data dimensionality reduction or data alteration are permitted	Powerful privacy-breaking classifier with extended data availability	Balances privacy and utility, provides automatic feature selection, incorporates privacy and utility objectives during the feature selection process

3.5 Summary

Maintaining a balance between privacy and data utility is a crucial aspect of creating and implementing techniques that preserve privacy while processing multivariate time series data. While existing privacy-preserving techniques, such as DP, are suitable in aggregate data releasing, their applicability to time series data release is limited, especially in contexts where the precision of the data values is critical. Moreover, the problem of protecting data in the context of TSC is even more challenging.

The current chapter presents FS-BUP, a novel privacy-oriented feature selection approach targeted for multivariate time series data in the context of TSC. By defining the problem in the context of two opposing classifiers (UFC and PBC) and by introducing the FIG score, the methodology identifies and selects features that are important for user-specific classification while minimizing their applicability to potential adversaries. Moreover, the developed approach is configurable, allowing the data owners to achieve a balance between privacy preservation and data utility. The technique is particularly valuable in scenarios where altering data to protect privacy is not possible due to the necessity of preserving data precision.

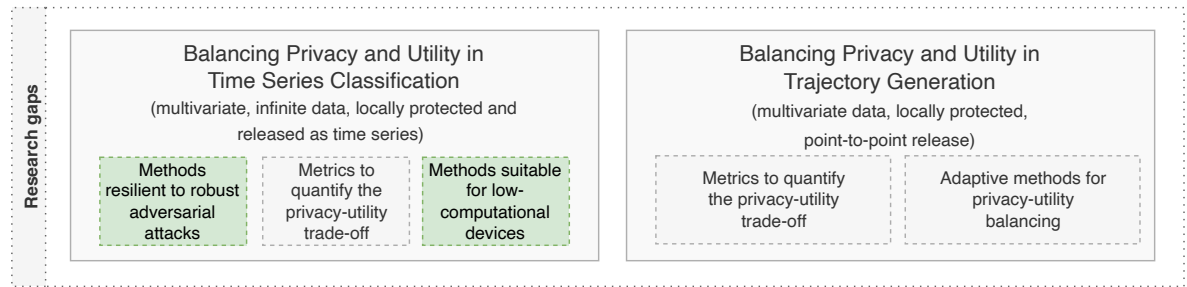


Figure 3.8 The research gaps targeted by the FS-BUP method (highlighted in green).

The presented methodology was evaluated in diverse scenarios, including driver and HAR datasets. Experimental results demonstrated that the presented approach effectively balances privacy and utility for certain datasets within the context of TSC. Nonetheless, for other datasets, the developed feature selection proved not to be sufficient, and other protection methods should be implemented. However, even for these datasets, the approach is considered a useful data exploratory step for better understanding the data and, equally important, the capabilities of potential attackers.

In the context of the identified research gaps (Figure 2.4), the FS-BUP method addresses (Figure 3.8) the need for resilience against robust adversarial attacks and the development of methods that can be implemented at the device level with minimal computational requirements.

Chapter 4

Utility-Preserving Data Perturbation for Multivariate Time Series Data Privacy

This chapter is based on the work presented in the following articles:

- **Adrian-Silviu Roman**, Béla Genge, and Piroška Haller. “Balancing Privacy and Utility in Multivariate Time-Series Classification”. In: *Computer Security. ESORICS 2024 International Workshops. Springer (2025). 19th International Workshop on Data Privacy Management (DPM 2024)*, 2024. [25]
- **Adrian-Silviu Roman**, Béla Genge, and Roland Bolboacă. “Privacy-Oriented Feature Selection for Multivariate Time Series Classification”. In: *Procedia Computer Science 246 (2024). 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024)*, 2024. [37]
- **Adrian-Silviu Roman**, Béla Genge, Adrian-Vasile Duka, and Piroška Haller. “Privacy-preserving tampering detection in automotive systems”. In: *Electronics 10.24*, 2021. [38]
- **Adrian-Silviu Roman**. “Evaluating the Privacy and Utility of Time-Series Data Perturbation Algorithms”. In: *Mathematics 11.5*, p. 1260, 2023. [39]

This chapter presents a data privacy mechanism through perturbation for multivariate time series data collected from sensors. Building on the context established in the previous chapter, it evaluates the accuracy of two opposing supervised classification models, the Privacy-Breaking Classifier (PBC) and Utility-Focused Classifier (UFC). The privacy and utility classifiers are analyzed to quantify their abilities to compromise privacy and preserve data utility in the context of protected time series data.

The Feature Selection for Balancing Utility and Privacy in Multivariate Time Series (FS-BUP) method, presented in the previous chapter, addresses two identified research gaps (Figure 2.4): the development of privacy-preserving techniques that are both resilient to robust adversarial attacks and compatible with low-computation environments. The Data Perturbation for Balancing Utility and Privacy in Multivariate Time Series (DP-BUP) mechanism, presented in this chapter, focuses on achieving the desired privacy-utility balance as well as higher resilience to adversarial attacks. It complements the FS-BUP approach as it increases the protection of remaining features and provides an alternative when reducing the number of features is not possible. This mechanism is configurable to achieve the desired balance between privacy and utility, and it takes into consideration the challenges documented in Section 2.2.1.2. In summary, the research presented in this chapter contributes to the research field in several key areas:

- It revisits the problem of balancing privacy and utility in the context of multivariate Time Series Classification (TSC), using a dual-model consisting of two opposing classifiers, a PBC and a UFC, presented in Chapter 3.
- It introduces DP-BUP, a data privacy technique independent of the perturbation type, applying the perturbation to multivariate time series data and utilizing feature importance to distribute the noise.
- It introduces the Classification Utility-Privacy Balance (BUP) score, denoted $\mathcal{B}_{UP}$, and a methodology for determining the appropriate value of the applied perturbation, to achieve the desired level of utility and privacy. Additionally, a method for parameter optimisation is presented and tested.
- It demonstrates the effectiveness and viability of the presented methodology by implementing a *w-event* Local Differential Privacy (LDP) perturbation and a Discrete Fourier Transform (DFT)-based perturbation on two well-known driver datasets [148, 147] and two Human Activity Recognition (HAR) datasets [28, 149]. Moreover, the results are compared with the outcomes of uniformly applying the same perturbation across all features.

4.1 Problem statement and research gaps

While methods to safeguard data collected from sensors during transmission to third-party systems [31, 10] exist, ensuring data privacy poses additional challenges. When applying data privacy mechanisms, monitoring and retaining data usefulness is essential to ensure that enough information is preserved for effective analysis. Additionally, implementing perturbation or other forms of privacy protection does not provide foolproof privacy guarantees. While Differential Privacy (DP) demonstrates effectiveness in aggregate data publishing scenarios, its applicability to time series data is limited. In the context of TSC, these limitations are more noticeable, as powerful classifiers are capable of identifying or re-identifying a user despite privacy measures.

Consider data collected from sensors, temporarily stored at the device level, and sent as data streams to a central data warehouse for classification. The objective is to protect the multivariate time series data from a privacy-breaking adversary classifier while preserving data utility. Furthermore, the restriction that no dimensionality reduction methods are permitted is imposed, ensuring that the full set of features is maintained.

Similar to the method introduced in Chapter 3, two distinct types of classifiers are evaluated: (i) a UFC, for identifying non-user related data patterns, and (ii) a PBC, designed to determine the identity of the users. The objective of this research is to develop a protection mechanism at the device level (see Figure 3.1), such that the UFC provides suitable detection accuracy while the accuracy of the PBC is as low as possible.

Use case: Driving behavior analysis for usage-based insurance. Consider a fleet of company-owned automotive vehicles that send data collected from sensors to an insurance company implementing the “pay-how-to-drive” scheme [158]. Data is collected locally on the vehicles and periodically sent to specified servers over the Internet. Within the presented context, consider two primary objectives of the insurance company: (i) to categorize the driving behavior, distinguishing between aggressive and normal driving styles; (ii) to categorize the type of road surfaces used by the vehicles. At the same time, the objective of the company that owns the auto vehicles is to extract statistical information (e.g., average speed) from collected data, thus requiring access to all available data features. The objectives do not break the privacy of the users as there is no personal information or identification involved. Thus, the company managing the vehicle fleet faces the challenge of minimizing the privacy risk of driver identification through the data collected and transmitted to the insurance company. Furthermore, the car owner must ensure that the data remains protected from potential attackers who might have access to both the collected data and the identities of the drivers. These attackers could potentially construct a PBC model capable of identifying a driver based on intercepted data. The presented method, DP-BUP, protects the collected data, at the device level, such that the classifications needed by the insurance company are feasible, while the accuracy of the PBC is as low as possible.

4.2 The DP-BUP method

Time series data, continuously collected from sensors, is perceived as infinite data streams, with each distinct value representing an event generated by the user (such as an event indicating “the driver u traveled at speed s at time t ”). Further, the presented approach is introduced to protect the sensitive data encapsulated in these events.

The DP-BUP method was developed, a time series perturbation method suitable for multivariate time series data that outperforms the uniform distribution of perturbation across all features. The approach is designed to be independent of the specific perturbation method used, ensuring robustness and flexibility in handling diverse types of time series data. Additionally, the method is developed for locally applied perturbation at the device level.

Let the continually collected data, consisting of records with d features, be stored locally on the device and then transmitted in groups of w records to a central location for processing. Thus, let $\mathbf{X}_i^t$ be the time series data collected for a period of time and stored locally at time t , on the sensor-based device i . $\mathbf{X}_i^t$ is represented as a matrix of w rows and d columns. Each column denotes measurements associated with a particular attribute/feature.

TSC refers to the task of assigning a label or category to a given time series data based on its trends, patterns, or features. A classifier model produces a function $f : \mathbb{R}^{w \cdot d} \rightarrow \mathcal{C}$, $\mathcal{C} = \{1, \dots, n\}$, where w is the number of records, d is the number of features and n the number of categories, that is applied to any input $\mathbf{X}$. The function $f(\mathbf{X})$ is regarded as an estimate of the category that $\mathbf{X}$ belongs to [106]. Let f_p and f_u be two classification functions, $f_p : \mathbb{R}^{w \cdot d} \rightarrow \mathcal{C}_u$ and $f_u : \mathbb{R}^{w \cdot d} \rightarrow \mathcal{C}_p$ that output the class that $\mathbf{X}_i^t$ belongs to, in case of the PBC, and UFC, respectively. $f_p(\mathbf{X}_i^t)$ is the PBC classifier that breaks the user privacy by identifying the set of collected values as belonging to a user from a specified class $\mathcal{C}_p$ and $f_u(\mathbf{X}_i^t)$, the UFC, classifies the non-sensitive information from a class $\mathcal{C}_u$.

In the training and testing phases of building the UFCs, for each $\mathbf{X}_i^t$ an ID of the device or/and of the user is also collected and stored, with the objective of labeling the data. Let $\mathbf{Y}_{pi}^t$ be the label information for the user identification, and let $\mathbf{Y}_{ui}^t$ be the label for non-user-related data classification. In the prediction phases, as a fundamental measure to preserve privacy, no data that could instantly identify individuals or devices is transmitted to the central data warehouse.

The objective is to perturb time series data $\mathbf{X}_i^t$ with a mechanism $\mathcal{M}$ such that two conditions are met: the classification accuracy of f_p on the perturbed data is minimum, while simultaneously maximizing the classification accuracy of f_u on the perturbed data.

Let the classification accuracy of the two classifiers, f_u and f_p , be denoted as $\mathcal{A}_u$ and $\mathcal{A}_p$, respectively, as defined in Equation 3.1 (Section 3.2). Similarly, let the classification accuracy for the perturbed data be defined as:

$$\mathcal{A}'_u(\theta) = \frac{1}{N \cdot T} \sum_{i=1}^N \sum_{t=1}^T \mathbb{I}(\mathbf{Y}_{ui}^t = f_u(\mathcal{M}(\mathbf{X}_i^t; \theta))), \text{ and} \quad (4.1)$$

$$\mathcal{A}'_p(\theta) = \frac{1}{N \cdot T} \sum_{i=1}^N \sum_{t=1}^T \mathbb{I}(\mathbf{Y}_{pi}^t = f_p(\mathcal{M}(\mathbf{X}_i^t; \theta))), \quad (4.2)$$

where $\mathcal{M}$ is the perturbation mechanism, $\mathcal{M} : \mathbb{R}^{w \cdot d} \rightarrow \mathbb{R}^{w \cdot d}$, and θ is the perturbation parameter set (e.g., ϵ for LDP).

Thus, the data perturbation objective is bounded by the following conditions:

$$\mathcal{A}'_p(\theta) \ll \mathcal{A}_p \text{ and } \mathcal{A}'_u(\theta) \approx \mathcal{A}_u, \quad (4.3)$$

such that the accuracy of PBC is significantly diminished when data is perturbed, while the accuracy of the UFC for perturbed data remains close to the accuracy of the classification on non-perturbed data.

Achieving a balance between privacy and utility presents a significant challenge, particularly when considering classification tasks involving two opposing classifiers. $\mathcal{B}_{UP}$, the BUP score, is introduced as a measure for balancing privacy and utility in the presented classification problem, computed as follows:

$$\mathcal{B}_{UP}(\theta) = 1 - \frac{\mathcal{A}'_u(\theta)}{\mathcal{A}_u} \cdot \left(1 - \frac{\mathcal{A}'_p(\theta)}{\mathcal{A}_p} \right). \quad (4.4)$$

The score $\mathcal{B}_{UP}$ is positive and close to zero when the perturbation successfully meets the specified objectives from Equation 4.3. This indicates that the accuracy of the UFC with perturbed data is close to its accuracy with unperturbed data, while the accuracy of the PBC with perturbed data is significantly lower than its accuracy with unperturbed data. A negative $\mathcal{B}_{UP}$ value signifies that the chosen perturbation has increased the accuracy of the PBC, representing the worst-case scenario. In the context of evaluating the performance of the presented approach in terms of classification accuracy for UFC and PBC, the score $\mathcal{B}_{up}$ highlights the most significant difference between UFC accuracy and PBC accuracy. In conclusion, $\mathcal{B}_{up}$ was introduced as a tool to facilitate the identification of appropriate perturbation parameters.

The introduced method for finding the perturbation parameters consists of the following steps (Figure 4.1):

- (i) Compute feature importance for the two classifications (UFC and PBC).
- (ii) Cluster features based on the computed importance coefficients, with the method presented in Chapter 3.
- (iii) Distribute and apply the perturbation to the features of the time series data $\mathbf{X}_i^t$.
- (iv) Select the perturbation parameter set θ^* such that $\mathcal{B}_{UP}(\theta)$ is minimum:

$$\theta^* = \operatorname{argmin}_{\theta} \{ \mathcal{B}_{UP}(\theta) | \mathcal{B}_{UP}(\theta) > 0 \}. \quad (4.5)$$

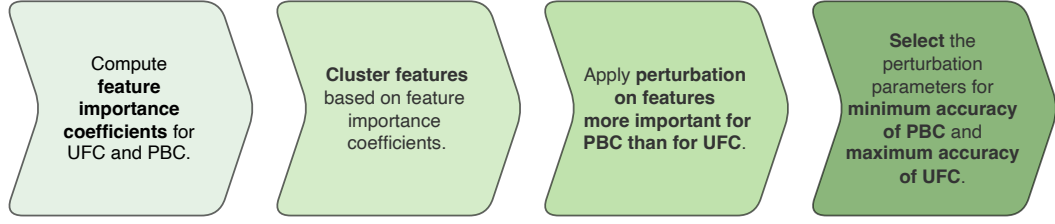


Figure 4.1 Steps of the DP-BUP method.

4.2.1 Feature clustering for data perturbation

Let $\mathcal{F}$ be the set of available features from the considered dataset, $\mathcal{F} = \{F_1, \dots, F_i, \dots\}$, with the objective to cluster $\mathcal{F}$ into three subsets: $\mathcal{F}_u$ with features necessary for the utility-focused classification, $\mathcal{F}_p$ with features that play a significant role in resolving the privacy-breaking classification, and $\mathcal{F}_{up}$ with features that are important for both classifications, such that $\mathcal{F} = \mathcal{F}_u \cup \mathcal{F}_p \cup \mathcal{F}_{up}$ (Figure 4.2).

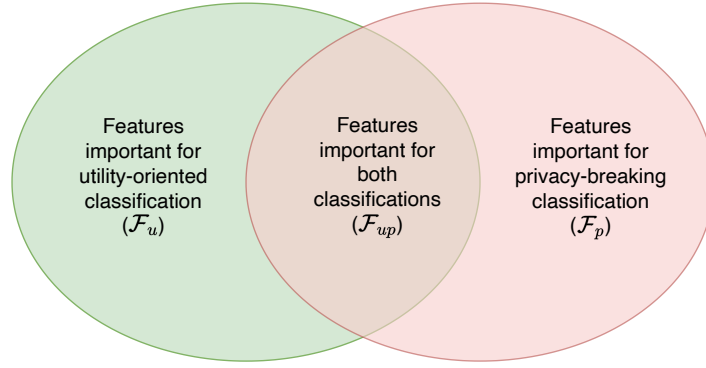


Figure 4.2 Feature clustering objective.

Consider the importance-based feature clustering method described in Section 3.2.2, which involves computing the Feature Importance Gap (FIG) score, $\mathcal{S}_{F_i}$. However, in the case of clustering the features into three groups, consider the following rules:

$$F_i \in \mathcal{F}_u, \text{ if } \mathcal{S}_{F_i}(i_i^u, i_i^p) > \rho_I, \quad (4.6)$$

$$F_i \in \mathcal{F}_p, \text{ if } \mathcal{S}_{F_i}(i_i^u, i_i^p) < -\rho_I, \quad (4.7)$$

$$F_i \in \mathcal{F}_{up}, \text{ if } \mathcal{S}_{F_i}(i_i^u, i_i^p) \in [-\rho_I, \rho_I], \quad (4.8)$$

where ρ_I is the decision boundary. The score $\mathcal{S}_{F_i}(i_i^u, i_i^p)$ determines when a feature shows a significantly greater significance in a classification context compared to another. When $\mathcal{S}_{F_i}(i_i^u, i_i^p) = 0$, only the features that have the same importance coefficients for both UFC and PBC belong to $\mathcal{F}_{up}$. Figure 4.3 exemplifies a set of ten features, their feature importance coefficients for a UFC and a PBC, and a possible clustering of the features based on rules 4.6, 4.7, and 4.8.

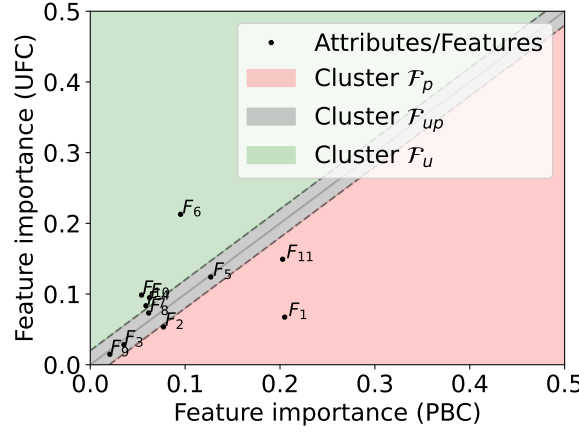


Figure 4.3 Example of feature clustering based on feature importance coefficients for two classifications (UFC and PBC).

4.2.2 Data privacy using data perturbation

The developed mechanism considers a global perturbation budget to be divided within the features in $\mathcal{F}$. To achieve the goal of reducing the PBC's impact while preserving the accuracy of the UFC, substantial perturbation is introduced to features within $\mathcal{F}_p$, moderate perturbation to features within $\mathcal{F}_{up}$, while maintaining the features within the cluster $\mathcal{F}_u$ unmodified.

Consider the mechanism $\mathcal{M}$ of perturbing $\mathbf{X}_i^t$ a composition of d mechanisms $\mathcal{M}_i$, one for each feature F_i in clusters $\mathcal{F}_p$, $\mathcal{F}_{up}$ and $\mathcal{F}_u$. Let the total perturbation budget β_T be allocated to $\mathcal{M}$ and distributed to mechanisms $\mathcal{M}_i$, based on the membership in clusters $\mathcal{F}_p$, $\mathcal{F}_{up}$ and $\mathcal{F}_u$. Let β_p , β_{up} , and β_u be the cumulative privacy budgets for features in clusters $\mathcal{F}_p$, $\mathcal{F}_{up}$ and $\mathcal{F}_u$, respectively, such that:

$$\beta_p = \alpha_p \cdot \beta_T, \beta_{up} = \alpha_{up} \cdot \beta_T, \beta_u = 0, \text{ and} \quad (4.9)$$

$$\alpha_p + \alpha_{up} = 1, \text{ with } \alpha_p, \alpha_{up} \in [0, 1], \quad (4.10)$$

where α_p and α_{up} are the perturbation budget distribution parameters. Within each cluster, budgets are uniformly distributed such that:

$$\beta_{pi} = \frac{\alpha_p \cdot \beta_T}{|\mathcal{F}_p|}, \text{ and } \beta_{upi} = \frac{\alpha_{up} \cdot \beta_T}{|\mathcal{F}_{up}|}, \quad (4.11)$$

where β_{pi} is the privacy budget allocated for each $F_i \in \mathcal{F}_p$, β_{upi} the budget allocated for each $F_i \in \mathcal{F}_{up}$, and $|\mathcal{F}_p|$, $|\mathcal{F}_{up}|$ the number of features in clusters $\mathcal{F}_p$ and $\mathcal{F}_{up}$. Additionally, the imposed requirement is that the perturbation applied to features in $\mathcal{F}_p$ be higher or equal to the one applied to features in $\mathcal{F}_{up}$. Thus, the sensitive information related to features that are important for PBC is more perturbed than in features that are important for both PBC and

UFC. The goal is to lower the accuracy of PBC to achieve data privacy while preserving the utility of the data by maintaining the accuracy of UFC at specified levels. When the amount of perturbation applied to data is proportional to the allocated budget β_T , meaning that the higher the budget, the higher the perturbation, we enforce that $\beta_{pi} \geq \beta_{upi}$. Otherwise, when a lower privacy budget inflicts higher perturbation (as is the case in DP), we consider $\beta_{pi} \leq \beta_{upi}$.

4.2.2.1 Data Perturbation using w-event-level LDP

The first perturbation method investigated in the proposed context involves implementing the state-of-the-art w -event level LDP (described in Section 2.2.1.2) with LDP Budget Uniform Method (LBU) (Section 2.2.1.2) to features in clusters $\mathcal{F}_p$ and $\mathcal{F}_{up}$. The DP-based perturbation provides rigorous noise distribution and ensures the protection of individual data values. Figure 4.4 shows the results of applying the w -event-level LDP perturbation on univariate time series data representing the speed of a vehicle.

The selection of perturbation budget parameters α_p and α_{up} needs to account for the number of features in the cluster. Proposition 1 provides the formal definition for computing the privacy budget ϵ allocated to each feature i . This computation ensures that the overall privacy guarantees adhere to the principles of DP while balancing the contributions of individual features to utility and privacy.

The objective is to identify ρ_I (the decision boundary used in Equations 4.6, 4.7, and 4.8), along with β_T , α_p , α_{up} , to ensure that $\mathcal{B}_{UP}(\theta)$, with $\theta = \{\rho_I, \beta_T, \alpha_p, \alpha_{up}\}$, is minimized while remaining positive. The experiments (Section 4.3) confirm that suitable values for the perturbation parameters can be determined and that the $\mathcal{B}_{UP}$ score effectively indicates the appropriate size of the perturbation.

Proposition 1. *Let $\mathcal{M}_{LDP}$ be a mechanism composed of m mechanisms $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_i, \dots, \mathcal{M}_m$, $m < d$, one for each attribute/feature F_i ($F_i \in \mathcal{F}_p$ or $F_i \in \mathcal{F}_{up}$), satisfying ϵ_i -LDP, such that the privacy budget for each mechanism $\mathcal{M}_i$ is defined as follows:*

$$\epsilon_i = \begin{cases} \frac{\alpha_p \cdot \beta_T}{|\mathcal{F}_p|}, & \text{if } F_i \in \mathcal{F}_p, \\ \frac{\alpha_{up} \cdot \beta_T}{|\mathcal{F}_{up}|}, & \text{if } F_i \in \mathcal{F}_{up}, \end{cases}$$

where $\alpha_p + \alpha_{up} = 1$, with $\alpha_p, \alpha_{up} \in [0, 1]$. If the following condition is fulfilled:

$$\frac{\alpha_p}{|\mathcal{F}_p|} \leq \frac{\alpha_{up}}{|\mathcal{F}_{up}|}. \quad (4.12)$$

then the perturbation of features in $\mathcal{F}_p$ is higher than or equal to the perturbation applied on features in $\mathcal{F}_{up}$.

Proof. Let ϵ_{pi} be the LDP privacy budget applied on features in cluster $\mathcal{F}_p$ and ϵ_{upi} the LDP privacy budget applied on features in cluster $\mathcal{F}_{up}$. Then,

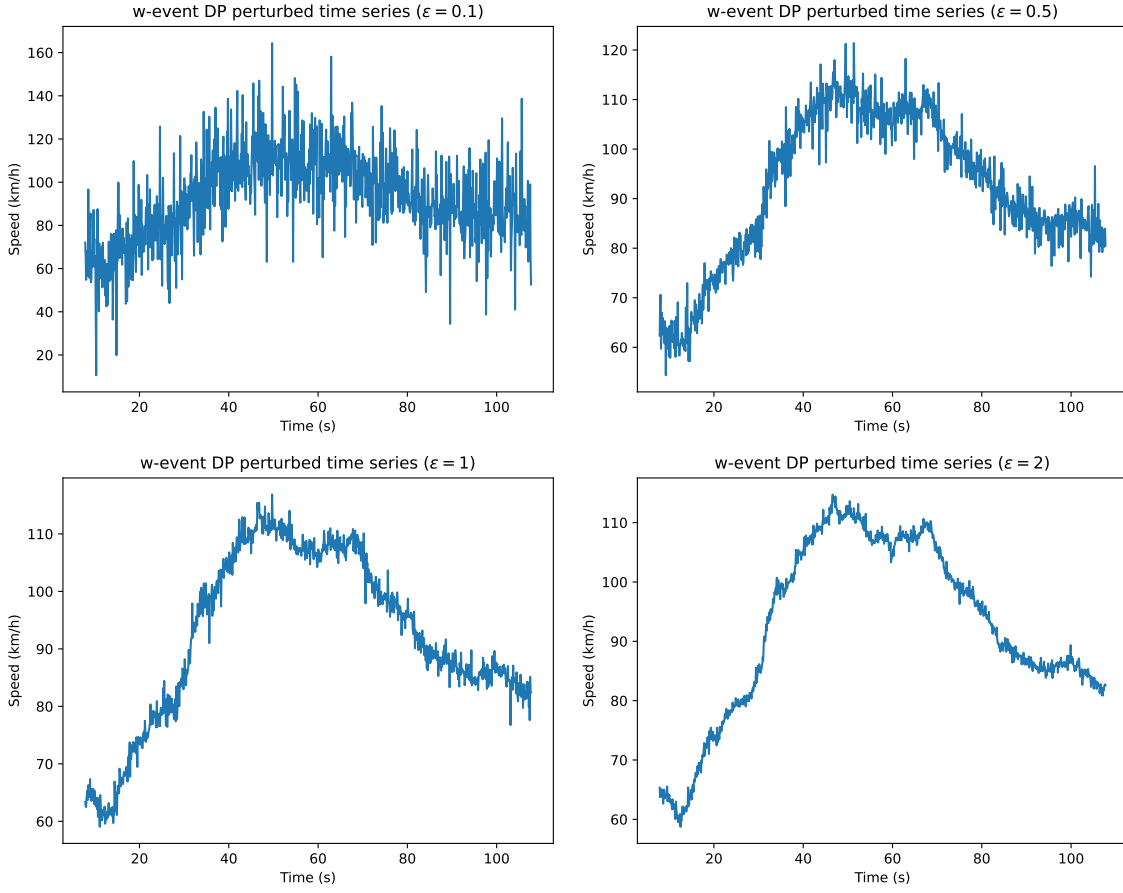


Figure 4.4 Example of w-event LDP perturbed speed data using various ϵ values.

$$\frac{\alpha_p}{|\mathcal{F}_p|} \leq \frac{\alpha_{up}}{|\mathcal{F}_{up}|} \Leftrightarrow \frac{\alpha_p \cdot \beta_T}{|\mathcal{F}_p|} \leq \frac{\alpha_{up} \cdot \beta_T}{|\mathcal{F}_{up}|} \Leftrightarrow \epsilon_{pi} \leq \epsilon_{upi}. \quad (4.13)$$

Taking into account that for LDP, “the smaller the ϵ , the higher the perturbation” [76], then the perturbation for features in cluster $\mathcal{F}_p$ is higher than the perturbation for features in $\mathcal{F}_{up}$. $\square$

4.2.2.2 DFT-based data perturbation

The second type of protection method considered is a transformation-based perturbation. Consider the DFT data distortion presented in Section 2.2.1.3. In addition to filtering, other data transformations are possible. These can be applied in the frequency domain either before or after the filtering. In this context, the choice is to use a Gaussian white noise with a given variance in the frequency domain after filtering to enhance the distortion of the resulting data and ensure data privacy. The main advantages of this type of perturbation are that smaller noise is necessary to protect the data than in the case when the perturbation is applied to the original data and the fact that noise is not directly applied to the time series data but to the transformed values, reducing the possible success of the filtering attacks [30].

The selected method [29] defines the *discord* σ as the standard deviation of the added perturbation. The *discord* σ determines the maximum information loss that the system is willing to tolerate and the maximum uncertainty that can be introduced. By design, the perturbation technique of adding Gaussian white noise with variance σ^2 preserves the signal's spectral and “smoothness” properties [29]. Thus, the approach preserves both the privacy and the utility of time series.

To increase resistance to filtering and other types of attacks, we proposed [38] adding the noise to only a fraction of the frequencies such that the total noise variance is maintained at σ^2 . In this case, the standard deviation of the Gaussian noise applied to the values in the frequency domain becomes $\sigma\sqrt{N/k}$.

Proposition 2. *A standard deviation of $\sigma\sqrt{N/k}$ for k out of N frequencies maintains the overall variance of the Gaussian white noise at σ^2 .*

Proof. Denote with n_i the Gaussian noise to be applied to frequency values. The variance $Var[n_i]$ is by definition:

$$Var[n_i] = \frac{1}{N} \sum_{i \in I} (n_i - \mu)^2 = \frac{1}{N} \sum_{i \in I} n_i^2, \quad (4.14)$$

where N is the total number of non-filtered frequencies ($N = \text{sum}(\bar{F} > 0)$), μ is the mean of noise values, $\mu = 0$, and I is the set of indices of the non-filtered frequencies. Further, the noise applied to the selected frequencies ($i \in I'$ was separated, where I' is the set of indices of frequencies with the magnitude greater than σ) from the noise applied to the rest of the frequencies ($i \in I \setminus I'$):

$$Var[n_i] = \frac{1}{N} \sum_{i \in I} n_i^2 = \frac{1}{N} \left(\sum_{i \in I'} n_i^2 + \sum_{i \in I \setminus I'} n_i^2 \right). \quad (4.15)$$

The variance of the noise values greater than σ is:

$$\sigma^2 \frac{N}{K} = \frac{1}{k} \sum_{i \in I'} n_i^2, \text{ and } \sum_{i \in I'} n_i^2 = \sigma^2 N. \quad (4.16)$$

By replacing (4.16) into equation (4.15), and taking into account that n_i is zero for frequencies with magnitude less than σ , the result $Var[n_i] = \sigma^2$ is obtained:

$$Var[n_i] = \frac{1}{N} \left(\sigma^2 N + (N - k) \cdot 0 \right) = \sigma^2. \quad (4.17)$$

□

The concept of frequency filtering is adapted to retain a specific number of components rather than applying a filter that selects components based on whether they fall above or below a certain frequency threshold. Algorithm 2 outlines the procedures performed by the developed

Algorithm 2: DFT Perturbation Algorithm

Input: $\mathcal{X} = (X_1, \dots, X_N)$ (Time series data); k (Number of selected coefficients); σ (Noise variance);

Output: $\mathcal{X}' = (X'_1, \dots, X'_N)$ (Perturbed time series data)

Function DFT Perturbation Algorithm(X, k, σ):

Step 1: Compute the discrete Fourier transform of $\mathcal{X}$, $F = DFT(\mathcal{X})$.

Step 2: Select the first k coefficients of F , denoted by F^k .

Step 3: Add a Gaussian white noise $\mathcal{G}(\sigma\sqrt{N/k})$ to each F^k and obtain $\hat{F}^k$.

Step 4: Compute the inverse discrete Fourier transform of $\hat{F}^k$, $\mathcal{X}' = IDFT(\hat{F}^k)$.

return $\mathcal{X}'$

End Function

perturbation method. Figure 4.5 exemplifies the application of the DFT perturbation algorithm on univariate time series data representing the speed of a vehicle.

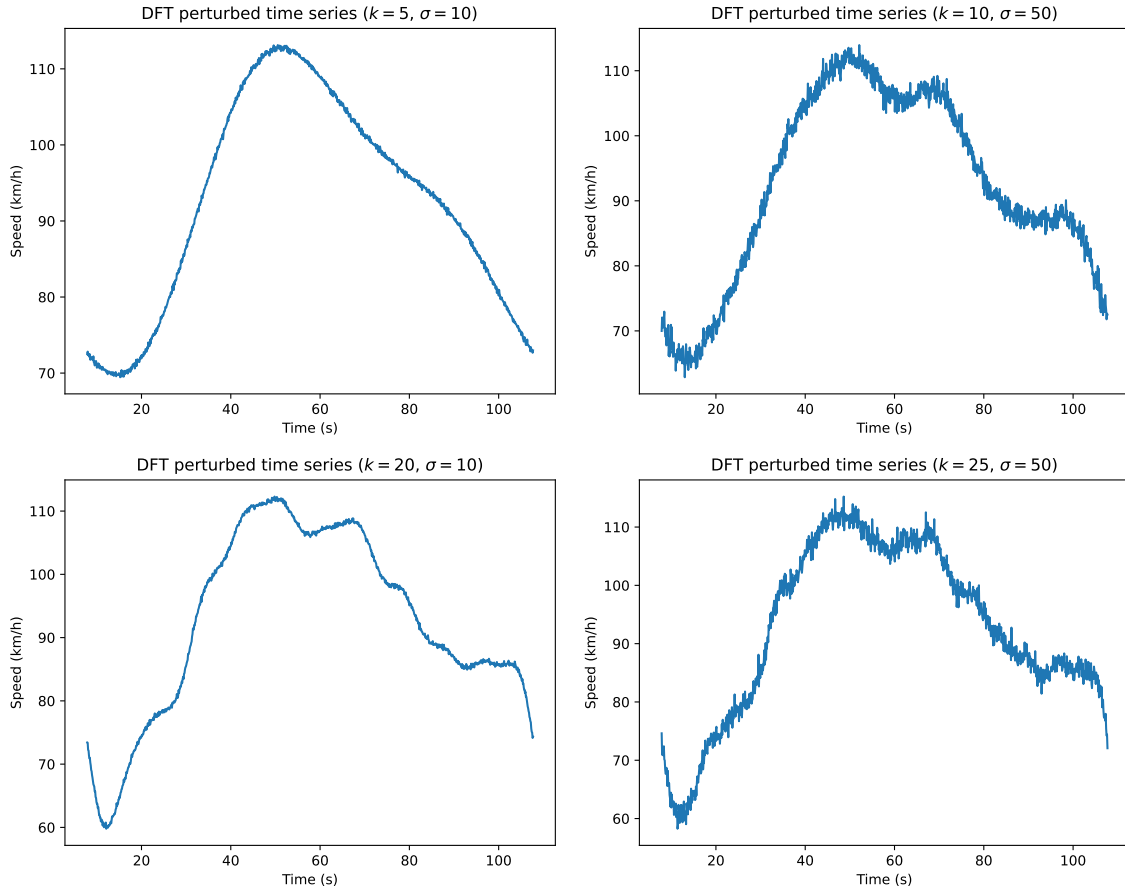


Figure 4.5 Example of DFT perturbed speed data using various k and σ values.

In the context of distributing the perturbation based on the feature importance, a mechanism $\mathcal{M}_{DFT}$ that applies the perturbation described in Algorithm 2 to features in clusters $\mathcal{F}_p$ and $\mathcal{F}_{up}$ is considered. For simplicity, let k fixed for all features to be perturbed.

The total privacy budget β_T represents the sum of noise (σ_i) to be applied to the selected features.

Proposition 3. *Let $\mathcal{M}_{DFT}$ be a mechanism composed of m mechanisms $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_i, \dots, \mathcal{M}_m$, $m < d$, one for each attribute/feature F_i ($F_i \in \mathcal{F}_p$ or $F_i \in \mathcal{F}_{up}$), such that for each mechanism applying a DFT-based perturbation the discord σ is computed as follows:*

$$\sigma_i = \begin{cases} \frac{\alpha_p \cdot \beta_T}{|\mathcal{F}_p|}, & \text{if } F_i \in \mathcal{F}_p, \\ \frac{\alpha_{up} \cdot \beta_T}{|\mathcal{F}_{up}|}, & \text{if } F_i \in \mathcal{F}_{up}, \end{cases}$$

where $\alpha_p + \alpha_{up} = 1$, with $\alpha_p, \alpha_{up} \in [0, 1]$. If the following condition is fulfilled:

$$\frac{\alpha_p}{|\mathcal{F}_p|} \geq \frac{\alpha_{up}}{|\mathcal{F}_{up}|}. \quad (4.18)$$

then the perturbation of features in $\mathcal{F}_p$ is higher than or equal to the perturbation applied on features in $\mathcal{F}_{up}$.

Proof. Let σ_{pi} be the noise size budget applied on features in cluster $\mathcal{F}_p$ and ϵ_{upi} the noise size applied on features in cluster $\mathcal{F}_{up}$. Then,

$$\frac{\alpha_p}{|\mathcal{F}_p|} \geq \frac{\alpha_{up}}{|\mathcal{F}_{up}|} \Leftrightarrow \frac{\alpha_p \cdot \beta_T}{|\mathcal{F}_p|} \geq \frac{\alpha_{up} \cdot \beta_T}{|\mathcal{F}_{up}|} \Leftrightarrow \sigma_{pi} \geq \sigma_{upi}. \quad (4.19)$$

Taking into account that for the Algorithm 2, the higher the σ , the larger the perturbation, then the perturbation for features in cluster $\mathcal{F}_p$ is higher than the perturbation for features in $\mathcal{F}_{up}$. $\square$

According to Proposition 3, the parameters α_p and α_{up} need to account for the number of features in the cluster. The challenge lies in identifying $\rho_I, \beta_T, \alpha_p, \alpha_{up}$, to ensure that $\mathcal{B}_{UP}(\theta)$, with $\theta = \{\rho_I, \beta_T, \alpha_p, \alpha_{up}\}$, is minimized while remaining positive.

4.2.3 Privacy and utility analysis

Current time series protection mechanisms are primarily designed to address basic queries, such as count, mean, and frequency [49], and in this context, data perturbation techniques, including DP, offer robust privacy guarantees. However, ensuring protection against more complex tasks like clustering or classification remains an open challenge.

The w -event LDP mechanism ensures data indistinguishability across batches of w records transmitted to a central server [159]. This mechanism makes it impossible to precisely identify a specific event or value within the transmitted batch, preventing an attacker from uniquely associating a user with the data obtained from a centralized repository. For instance, while an attacker might be able to observe a driver's speed s at a given time t in traffic, they cannot infer which specific records in the data warehouse belong to that driver based solely on the information they observe. LDP offers stronger protection than Centralized Differential Privacy

(CDP) by defending against adversaries who can intercept all communications involving a specific user. When a protocol satisfies LDP, it inherently fulfills the criteria for central differential privacy as well. However, without additional assumptions, stronger claims about CDP cannot be made. In fact, in centralized settings, a mechanism could potentially expose full transcripts of its interactions with users, including their identities, making LDP the only safeguard against data leaks through the mechanism's outputs [160].

The DFT-based perturbation mechanism provides no explicit guarantees in terms of value and user indistinguishability. In the context of classification tasks, mechanisms like w -event LDP and the introduced DFT-based method provide limited privacy guarantees due to the nature of the classification process itself. The presented classification scenario involves two competing models: a UFC and a PBC, each with opposing goals. The perturbation process must obscure sensitive attributes relevant to user privacy (to lower the PBC) while retaining the features necessary for accurate classification by the UFC. Despite these challenges, the presented approach ensures a satisfactory level of privacy by maximizing $\mathcal{P}_{PBC}(\epsilon)$ to limit the effectiveness of the PBC, and by keeping $\mathcal{A}'_p(\epsilon)$ below a specified threshold, ensuring privacy is maintained without significantly sacrificing utility.

4.2.4 Parameter optimisation

The BUP score, $\mathcal{B}_{UP}$, was defined with the goal of identifying the set of parameters θ that maximize the accuracy of the UFC, $\mathcal{A}'_u(\theta)$, while ensuring that the accuracy of the PBC aligns with the objectives specified in Equation 3.2.

Consider the problem of determining the parameter set θ that minimizes the objective function (as formulated in Equation 4.5), and let the objective function be:

$$\mathcal{B}_{UP}(\theta) = 1 - \frac{\mathcal{A}'_u(\theta)}{\mathcal{A}_u} \cdot \left(1 - \frac{\mathcal{A}'_p(\theta)}{\mathcal{A}_p}\right) \text{ with the constraint } \mathcal{B}_{UP} > 0, \quad (4.20)$$

where: $\theta = (\theta_1, \theta_2, \dots, \theta_n)$ is a set of parameters influencing both UFC and PBC, $\mathcal{A}'_u(\theta)$ and $\mathcal{A}'_p(\theta)$ are the classification accuracies, dependent of θ , and $\mathcal{A}_u$ and $\mathcal{A}_p$ are the baseline accuracies (the classification accuracies for non-perturbed data).

To optimize the objective function (Equation 4.20), a general gradient descent approach [161] is utilized. To apply the gradient descent method, the function $\mathcal{B}_{UP}(\theta)$ needs to be continuous and differentiable. Consider that $\mathcal{A}'_u(\theta)$ and $\mathcal{A}'_p(\theta)$ modify gradually with changes in θ , as empirically observed. Additionally, consider $\mathcal{B}_{UP}(\theta)$ differentiable with respect to θ , as $\mathcal{A}'_u(\theta)$ and $\mathcal{A}'_p(\theta)$ are considered smooth functions.

For each parameter θ_i , compute:

$$\frac{\partial \mathcal{B}_{UP}}{\partial \theta_i} = -\frac{1}{\mathcal{A}_u} \frac{\partial \mathcal{A}'_u(\theta)}{\partial \theta_i} + \frac{1}{\mathcal{A}_u \mathcal{A}_p} \left(\mathcal{A}'_u(\theta) \frac{\partial \mathcal{A}'_p(\theta)}{\partial \theta_i} + \mathcal{A}'_p(\theta) \frac{\partial \mathcal{A}'_u(\theta)}{\partial \theta_i} \right). \quad (4.21)$$

Further, the expression is simplified and obtain:

$$\frac{\partial \mathcal{B}_{UP}}{\partial \theta_i} = -\frac{1}{\mathcal{A}_u} \left(1 - \frac{\mathcal{A}'_p(\theta)}{\mathcal{A}_p} \right) \frac{\partial \mathcal{A}'_u(\theta)}{\partial \theta_i} + \frac{\mathcal{A}'_u(\theta)}{\mathcal{A}_u \mathcal{A}_p} \frac{\partial \mathcal{A}'_p(\theta)}{\partial \theta_i}. \quad (4.22)$$

The computed gradient vector is:

$$\nabla_{\theta} \mathcal{B}_{UP}(\theta) = \left[\frac{\partial \mathcal{B}_{UP}}{\partial \theta_1}, \frac{\partial \mathcal{B}_{UP}}{\partial \theta_2}, \dots, \frac{\partial \mathcal{B}_{UP}}{\partial \theta_n} \right]^T. \quad (4.23)$$

In practice, there are no guarantees that $\mathcal{A}'_u(\theta)$ and $\mathcal{A}'_p(\theta)$ are differentiable and smooth. To address this issue, a numerical approximation of the gradients for each parameter θ_i using finite differences [161] is employed:

$$\frac{\partial \mathcal{A}'_u(\theta)}{\partial \theta_i} \approx \frac{\mathcal{A}'_u(\theta + h\mathbf{e}_i) - \mathcal{A}'_u(\theta)}{h}, \quad (4.24)$$

$$\frac{\partial \mathcal{A}'_p(\theta)}{\partial \theta_i} \approx \frac{\mathcal{A}'_p(\theta + h\mathbf{e}_i) - \mathcal{A}'_p(\theta)}{h}, \quad (4.25)$$

where $\mathbf{e}$ is the unit vector and h the small value used to perturb the parameter θ_i , enabling the computation of the numerical estimate of the derivative.

Using these numerical gradients, the parameters are updated iteratively using the gradient descent rule:

$$\theta^{(k+1)} = \theta^{(k)} - \eta \nabla_{\theta} \mathcal{B}_{UP}(\theta^{(k)}), \quad (4.26)$$

where η is the learning rate and k the iteration number.

The empirical nature of accuracy computations presents a significant computational challenge, particularly when working with large datasets or complex models. Additionally, the selection of h in the numerical gradient computation is critical, as it must be small enough to accurately approximate the derivative without introducing numerical instability. Equally important is the choice of the learning rate η . A large learning rate can cause divergence, while a rate that is too small may result in slow convergence, highlighting the importance of fine-tuning these parameters to ensure efficient and stable optimization.

4.3 Experimental evaluation

To demonstrate the validity of the DP-BUP method, the experiments focus on the same datasets presented in Section 3.3.

Recall that the research addresses two objectives: to protect sensitive information while concurrently preserving data utility in the context of multivariate TSC. Consequently, data perturbation necessitates a meticulous approach wherein the accuracy of driver detection is as low as possible while maintaining the accuracy of behavior or road-type detection close to the classification accuracy for the non-perturbed data. Two distinct scenarios for the automotive datasets are considered: (i) driver detection as PBC with road type detection as UFC, which is

termed *driver-vs-road-type classification*; and (ii) driver detection functioning as the PBC and behavior detection operating as the UFC, a setup referred to as *driver-vs-behavior classification*. For the HAR datasets the user detection is considered as PBC with activity detection as UFC, which is termed *user-vs-activity classification*.

The first experimental stage involved constructing classification models specifically trained to identify the driver, determine driver behavior, and categorize road types. These models utilized the FCN-LSTM architecture (Figure 3.7) and were trained on non-perturbed data from the selected datasets. The driving datasets were preprocessed following the same procedures proposed by El Mekki *et al.* [155], using the same feature selection (Table 3.1), being divided into training sets (comprising 70% of the total data) and test sets (consisting of the remaining 30%). Similarly, the HAR datasets were normalized and divided into training and test sets. The obtained results (Table 4.1) represented the accuracy achieved by the classification models in various tasks. In driver detection, the model achieved 0.9240 accuracy compared to 0.8986 for the UAH dataset and 0.9125 compared to 0.9510 for the HCRL dataset, comparable to reference experiments [155]. Additionally, for road type classification, the model achieved 0.9998 accuracy on the UAH dataset and 0.9615 on the HCRL dataset. For behavior detection, the accuracy was 0.8863 on the UAH dataset. The user and activity detection in MotionSense and PAMAP2 datasets were all above 0.96. These results indicated the models' accuracy in identifying drivers/users, road types, and behaviors/activities on non-protected data.

The process of generating the UFC model assumes a secure environment without any device-level perturbations. In both the training and testing phases of constructing the UFCs, data labels for each batch of records are stored in the data warehouse. After constructing the UFC models, there is the possibility to remove the data labels from the database. No label information is shared with third-party systems. Moreover, during the prediction phase, no explicit information that could identify the user is transmitted to the central database or recorded in any form. This procedure serves as a fundamental privacy protection strategy.

Further, the experiments involved computing feature importance to evaluate the relevance of each feature for driver detection and non-driver-related data classification. Subsequently, features were clustered (Figure 4.6) in $\mathcal{F}_p$, $\mathcal{F}_{up}$, and $\mathcal{F}_u$, according to the rules described in Section 4.2.1, considering the computed feature importance for driver/user identification and behavior/road type/activity detection, respectively. Figure 4.7 illustrates the number of features in each cluster. Further, the features in $\mathcal{F}_p$ and $\mathcal{F}_{up}$ were perturbed using w -event LDP approach, as described in Section 4.2.2.1, with variable privacy budget for each cluster, on the one hand, and, secondly, using DFT-based perturbation methods, described in Section 4.2.2.2.

4.3.1 Data perturbation using w -event level LDP

To evaluate the impact of distributing perturbation unequally between features, as outlined in Section 4.2.2, a scenario where features were not clustered was initially explored, with each feature receiving an identical amount of perturbation. This approach was a direct extension of

Table 4.1 Classification accuracy for unprotected test data.

Dataset	Classification	Achieved accuracy	Benchmark accuracy
UAH [147]	Driver detection	0.9240	0.8986 [155]
	Behavior detection	0.8863	-
	Road type detection	0.9998	-
HCRL [148]	Driver detection	0.9120	0.9510 [155]
	Road type detection	0.9615	-
Motionsense [28]	User detection	0.9796	0.9251 [28]
	Activity detection	0.9862	0.9620 [28]
PAMAP2 [149] (31 features)	User detection	0.9980	-
	Activity detection	0.9653	0.9527 [162]
PAMAP2 [149] (22 features)	User detection	0.9663	-
	Activity detection	0.9317	-

perturbation methods originally designed for univariate data, specifically the w -event LDP [22, 75]. In this case, $\epsilon_i = \epsilon_j$ for all $i, j \in \{1, \dots, d\}$, and the total privacy budget for all features was:

$$\epsilon_T = \sum_{i=0}^d \epsilon_i. \quad (4.27)$$

The budget ϵ_i for each feature F_i was uniformly distributed among the values belonging to the same feature, according to the formula $w \cdot \Delta g / \epsilon_i$ proposed by Kellaris et al. [22] for w -event LDP, where Δg represents the data sensitivity. The noise was generated using the Laplace mechanism [60] and applied to data batches of $w = 60$ records of sensor values. Figure 4.8 and Table 4.2 demonstrate that applying an equal perturbation to all considered features fulfilled the objectives stated in Equation 4.3 for the UAH and HCLR datasets, but failed to achieve the desired classification objectives for the PAMAP2 and MotionSense datasets. However, in all cases, the accuracy $\mathcal{A}'_p(\theta)$ of the PBC remained close to $\mathcal{A}'_u(\theta)$ for smaller perturbations, where $\mathcal{A}'_u(\theta) \approx \mathcal{A}_u$.

The presented approach for distributing the perturbation is based on a set of parameters, including the clustering parameter ρ_I , the privacy budget ϵ_T , and the cluster privacy budget distribution parameters α_p and α_{up} . The number of features in each cluster to determine potential values for ρ_I was analyzed. The criteria ensured that there is at least one feature in both $\mathcal{F}_u$ and $\mathcal{F}_p$ and that the number of features in $\mathcal{F}_{up}$ is less than the sum of the features in $\mathcal{F}_p$ and $\mathcal{F}_u$ to maintain manageable uncertainty. For the cluster privacy budget distribution parameters α_p and α_{up} , the rule stated in Proposition 1 was followed, with the condition that $\frac{\alpha_p}{|\mathcal{F}_p|} \leq \frac{\alpha_{up}}{|\mathcal{F}_{up}|}$. The considered values for ρ_I , α_p , and α_{up} are listed in Table 4.2.

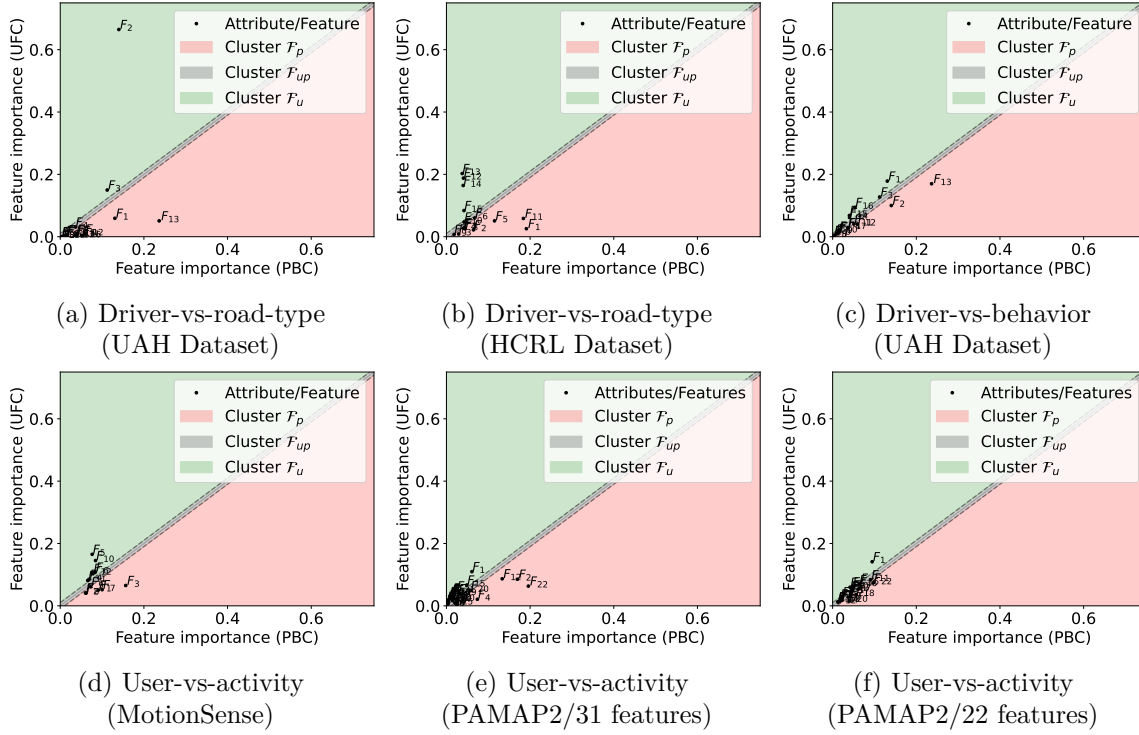


Figure 4.6 Feature clustering based on feature importance coefficients for two classifications (UFC and PBC), conducted using Random Forest with Gini importance ($\rho_I = 0.01$).

The perturbed data was verified against previously trained classification models to determine an appropriate perturbation level ϵ_T , aiming to achieve the desired levels of privacy and utility while adhering to the established constraints. Figures 4.9, 4.10, 4.11, 4.12, 4.13, and 4.14 show the classification accuracy for the considered parameter values. The findings, as presented in Table 4.2 (with the best results highlighted in bold), demonstrated that certain parameter settings for the introduced approach produced superior outcomes compared to uniformly distributing noise across all features, aligning with the objectives outlined in Equation 4.3. The overall performance evaluation was based on the calculation of the classification *utility-privacy balance score*, $\mathcal{B}_{UP}(\theta)$ (Equation 4.4).

For the *driver-vs-road-type* classification using the UAH dataset, the accuracy $\mathcal{A}'_p(\theta)$ of the PBC was lower than the reference accuracy, while the classification accuracy $\mathcal{A}'_u(\theta)$ of the UFC is higher. The minimum value of $\mathcal{B}_{UP}(\theta)$ was achieved when clustering resulted in a large number of features in $\mathcal{F}_p$ and a small number of features in $\mathcal{F}_{up}$ and $\mathcal{F}_u$. This configuration allocated most of the perturbation to the features in $\mathcal{F}_p$, amplified by the large number of features in this cluster. Additionally, the Mean Average Error (MAE) value was lower than the reference, indicating higher data utility. The privacy budget ϵ_T in this scenario was significantly higher ($\epsilon_T = 100$ compared to $\epsilon_T = 15$) than the presented approach, indicating an overall lower perturbation.

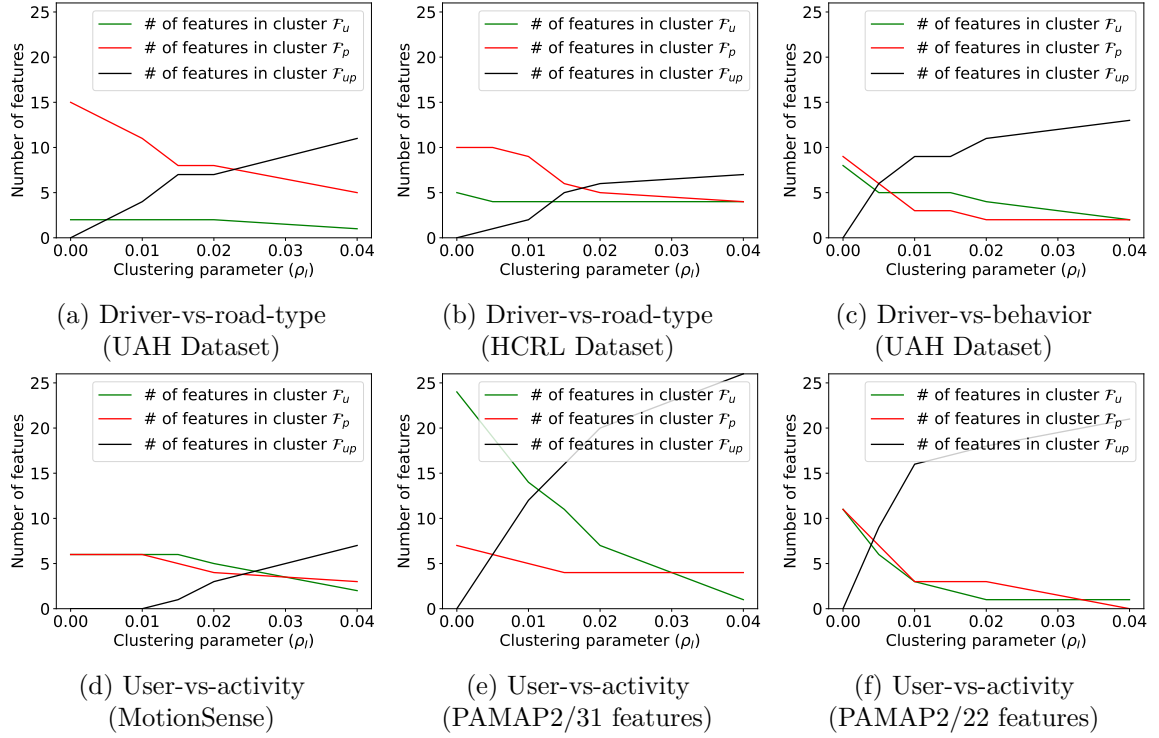


Figure 4.7 The number of features in each cluster ($\rho_I = 0.01$).

For the HCRL dataset, in the same type of classification, the best results were obtained when the clustering ($\rho_I = 0$) did not include features in $\mathcal{F}_{up}$. Similar to the UAH dataset, $\mathcal{A}'_u(\theta)$ and $\mathcal{A}'_p(\theta)$ were comparable to those obtained with uniform perturbation across all features, with the crucial difference being that smaller perturbation was needed ($\epsilon_T = 120$ compared to $\epsilon_T = 30$).

In the case of the *driver-vs-behavior* classification, both classifiers relied on a similar set of features (Figure 4.6c). However, Figure 4.11 showed that the protection of data is possible under the considered constraints with $\rho_I = 0$, $\alpha_p = 0.3$, $\alpha_{up} = 0.7$, and $\epsilon_T = 50$. Similar accuracy for PBC and UFC were obtained for significantly lower perturbation ($\epsilon_T = 50$ compared to $\epsilon_T = 30$).

The HAR datasets were analyzed using the complete set of features, making the feature separation process challenging. Figures 4.6d and 4.6e demonstrated that, for both datasets, all features were nearly equally important for the UFC and PBC classifications. Applying identical perturbations to all features (Figures 4.8d and 4.8e) did not yield the desired results, as the accuracy levels of UFC and PBC remained similar. Additionally, for the PAMAP2 data (Figure 4.8e), the accuracy of PBC consistently exceeded that of UFC across all perturbations considered.

Implementing the protection approach for *user-vs-behavior* classification, which involves applying perturbation solely to the features significant for the PBC classification, produced favorable results for the MotionSense data with $\rho_I = 0$, $\alpha_p = 0.1$, $\alpha_{up} = 0.9$, and $\epsilon_T = 25$ (Table

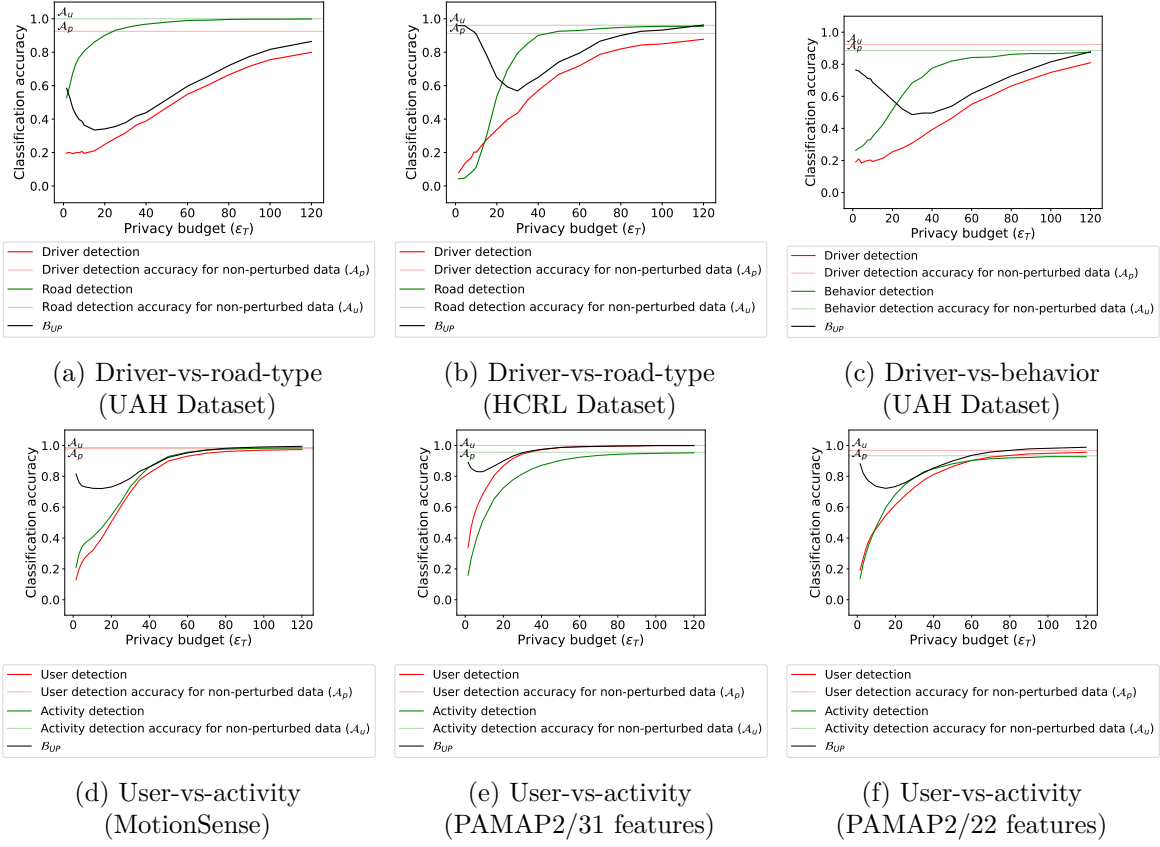


Figure 4.8 Classification accuracy with equally distributed perturbation for all features using the w -event LDP approach.

4.2). The user detection accuracy was reduced to 0.3891, while the behavior classification accuracy was 0.6302. However, for the PAMAP2 dataset, the results were not satisfactory, as the user classification accuracy exceeded the behavior detection accuracy.

To achieve the targeted objectives, the following feature selection process for the PAMAP2 dataset was used:

- All 31 features in the original dataset were clustered, as described in Section 4.2.1.
- Next, 22 features were selected, all from the cluster $\mathcal{F}_u$, obtained using the clustering parameter $\rho_I = 0$.
- The feature clustering process was repeated a second time, but only on the selected 22 features.
- Finally, the perturbation was applied.

Figure 4.14 demonstrates that the presented approach can be effectively applied even to datasets with features that are challenging to separate for the two classifiers. For the configuration with 22 features, $\rho_I = 0.015$, $\alpha_p = 0.3$, $\alpha_{up} = 0.7$, and $\epsilon_T = 15$, an accuracy of 0.5314 for user detection and 0.6269 for behavior detection (Table 4.2) was obtained.

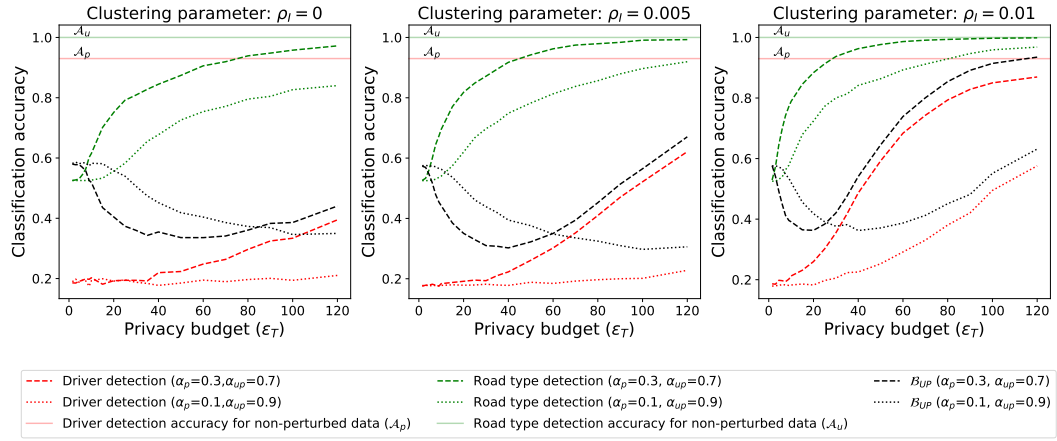


Figure 4.9 Driver-vs-road-type classification accuracy on the LDP perturbed data (UAH dataset).

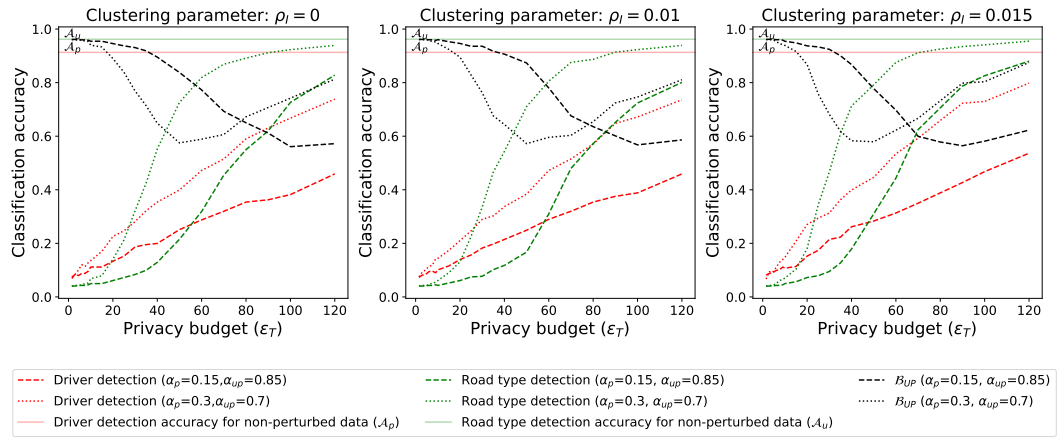


Figure 4.10 Driver-vs-road-type classification accuracy on the LDP perturbed data (HCRL dataset).

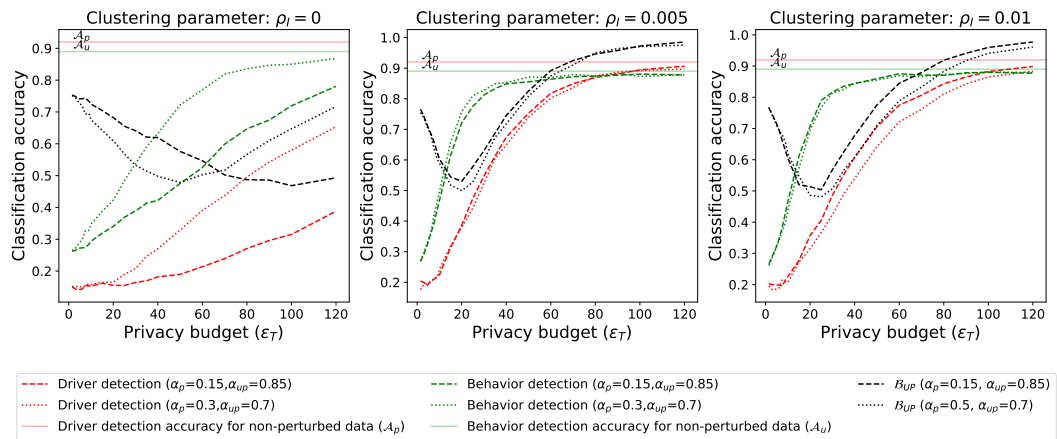


Figure 4.11 Driver-vs-behavior classification accuracy on the LDP perturbed data (UAH dataset).

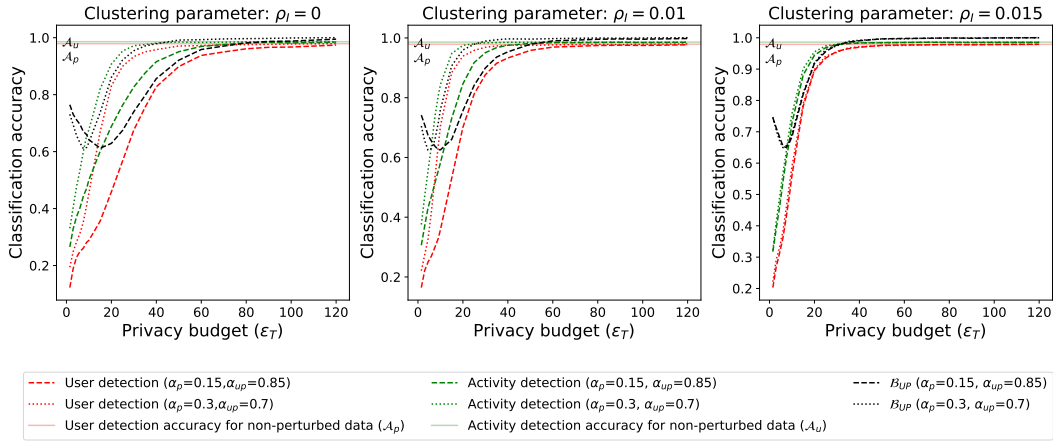


Figure 4.12 User-vs-activity classification accuracy on the LDP perturbed data (MotionSense dataset).

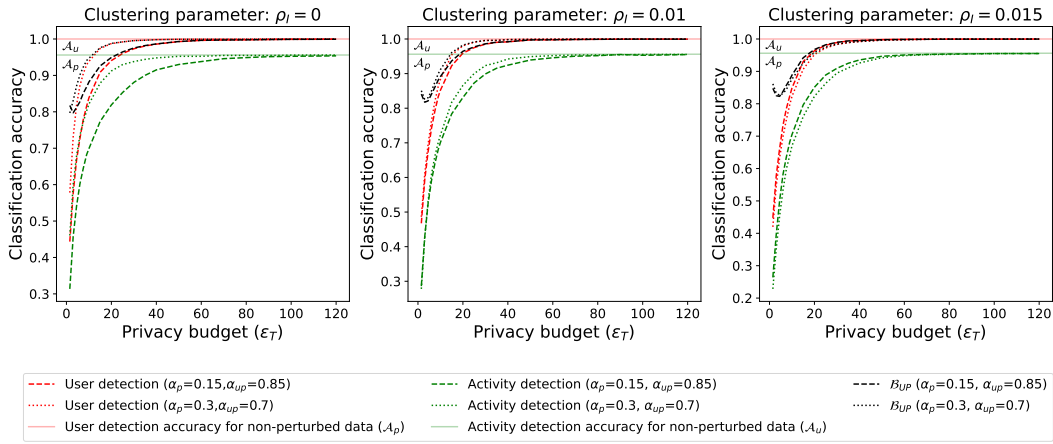


Figure 4.13 User-vs-activity classification accuracy on the LDP perturbed data (PAMAP2 dataset with 31 features).

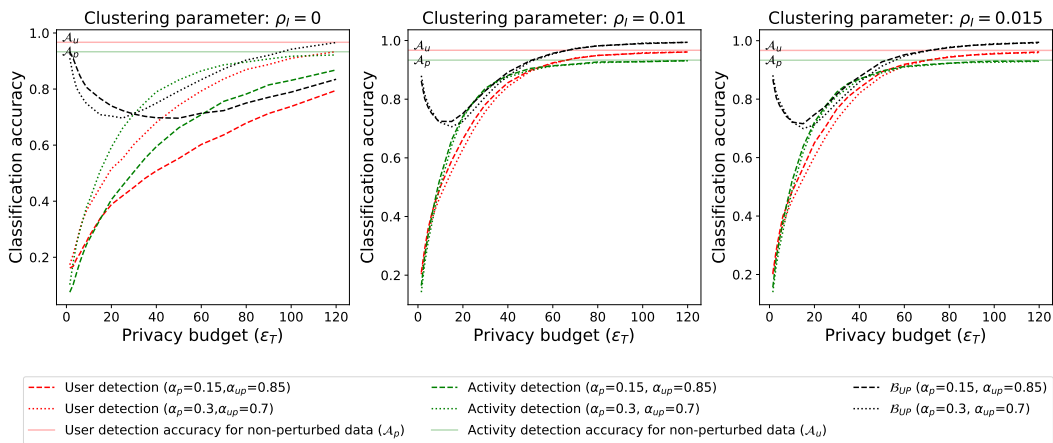


Figure 4.14 User-vs-activity classification accuracy on the LDP perturbed data (PAMAP2 dataset with 22 features).

Table 4.2 Classification accuracy for perturbed data using DP-BUP with w-event LDP.

Dataset	Classification scenario	Perturbation approach	Clustering parameter (ρ_I)	# of features per cluster ($\mathcal{F}_p, \mathcal{F}_{up}, \mathcal{F}_u$)	Perturbation parameters (α_p, α_{up})	$\min(\mathcal{B}_{UP}) \epsilon_T$	$\mathcal{A}'_p(\theta)$	$\mathcal{A}'_u(\theta)$	MAE	
UAH	Driver-vs-road-type	w-event LDP [22, 75] (no feature clustering)	-	-	-	0.3348	15	0.2103	0.8610	1.017
		w-event LDP (feature clustering, DP-BUP method)	0	{15,0,2}	{0.3,0.7}	0.3362	50	0.2253	0.8759	0.8906
					{0.15,0.85}	0.3189	120	0.2337	0.9097	0.7396
					{0.1,0.9}	0.3413	120	0.2071	0.8474	1.1156
			0.005	{12,3,2}	{0.3,0.7}	0.3027	35	0.2000	0.8883	0.7667
					{0.15,0.85}	0.2975	80	0.2077	0.9045	0.6604
					{0.1,0.9}	0.3145	120	0.2201	0.8980	0.6556
			0.01	{10,5,2}	{0.3,0.7}	0.3601	15	0.2272	0.8467	1.0856
					{0.15,0.85}	0.3693	30	0.2441	0.8551	0.9427
					{0.1,0.9}	0.3674	50	0.2493	0.8642	0.8135
HCRL	Driver-vs-road-type	w-event LDP [22, 75] (no clustering)	-	-	-	0.5683	30	0.4356	0.7941	0.5291
		w-event LDP (feature clustering, DP-BUP method)	0	{10,0,5}	{0.3,0.7}	0.5765	60	0.4613	0.8235	0.4306
					{0.15,0.85}	0.5629	120	0.4377	0.8076	0.4302
					{0.1,0.9}	0.6217	120	0.3326	0.5723	0.6453
			0.01	{9,2,4}	{0.3,0.7}	0.5785	60	0.4635	0.8235	0.4304
					{0.15,0.85}	0.5749	100	0.3755	0.6945	0.5168
					{0.1,0.9}	0.6252	120	0.3240	0.5588	0.6446
			0.015	{7,4,4}	{0.3,0.7}	0.5803	40	0.3841	0.6968	0.4946
					{0.15,0.85}	0.5707	90	0.4263	0.7750	0.4428
					{0.1,0.9}	0.5801	120	0.3905	0.7058	0.4866
UAH	Driver-vs-behavior	w-event LDP [22, 75] (no clustering)	-	-	-	0.4859	30	0.3090	0.6844	0.5077

		w-event LDP (feature clustering, DP-BUP method)	0	{9,0,8}	{0.3,0.7} {0.15,0.85}	0.4692 0.4834	50 100	0.3181 0.3305	0.7220 0.7175	0.3651 0.3655
			0.005	{6,6,5}	{0.1,0.9}	0.4861	120	0.2714	0.6487	0.4570
					{0.3,0.7}	0.4986	20	0.3759	0.7545	0.4110
					{0.15,0.85}	0.5426	20	0.4000	0.7201	0.5002
			0.01	{4,8,5}	{0.1,0.9}	0.5546	20	0.3435	0.6324	0.6269
					{0.3,0.7}	0.4836	25	0.3714	0.7707	0.3609
					{0.15,0.85}	0.4965	20	0.3480	0.7207	0.4458
					{0.1,0.9}	0.5282	20	0.3603	0.6902	0.4888
MotionSense	User-vs-behavior	w-event LDP [22, 75] (no clustering)	-	-	-	0.7205	15	0.3999	0.4658	1.1431
		w-event LDP DP-BUP method)	0	{6,0,6}	{0.3,0.7}	0.6152	7.5	0.3594	0.5994	1.0586
		(feature clustering,			{0.15,0.85}	0.6149	15	0.3536	0.5944	1.0563
			0.005	{6,1,5}	{0.1,0.9} {0.3,0.7}	0.6148 0.6282	25 6	0.3891 0.4207	0.6302 0.6428	0.9520 0.8359
					{0.15,0.85}	0.6232	10	0.3559	0.5837	0.9839
					{0.1,0.9}	0.6239	15	0.3580	0.5846	0.9790
			0.01	{4,5,4}	{0.3,0.7}	0.6493	6	0.3830	0.5679	0.9770
					{0.15,0.85}	0.6497	7.5	0.4292	0.6150	0.9456
					{0.1,0.9}	0.6494	9	0.4205	0.6058	0.9823
PAMAP2 (31 features)	User-vs-behavior	w-event LDP [22, 75] (no clustering)	-	-	-	0.8643	7.5	0.6978	0.4292	3.8142
		w-event LDP	0	{7,0,24}	{0.3,0.7} {0.15,0.85}	0.7969 0.7972	1.5 3	0.5781 0.5763	0.4602 0.4574	2.0301 2.0302
		(feature clustering, DP-BUP method)			{0.1,0.9}	0.7975	4.5	0.5799	0.4607	2.0285
			0.01	{5,13,13}	{0.3,0.7}	0.8262	3	0.6083	0.4241	3.6400
					{0.15,0.85}	0.8173	3	0.5919	0.4278	3.6966
					{0.1,0.9}	0.8131	4.5	0.6275	0.4795	2.7673
			0.015	{4,6,11}	{0.3,0.7}	0.8260	4.5	0.6327	0.4529	3.2755

					{0.15,0.85}	0.8228	4.5	0.6615	0.5003	2.8449
					{0.1,0.9}	0.8230	4.5	0.6597	0.4972	2.8259
PAMAP2 (22 features)	User-vs-behavior	w-event LDP [22, 75] (no clustering)	-	-	-	0.7031	15	0.5423	0.6201	1.572
		w-event LDP (feature clustering, DP-BUP method)	0	{12,0,10}	{0.3,0.7}	0.6978	20	0.4933	0.5701	1.6605
					{0.15,0.85}	0.6964	50	0.5348	0.6238	1.3299
					{0.1,0.9}	0.6988	80	0.5510	0.6412	1.2466
			0.01	{3,16,3}	{0.3,0.7}	0.6959	15	0.5519	0.6487	1.3336
					{0.15,0.85}	0.7040	10	0.5031	0.5693	1.8498
					{0.1,0.9}	0.7087	9	0.4821	0.5376	2.1530
			0.015	{3,17,2}	{0.3,0.7}	0.6927	15	0.5314	0.6269	1.4866
					{0.15,0.85}	0.7060	15	0.5737	0.6594	1.3615
					{0.1,0.9}	0.7061	15	0.5664	0.6481	1.4119

4.3.2 Data perturbation using the DFT-based approach

A similar procedure was employed to evaluate the effectiveness of the DFT approach. Initially, the perturbation was distributed equally across all features, ensuring each feature received the same amount. Further, the impact of this equal distribution on the classification accuracy for both PBC and UFC was measured. In this scenario, $\sigma_i = \sigma_j$ for all $i, j \in \{1, \dots, d\}$, where d represents the total number of features. The total perturbation budget allocated across all features was:

$$\sigma_T = \sum_{i=0}^d \sigma_i. \quad (4.28)$$

The perturbation σ_i for each feature F_i was uniformly distributed across the values within the same feature using a Gaussian distribution. Similar to the w-event LDP approach, Figure 4.15 and Table 4.3 demonstrate that applying equal perturbation to all considered features met the objectives outlined in Equation 4.3 for the UAH and HCLR datasets. However, this approach did not achieve the expected classification outcomes for the PAMAP2 and MotionSense datasets.

Furthermore, it was demonstrated that applying the DFT perturbation selectively to specific features, as outlined in Section 4.2.2.2, yields superior results compared to the equal distribution approach. In the case of DFT perturbation, the introduced approach is based on a set of parameters consisting of the clustering parameter ρ_I , the perturbation budget σ_T , the number of selected Fourier coefficients k , and the cluster privacy budget distribution parameters α_p and α_{up} . The same procedure was utilized as in the case of w -event-level LDP perturbation of choosing the values for ρ_I , α_p , and α_{up} , based on Proposition 3. The considered values for ρ_I , α_p , and α_{up} are listed in Table 4.3. For simplicity, $k = 25$ was considered.

The DFT perturbed data was verified against the trained classification models for PBC and UFC. The objective was to determine a suitable perturbation level σ_T for achieving the stated performance constraints. Figures 4.16, 4.17, 4.18, 4.19, 4.20, and 4.21 show the classification accuracy for the considered parameter values. As shown in Table 4.3 (with the best results highlighted in bold), the findings indicate that specific parameter settings for the developed approach yielded superior results compared to the uniform distribution of noise across all features, in alignment with the objectives outlined in Equation 4.3. The overall performance evaluation was conducted by calculating the classification *utility-privacy balance score*, $\mathcal{B}_{UP}(\theta)$ (Equation 4.4).

The results, similar to those obtained using the w-event LDP perturbation, confirm that the presented approach also delivers strong performance for the DFT perturbation across the tested datasets within the presented scenario.

Even though the results for the two perturbation methods tested were similar, in certain cases, the LDP method provided superior outcomes by simultaneously achieving higher $\mathcal{A}'_u(\theta)$

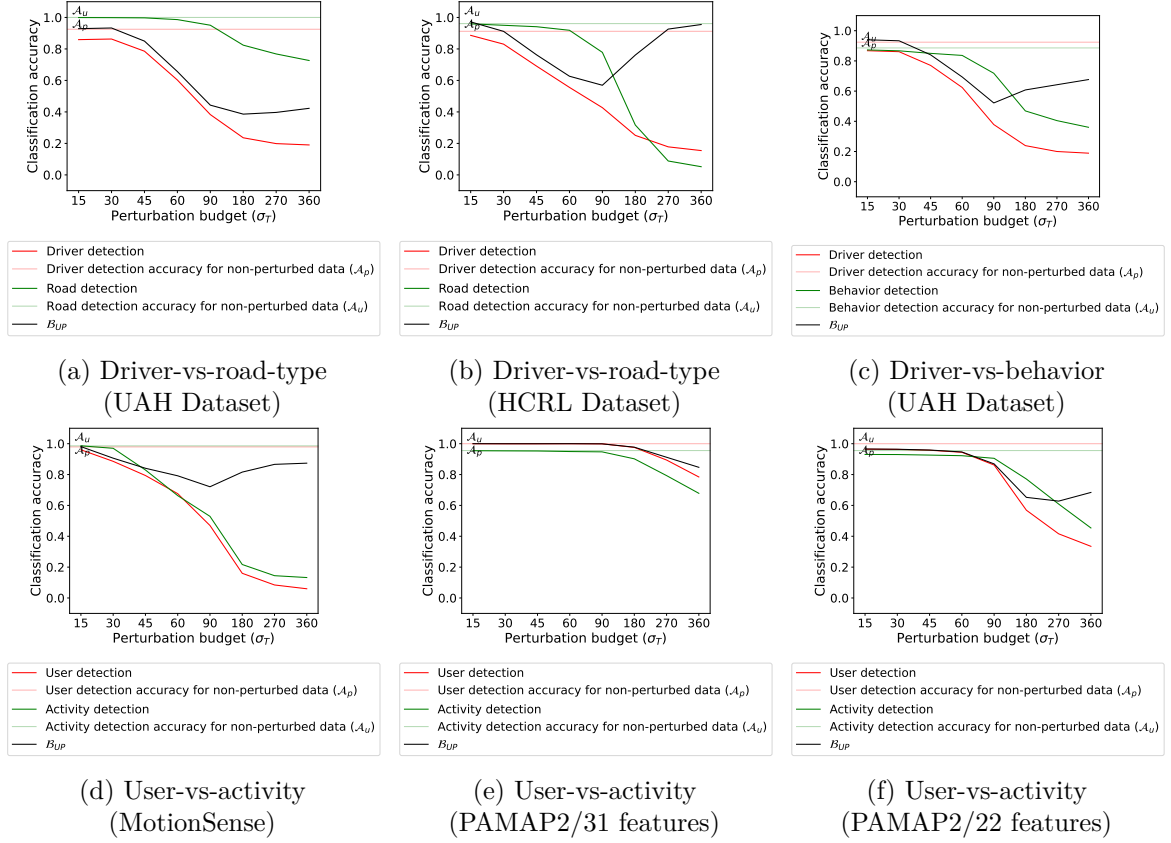


Figure 4.15 Classification accuracy with equally distributed perturbation for all features using the DFT approach.

and lower $\mathcal{A}'_p(\theta)$. Conversely, the DFT method performed better in other instances. For example, in the *driver-vs-road-type* classification task for the UAH dataset, using the optimal parameter set (minimizing $\mathcal{B}_{UP}$), the accuracy of the UFC was significantly higher for the LDP method (0.9045) compared to the accuracy of the UFC with the DFT perturbation (0.8597). Similarly, for the same dataset, for the *driver-vs-behavior* classification, the accuracy of the UFC was again higher for the LDP method (0.7220) compared to the accuracy of the UFC with the DFT perturbation (0.6993), while the accuracy of the PBC was lower for the LDP method (0.3181) compared to the accuracy of the PBC with the DFT perturbation (0.3402). However, for the HCRL, MotionSense, and PAMAP2 datasets, the DFT approach performed better (Tables 4.2 and 4.3).

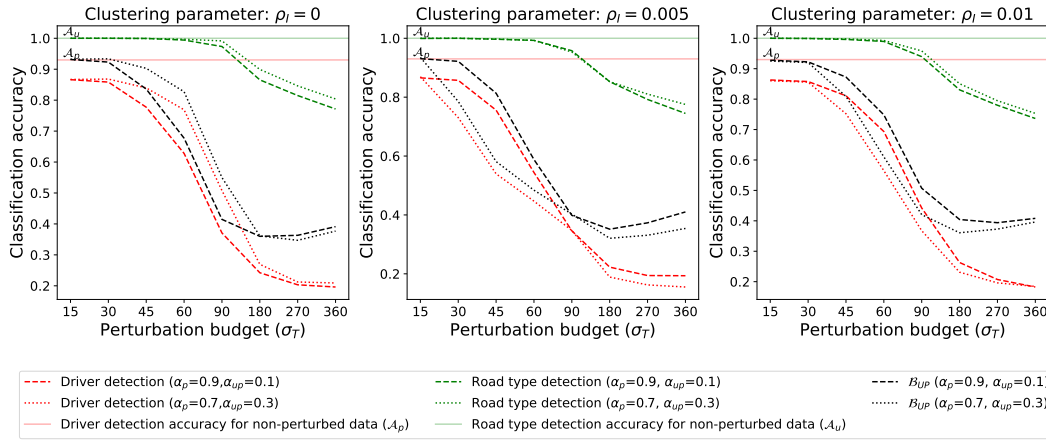


Figure 4.16 Driver-vs-road-type classification accuracy on the DFT perturbed data (UAH dataset).

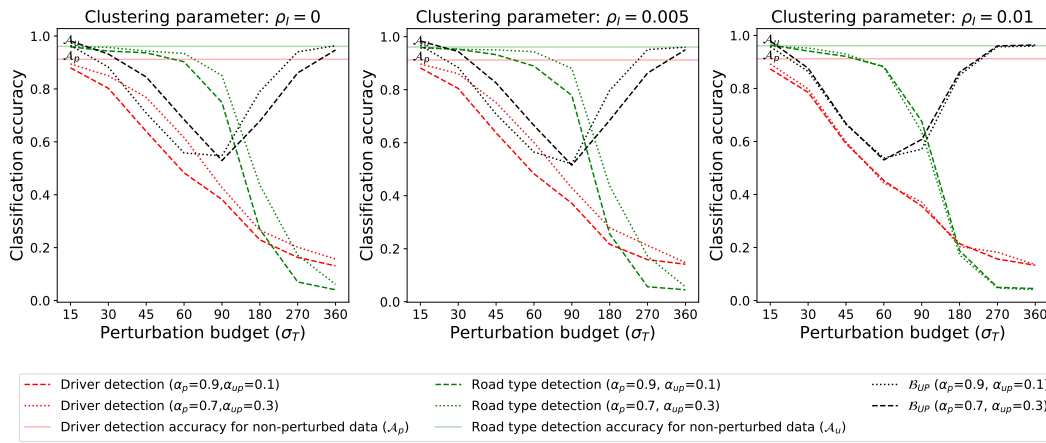


Figure 4.17 Driver-vs-road-type classification accuracy on the DFT perturbed data (HCRL dataset).

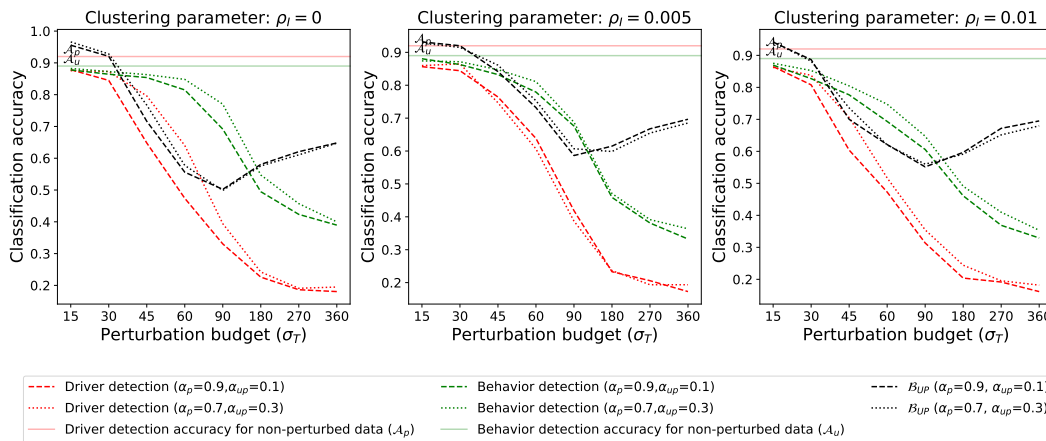


Figure 4.18 Driver-vs-behavior classification accuracy on the DFT perturbed data (UAH dataset).

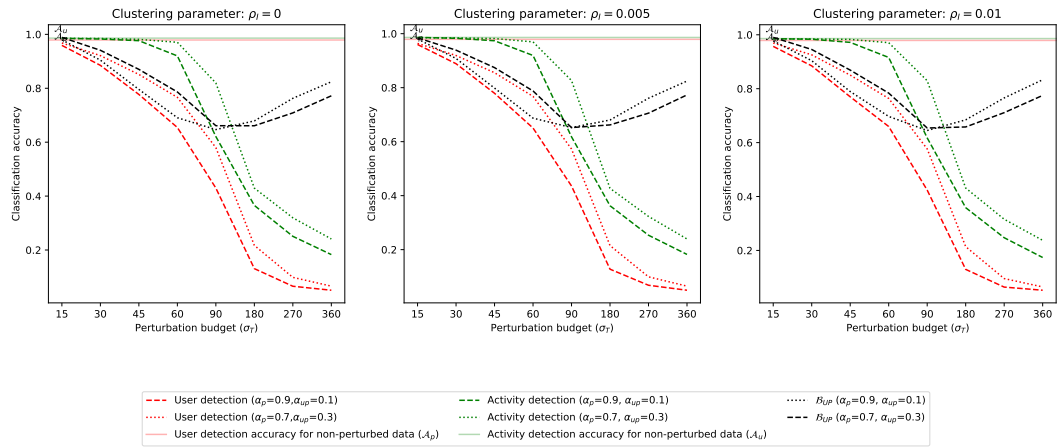


Figure 4.19 User-vs-activity classification accuracy on the DFT perturbed data (MotionSense dataset).

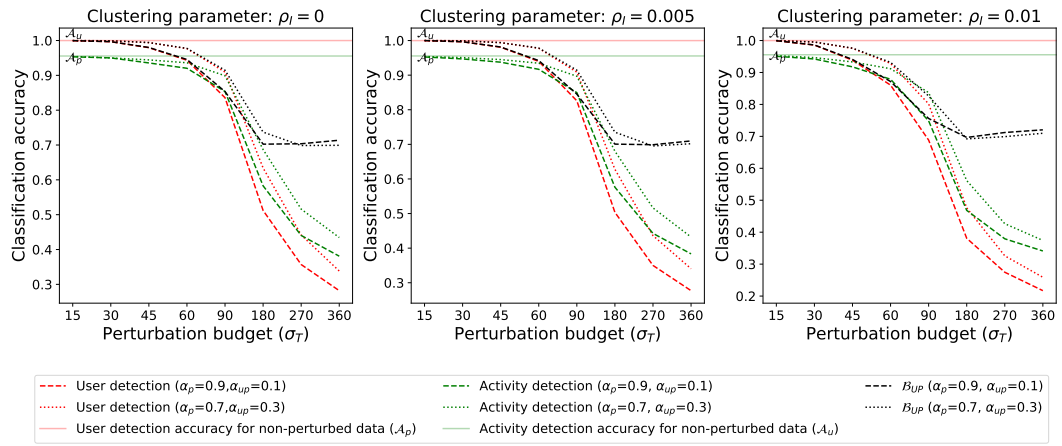


Figure 4.20 User-vs-activity classification accuracy on the DFT perturbed data (PAMAP2 dataset with 31 features).

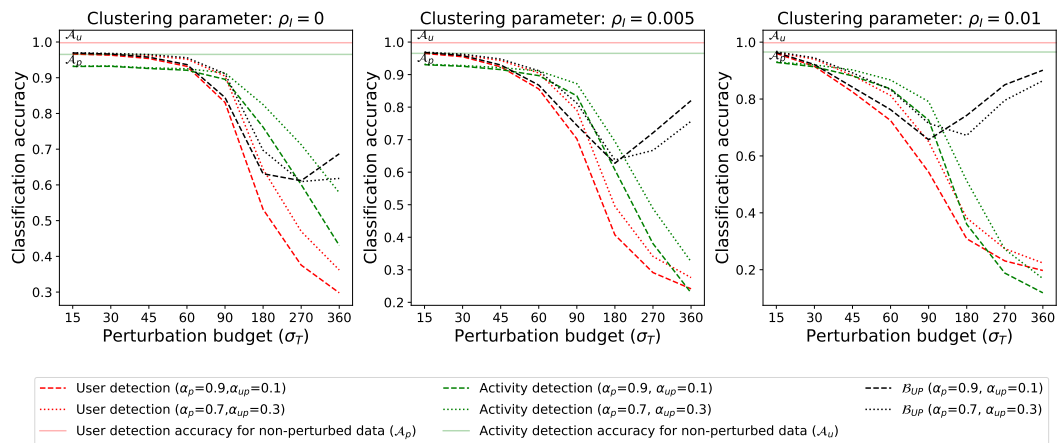


Figure 4.21 User-vs-activity classification accuracy on the DFT perturbed data (PAMAP2 dataset with 22 features).

Table 4.3 Classification accuracy for perturbed data using DP-BUP with DFT-based perturbation ($k = 25$).

Dataset	Classification scenario	Perturbation approach	Clustering parameter (ρ_I)	# of features per cluster ($\mathcal{F}_p, \mathcal{F}_{up}, \mathcal{F}_u$)	Perturbation parameters (α_p, α_{up})	$\min(\mathcal{B}_{UP}) \sigma_T$	$\mathcal{A}'_p(\theta)$	$\mathcal{A}'_u(\theta)$	MAE	
UAH	Driver-vs-road-type	DFT-based [38, 29] (no feature clustering)	-	-	-	0.3859	180	0.2357	0.8240	1.2183
		DFT-based (feature clustering, DP-BUP method)	0	{15,0,2}	{0.9,0.1}	0.3533	180	0.2383	0.8694	1.0965
					{0.85,0.15}	0.3609	180	0.2500	0.8740	1.0342
					{0.7,0.3}	0.3484	270	0.2181	0.8512	1.2776
			0.005	{12,3,2}	{0.9,0.1}	0.3476	180	0.2194	0.8538	1.2183
					{0.85,0.15}	0.3559	180	0.2253	0.8500	1.2165
					{0.7,0.3}	0.3293	180	0.2045	0.8597	1.2149
			0.01	{10,5,2}	{0.9,0.1}	0.3867	180	0.2564	0.8467	1.2251
					{0.85,0.15}	0.3828	180	0.2532	0.8480	1.2210
					{0.7,0.3}	0.3534	180	0.2279	0.8564	1.2178
HCRL	Driver-vs-road-type	DFT-based [38, 29] (no clustering)	-	-	-	0.5693	90	0.4270	0.7782	0.6958
		DFT-based (feature clustering, DP-BUP method)	0	{10,0,5}	{0.9,0.1}	0.5471	90	0.3819	0.7488	0.6236
					{0.85,0.15}	0.5434	90	0.3776	0.7488	0.5945
					{0.7,0.3}	0.5292	90	0.4270	0.8506	0.4911
			0.01	{9,2,4}	{0.9,0.1}	0.5376	60	0.4527	0.8823	0.4673
					{0.85,0.15}	0.5365	80	0.3562	0.7307	0.6910
					{0.7,0.3}	0.5302	60	0.4442	0.8800	0.4686
			0.015	{7,4,4}	{0.9,0.1}	0.5099	60	0.4227	0.8778	0.4672
					{0.85,0.15}	0.5343	60	0.4506	0.8846	0.4658
					{0.7,0.3}	0.5575	90	0.3519	0.6923	0.6943
UAH	Driver-vs-behavior	DFT-based [38, 29] (no clustering)	-	-	-	0.5215	90	0.3785	0.7181	0.6170

		DFT-based (feature clustering, DP-BUP method)	0	{9,0,8}	{0.9,0.1}	0.5090	90	0.3298	0.6811	0.5503
					{0.85,0.15}	0.5048	90	0.3402	0.6993	0.5195
			0.005	{6,6,5}	{0.7,0.3}	0.5153	90	0.4038	0.7688	0.4304
					{0.9,0.1}	0.5984	90	0.4090	0.6435	0.6335
					{0.85,0.15}	0.5891	90	0.4175	0.6694	0.6265
			0.01	{4,8,5}	{0.7,0.3}	0.5540	90	0.3844	0.6818	0.6176
					{0.9,0.1}	0.5623	90	0.3175	0.5948	0.6397
					{0.85,0.15}	0.5727	90	0.3259	0.5889	0.6319
					{0.7,0.3}	0.5535	90	0.3564	0.6487	0.6228
MotionSense	User-vs-behavior	DFT-based [38, 29] (no clustering)	-	-	-	0.7208	90	0.4697	0.5290	0.8726
		DFT-based (feature clustering, DP-BUP method)	0	{6,0,6}	{0.9,0.1}	0.6467	90	0.4298	0.6208	0.7722
					{0.85,0.15}	0.6484	90	0.4668	0.6625	0.7298
			0.005	{6,1,5}	{0.7,0.3}	0.6609	180	0.2171	0.4296	1.2012
					{0.9,0.1}	0.6546	90	0.3724	0.5495	0.8590
					{0.85,0.15}	0.6448	90	0.4238	0.6174	0.7731
			0.01	{4,5,4}	{0.7,0.3}	0.6474	90	0.4645	0.6614	0.7307
					{0.9,0.1}	0.6448	90	0.3679	0.5609	0.8586
					{0.85,0.15}	0.6475	90	0.3706	0.5593	0.8581
					{0.7,0.3}	0.6548	90	0.3257	0.5099	0.8585
PAMAP2 (31 features)	User-vs-behavior	DFT-based [38, 29] (no clustering)	-	-	-	0.8469	360	0.7842	0.6783	1.3322
		DFT-based (feature clustering, DP-BUP method)	0	{7,0,24}	{0.9,0.1}	0.7024	180	0.5121	0.5829	0.5988
					{0.85,0.15}	0.6985	270	0.3698	0.4571	0.8472
			0.005	{7,5,19}	{0.7,0.3}	0.6985	270	0.4418	0.5160	0.6979
					{0.9,0.1}	0.6989	270	0.3500	0.4426	0.9987
					{0.85,0.15}	0.6978	270	0.3704	0.4585	0.9989
			0.01	{5,13,13}	{0.7,0.3}	0.6953	270	0.4361	0.5162	0.9976
					{0.9,0.1}	0.6965	180	0.3808	0.4682	0.6710

					{0.85,0.15}	0.6937	180	0.4035	0.4906	0.6701
					{0.7,0.3}	0.6921	180	0.4760	0.5615	0.6683
PAMAP2 (22 features)	User-vs-behavior	DFT-based [38, 29] (no clustering)	-	-	-	0.6816	270	0.4905	0.6291	1.4072
		DFT-based (feature clustering, DP-BUP method)	0	{12,0,10}	{0.9,0.1}	0.6058	270	0.3750	0.6009	1.2643
					{0.85,0.15}	0.5987	270	0.3938	0.6317	1.1950
					{0.7,0.3}	0.6062	270	0.4683	0.7126	0.9843
			0.005	{7,8,7}	{0.9,0.1}	0.6275	180	0.4135	0.6073	0.9420
					{0.85,0.15}	0.6301	180	0.4374	0.6302	0.9409
					{0.7,0.3}	0.6290	180	0.4850	0.6946	0.9398
			0.01	{3,16,3}	{0.9,0.1}	0.6602	90	0.5447	0.7262	0.4828
					{0.85,0.15}	0.6719	90	0.5680	0.7422	0.4805
					{0.7,0.3}	0.6685	180	0.3831	0.5123	0.9416

4.3.3 Testing the adversarial models

The scenario in which the attacker (Figure 3.2) possessed access to all potential data sources was examined. The data contained both the training dataset with labels, the unlabeled data collected from the devices, and other observable, non-perturbed data. Consequently, the attacker was capable of constructing a PBC that promptly and accurately re-identifies users based on intercepted data. Another potential privacy breach is associated with “honest-but-curious” entities, such as a data analyst who has access to all the data needed to construct a classifier that compromises privacy. Tables 4.2 and 4.3 confirm that the proposed privacy model is constructed to effectively safeguard against such attacks.

In addition to the initial adversarial model, which assumes that the attacker has access to non-perturbed training data, a scenario where the attacker could obtain a sufficient amount of perturbed data along with the corresponding labels to construct a PBC was also considered.

Let $\hat{\mathbf{X}}_i^t$ be the time series data collected for a period of time, stored locally at time t , on the sensor-based device i and perturbed using a perturbation mechanism $\mathcal{M}$, such that $\hat{\mathbf{X}}_i^t = \mathcal{M}(\mathbf{X}_i^t)$. Moreover, let the adversarial PBC model trained using perturbed data be defined by the function $\hat{f}_p : \mathbb{R}^{w \cdot d} \rightarrow \mathcal{C}_p, \mathcal{C}_p = \{1, \dots, n\}$, where d is the number of features, w is the number of records, and n the number of categories, that is applied to any input data. The function $\hat{f}$ breaks the user privacy by identifying the set of collected values as belonging to a user from a specified class $\mathcal{C}_p$. Similarly to Equation 4.2, let us define the classification accuracy for the model trained with perturbed data:

$$\hat{\mathcal{A}}'_p(\theta) = \frac{1}{N \cdot T} \sum_{i=1}^N \sum_{t=1}^T \mathbb{I}(\mathbf{Y}_{pi}^t = \hat{f}_p(\hat{\mathbf{X}}_i^t)), \quad (4.29)$$

where θ is the set of perturbation parameters.

For evaluation, the optimal perturbation parameters for each dataset, classification type, and the two tested perturbation methods (LDP and DFT-based), as highlighted in Tables 4.2 and 4.3 were considered. Using these parameter sets, the perturbed data and corresponding labels to train models using the FCN-LSTM architecture (Figure 3.7) were employed. The driver/user detection accuracy was then computed using Equation 4.29. The results, presented in Table 4.4, indicate an increase in the classification accuracy of the PBC ($\hat{\mathcal{A}}'_p(\theta)$) when the classification model is trained with perturbed data, compared to when unperturbed data is used ($\hat{\mathcal{A}}_p(\theta)$). For the UAH, HCRL, MotionSense, and PAMAP2 (with 22 features) datasets, the classification accuracy $\hat{\mathcal{A}}'_p(\theta)$ remained lower than when unperturbed data is used for both training and testing ($\mathcal{A}_p$). Thus, the privacy objective stated in Equation 4.3 was still fulfilled, as $\hat{\mathcal{A}}'_p(\theta) \ll \mathcal{A}_p$. As expected, in the case of the PAMAP2 dataset with all 31 features, the accuracy $\hat{\mathcal{A}}'_p(\theta)$ was close to $\mathcal{A}_p$, indicating that a feature selection process is essential for enhancing privacy.

Additionally, the results demonstrate that the DFT-based perturbation offers better protection under the advanced adversarial model, even though protection was similar for both perturbation methods when applied to the initial adversarial model.

Table 4.4 Classification accuracy for the advanced adversarial model.

Dataset	Classification scenario	Perturbation approach	Clustering using parameter (ρ_I)	# of features per cluster $(\mathcal{F}_p, \mathcal{F}_{up}, \mathcal{F}_u)$	Perturbation parameters (α_p, α_{up})	$\mathcal{A}'_p(\theta)$	$\hat{\mathcal{A}}'_p(\theta)$	$\mathcal{A}_p$
		feature clustering (DP-BUP method)						
UAH	Driver-vs-road-type	LDP-based	0.005	{12,3,2}	{0.15,0.85}	0.2077	0.7681	0.9240
		DFT-based	0.005	{12,3,2}	{0.7,0.3}	0.2045	0.4396	0.9240
HCRL	Driver-vs-road-type	LDP-based	0	{10,0,5}	{0.15,0.85}	0.4377	0.7896	0.9120
		DFT-based	0.015	{7,4,4}	{0.9,0.1}	0.4227	0.7081	0.9120
UAH	Driver-vs-behavior	LDP-based	0	{9,0,8}	{0.3,0.7}	0.3181	0.7590	0.9240
		DFT-based	0	{9,0,8}	{0.85,0.15}	0.3402	0.7233	0.9240
MotionSense	User-vs-behavior	LDP-based	0	{6,0,6}	{0.1,0.9}	0.3891	0.9567	0.9796
		DFT-based	0.005	{6,1,5}	{0.85,0.15}	0.4238	0.8780	0.9796
			0.01	{4,5,4}	{0.9,0.1}	0.3679	0.8561	0.9796
PAMAP2 (31 features)	User-vs-behavior	LDP-based	0	{7,0,24}	{0.3,0.7}	0.5781	0.9990	0.9980
		DFT-based	0.01	{5,13,13}	{0.7,0.3}	0.4760	0.9755	0.9980
PAMAP2 (22 features)	User-vs-behavior	LDP-based	0.015	{3,17,2}	{0.3,0.7}	0.5314	0.8931	0.9980
		DFT-based	0	{12,0,10}	{0.85,0.15}	0.3938	0.8660	0.9980

4.3.4 Computing the optimum perturbation parameters

Following the optimization strategy presented in Section 4.2.4, numerical gradients were computed using the finite differences method, and gradient descent was iteratively applied to update the parameters θ . This approach was utilized to identify the optimal parameters across all datasets for both data perturbation techniques, namely DP and DFT-based perturbation. The primary objective is to determine the parameter set θ such that it minimizes $\mathcal{B}_{UP}(\theta)$, for balancing the performance of the PBC and UFC classifiers.

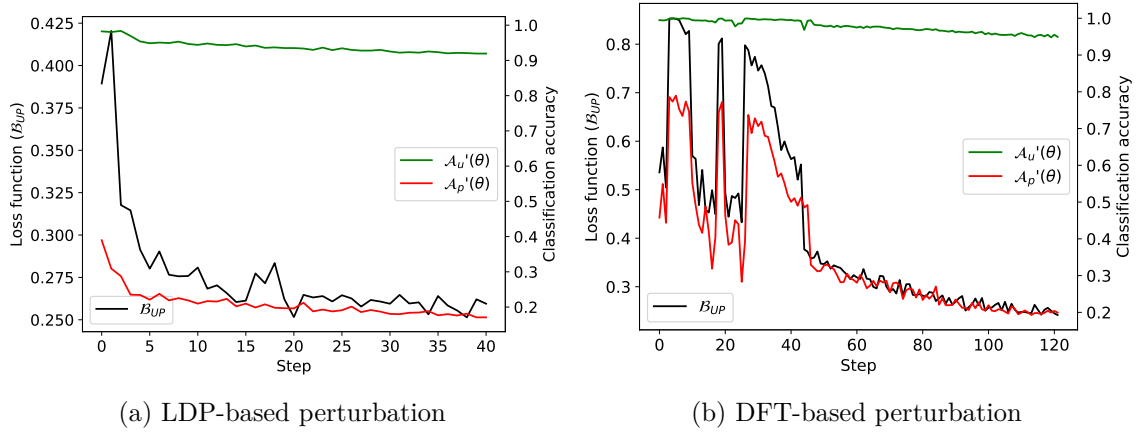
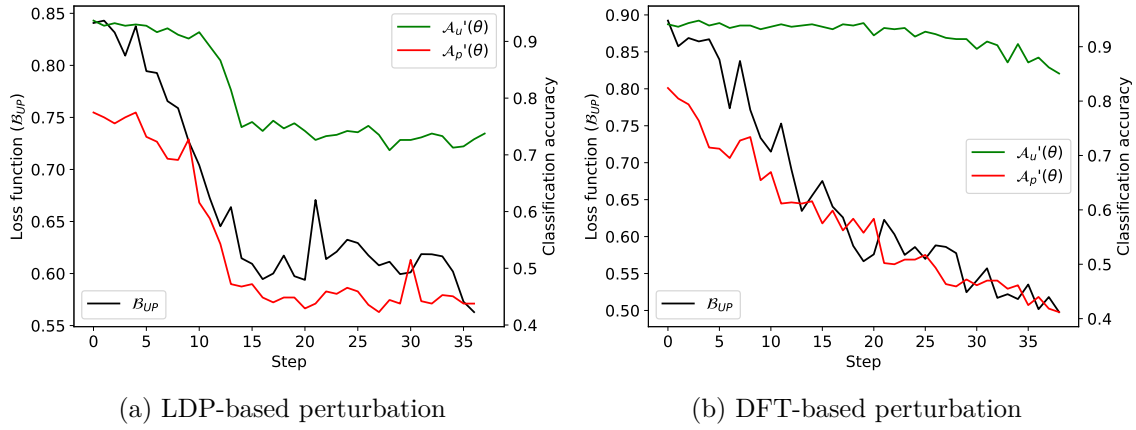
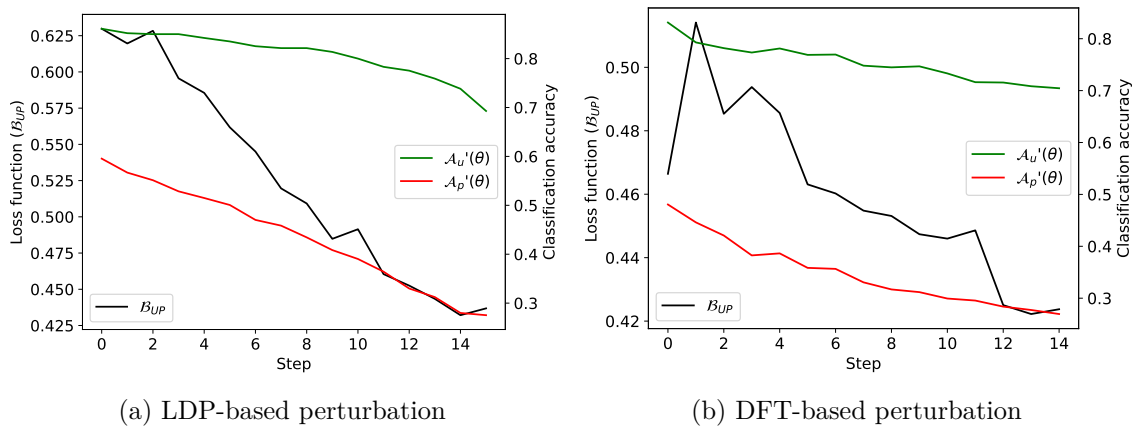
The approach enables determining parameter sets when the objective function depends on empirical metrics like classifier accuracy. This makes it well-suited for real-world machine learning applications where the analytic form of the objective function is not available. The implementation involves several iterative steps designed to refine parameters using results from measuring the performance of the PBC and UFC classifiers. The following procedure was applied:

- Initialize the parameters θ with values within the specified intervals, set the *learning rate* (η) and *convergence condition*.
- Train the PBC and UFC classifiers using the current θ and calculate $\mathcal{A}'_u(\theta)$ and $\mathcal{A}'_p(\theta)$.
- Derive the gradient vector $\nabla_{\theta}\mathcal{B}_{UP}(\theta)$ using Equation 4.23.
- Update θ using the gradient descent formula in Equation 4.26.
- Check the *convergence condition* by evaluating changes in θ .
- Repeat until the convergence criterion is satisfied.

Table 4.5 summarizes the parameters obtained from the optimization process and provides a comparison between these computed parameters and those determined empirically. Figures 4.22, 4.23, 4.24, 4.25, 4.26, and 4.27 illustrate the convergence of the score $\mathcal{B}_{UP}$ for each dataset under the tested perturbation approaches. Additionally, they illustrate the corresponding accuracies $\mathcal{A}'_p(\theta)$ and $\mathcal{A}'_u(\theta)$, providing insights into the performance dynamics across different methods.

Table 4.5 Results for the presented gradient descent-based parameter optimization approach.

Dataset	Classification scenario	Perturbation approach	Gradient parameters	descent	computed	Best parameters	empirically-identified parameters
			$\min(\mathcal{B}_{UP}) \theta^*$	# of steps		$\min(\mathcal{B}_{UP}) \theta^*$	
UAH	Driver-vs-road-type	LDP-based	0.2594	$\rho_I = 0.001, \alpha_p = 0.32, \alpha_{up} = 0.68, \epsilon_T = 34.5$	41	0.2975	$\rho_I = 0.005, \alpha_p = 0.15, \alpha_{up} = 0.85, \epsilon_T = 80$
		DFT-based	0.2417	$\rho_I = 0.005, \alpha_p = 0.98, \alpha_{up} = 0.02, \sigma_T = 104.8$	121	0.3293	$\rho_I = 0.005, \alpha_p = 0.7, \alpha_{up} = 0.3, \sigma_T = 180$
HCRL	Driver-vs-road-type	LDP-based	0.5626	$\rho_I = 0.002, \alpha_p = 0.10, \alpha_{up} = 0.90, \epsilon_T = 141$	35	0.5629	$\rho_I = 0.005, \alpha_p = 0.15, \alpha_{up} = 0.85, \epsilon_T = 120$
		DFT-based	0.4977	$\rho_I = 0.016, \alpha_p = 0.96, \alpha_{up} = 0.04, \sigma_T = 65.1$	38	0.5099	$\rho_I = 0.015, \alpha_p = 0.9, \alpha_{up} = 0.1, \sigma_T = 60$
UAH	Driver-vs-behavior	LDP-based	0.4368	$\rho_I = 0.002, \alpha_p = 0.33, \alpha_{up} = 0.67, \epsilon_T = 36.7$	15	0.4692	$\rho_I = 0, \alpha_p = 0.3, \alpha_{up} = 0.7, \epsilon_T = 50$
		DFT-based	0.4237	$\rho_I = 0.004, \alpha_p = 0.89, \alpha_{up} = 0.11, \sigma_T = 69.3$	13	0.5048	$\rho_I = 0, \alpha_p = 0.85, \alpha_{up} = 0.15, \sigma_T = 90$
MotionSense	User-vs-behavior	LDP-based	0.6072	$\rho_I = 0.001, \alpha_p = 0.14, \alpha_{up} = 0.86, \epsilon_T = 23.7$	34	0.6148	$\rho_I = 0, \alpha_p = 0.1, \alpha_{up} = 0.9, \epsilon_T = 25$
		DFT-based	0.6372	$\rho_I = 0.003, \alpha_p = 0.92, \alpha_{up} = 0.08, \sigma_T = 93.7$	31	0.6488	$\rho_I = 0.005, \alpha_p = 0.9, \alpha_{up} = 0.1, \sigma_T = 90$
PAMAP2 (31 features)	User-vs-behavior	LDP-based	0.7815	$\rho_I = 0.001, \alpha_p = 0.39, \alpha_{up} = 0.61, \epsilon_T = 1.98$	24	0.7969	$\rho_I = 0, \alpha_p = 0.3, \alpha_{up} = 0.7, \epsilon_T = 1.5$
		DFT-based	0.6897	$\rho_I = 0.011, \alpha_p = 0.63, \alpha_{up} = 0.37, \sigma_T = 188.3$	41	0.6921	$\rho_I = 0.01, \alpha_p = 0.7, \alpha_{up} = 0.3, \sigma_T = 180$
PAMAP2 (22 features)	User-vs-behavior	LDP-based	0.6856	$\rho_I = 0.018, \alpha_p = 0.38, \alpha_{up} = 0.62, \epsilon_T = 18.3$	32	0.6927	$\rho_I = 0.015, \alpha_p = 0.3, \alpha_{up} = 0.7, \epsilon_T = 15$
		DFT-based	0.5854	$\rho_I = 0.001, \alpha_p = 0.82, \alpha_{up} = 0.18, \sigma_T = 253.7$	34	0.5987	$\rho_I = 0, \alpha_p = 0.85, \alpha_{up} = 0.15, \sigma_T = 270$

Figure 4.22 Parameter optimisation for the UAH dataset (*driver-vs-road-type* classification).Figure 4.23 Parameter optimisation for the HCRL dataset (*driver-vs-road-type* classification).Figure 4.24 Parameter optimisation for the UAH dataset (*driver-vs-behavior* classification).

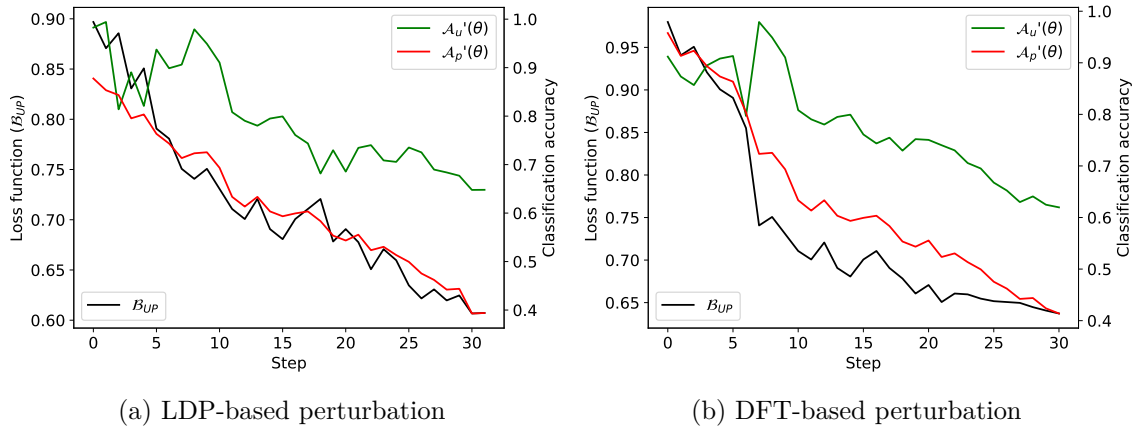


Figure 4.25 Parameter optimisation for the Motionsense dataset (*user-vs-behavior* classification).

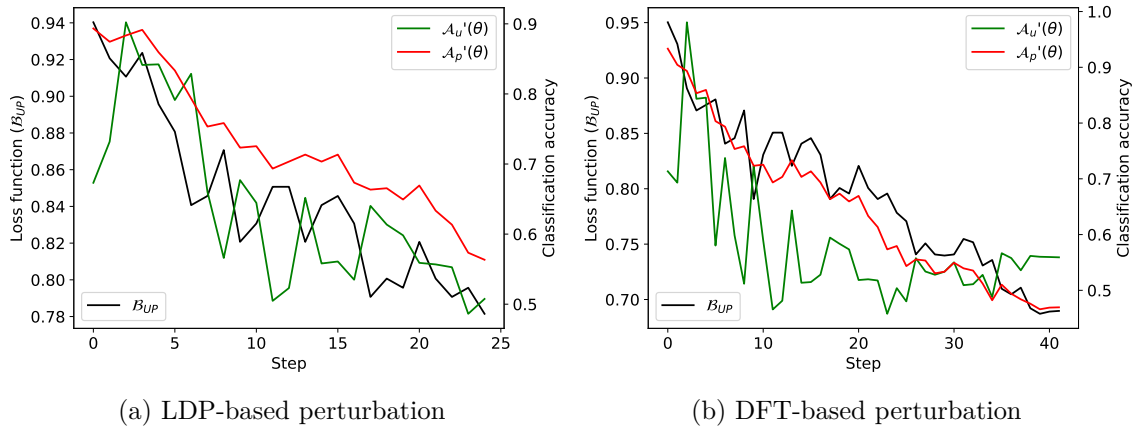


Figure 4.26 Parameter optimisation for the PAMAP2 dataset with 31 features (*user-vs-behavior* classification).

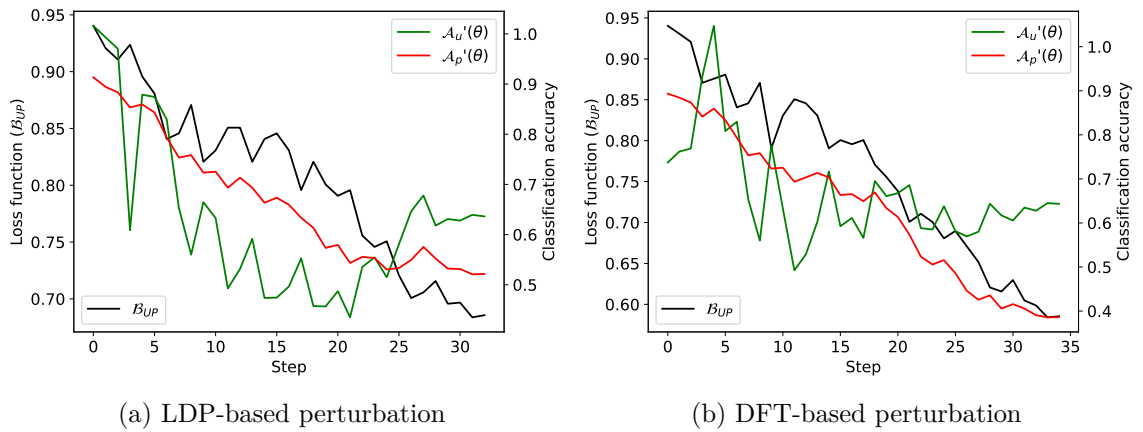


Figure 4.27 Parameter optimisation for the PAMAP2 dataset with 22 features (*user-vs-behavior* classification).

4.4 Comparison to the state-of-the-art time series protection approaches

Current time series protection mechanisms [77, 163, 132, 160, 75] are primarily designed for aggregate data publication (e.g., count, mean, frequency). However, in many real-world applications, raw time series data must be released for advanced tasks like classification or clustering, which require a more comprehensive analysis and manipulation of the data.

In the field of multivariate time series data classification, data privacy predominantly relies on two primary methodologies: (i) secure input data through diverse techniques such as perturbation, encryption, de-identification, data transformation, machine learning methodologies (e.g., using deep autoencoders) [63, 120, 28, 75, 38, 119]; and (ii) employing classification models that preserve privacy, thereby securing the training data against potential adversaries [122, 63].

A significant proportion of existing methods [122, 164, 71, 120, 38, 75, 119] fail to account for an adversarial model characterized by a powerful classifier. Moreover, very few approaches, such as those in [122, 75], adequately address the challenges of multivariate, continuously generated time series data. In contrast, other approaches are tailored for univariate data, with potential adaptations that apply the same methods to each feature individually. While machine learning models [122] provide some benefits, their effectiveness is limited as data is not protected locally. Moreover, our review indicates that existing perturbation-based time series privacy protection techniques [164, 71, 75] have not been evaluated for both privacy and utility using a dual-model approach, consisting of two opposing classifiers. Although Malekzadeh *et al.* [28] present a similar perspective on the issue, the suggested solution using a deep autoencoder falls short in considering the practical utility constraints imposed by data analysis and the capabilities of potential adversaries when establishing the size of data perturbation.

One innovative approach [77] to time series data privacy is the concept of w -event privacy. Introduced to fill the gap between event-level and user-level privacy models, w -event privacy protects groups of events within a sliding window of consecutive values. It tries to obtain a balance between the need for data usability and privacy concerns. Two methods, Budget Distribution, and Budget Absorption, were proposed to allocate privacy budgets efficiently across time series data, ensuring that the total privacy budget does not exceed a certain threshold. These methods dynamically adjust how much of the privacy budget is used based on prior data points, either gradually reducing the budget or redistributing unused portions for future data releases.

Another method, RescueDP [163], provides w -event privacy guarantees for infinite data streams through adaptive sampling and budget allocation techniques. RescueDP reduces privacy risks by limiting the number of users queried and processes data efficiently by grouping users based on location or other relevant factors. Similarly, the LDP-IDS [75] framework offers local differential privacy for infinite data streams. This framework addresses key challenges in

balancing privacy and utility, particularly the difficulty of accessing raw data for noise reduction and the utility loss caused by budget division techniques commonly used in differential privacy models.

In the context of wearable device data, LDP has also been explored. A collection scheme based on LDP [24] is proposed for single-attribute numerical stream data. This method first identifies salient data points from the stream and applies noise to these points using an adaptive Laplacian mechanism governed by a privacy budget. The collector then reconstructs the data curve by fitting it to the noise-added salient points, ensuring effective privacy protection while maintaining the utility of the data.

A separate method, Local Differential Privacy in the Temporal Setting (TLDP) [23], focuses on protecting the temporal order of time series data rather than the values themselves, introducing perturbations that alter the sequence without compromising key values. These efforts to balance privacy and utility represent an important direction in time series data protection.

Fang, Zeng, and Yang [132] introduced Local Differential Privacy Streaming (LDPS), a framework for data streams, using LDP and w -event privacy. The approach enables publishing statistical data over time by utilizing a sliding window technique for real-time data collection. It detects significant data changes, captures the most current trends in data distribution, and allocates the privacy budget accordingly.

Erlingsson et al. [160] proposed an approach for amplifying privacy in data collection using LDP models, leading to stronger CDP guarantees through a technique called *amplification by shuffling*. The method demonstrates how adding noise and shuffling in data collection can greatly enhance privacy protections beyond what is typically achieved using LDP protocols, thus reducing the overall privacy cost in industrial applications that anonymize reports.

Malekzadeh et al. [28] presented a similar perspective on the issue to the one proposed in this chapter, with two competing classifiers. However, the suggested solution using a deep autoencoder does not meet expectations in considering the practical utility constraints imposed by data analysis and the capabilities of potential adversaries when establishing the size of data perturbation.

In conclusion, most existing approaches to time series privacy protection focus on univariate data, with extensions for multiple features often being an afterthought. However, the method proposed in this chapter is specifically designed for multivariate data, distributing noise at the feature level to provide greater flexibility. Certain features may remain unperturbed, preserving utility while still protecting privacy. In the context of an adversarial model involving a powerful classifier, the approach provides stronger overall privacy compared to methods initially designed for univariate data and later adapted to multivariate scenarios. The results of this comparison, along with a detailed breakdown of the state-of-the-art techniques, are presented in Table 4.6.

Table 4.6 Comparison to the state-of-the-art data perturbation approaches for multivariate TSC.

Research	Data perturbation approach	Time series data type	Applicability	Adversarial model	Strengths
Kellaris and Papadopoulos [77]	Centralized DP	w-event Infinite, (sparse, binary data)	univariate Aggregate publication	data Linkage attacks	Introduces w -event privacy, focuses on continuous publication of statistics
RescueDP [163]	Centralized DP	w-event Infinite, (sparse, binary data)	univariate Aggregate publication (frequency of categorical data)	data NA	Incorporates perturbation and filtering techniques to mask individual user information
LDPS [132]	Local w-event DP	Finite, (dense, data)	univariate Aggregate publication	data NA	Allocates the privacy budget according to data changes and current trends in data distribution
Erlingsson et al. [160]	Local event-level DP	Infinite, (dense, data)	univariate Aggregate publication	data NA	Introduces the <i>amplification by shuffling</i> technique
LDP-IDS [75]	Local w-event DP	Infinite, (sparse, binary data)	univariate Aggregate publication	data NA	Secures the data locally and releases an estimated histogram of data at the server level
Li et al. [24]	Local user-level DP	Finite, (dense, data)	univariate Time series release	data NA	Introduces a data collection scheme using LDP, specifically designed for wearable devices
LoPub [27]	Local event-level DP	Infinite, (dense data)	multivariate Time series release	data Classification	Identifies attribute correlations to reduce data dimensionality
Malekzadeh et al. [28]	Autoencoder perturbation	Infinite, (dense, data)	multivariate Time series release	data Privacy-breaking classifier	Uses a pre-trained autoencoder for data perturbation
DP-BUP (presented in this chapter)	Local w-event DFT-based	DP, Infinite, (dense data, non-binary data)	multivariate Time series release	data Privacy-breaking classifier	Built specifically for multivariate time series data, applies perturbation based on feature clustering, enables the usage of existing univariate perturbation methods

4.5 Summary

The DP-BUP approach for clustering features based on their importance coefficients offers an effective solution for protecting multivariate sensor data, producing better results than uniformly distributing noise across all features. The experimental assessment confirmed that it is possible to identify values for the used parameters that ensure the resulting perturbations meet the objective outlined in Equation 4.3. Key advantages of this method include a lower $\mathcal{A}'_p$ (PBC accuracy), a higher $\mathcal{A}'_u$ (UFC accuracy), and increased utility as measured by MAE. Additional benefits are: (i) certain features, specifically those in $\mathcal{F}_u$ that are important for utility classification, remain unperturbed, thereby preserving the utility for other data processing tasks beyond classification; and (ii) the perturbation budget is significantly higher, resulting in lower noise, when clustering is employed to achieve comparable privacy and utility objectives.

The BUP, $\mathcal{B}_{UP}$, which provides a detailed assessment of the privacy and utility achieved by the selected parameter set, is introduced. This score maximizes the distance between $\mathcal{A}'_u(\theta)$ and $\mathcal{A}'_p(\theta)$, effectively balancing the trade-offs. However, alternative perturbation parameters may be selected if the accuracy of *PBC* or *UFC* is prioritized. Depending on specific requirements, such as maximizing the classification accuracy for *PBC* or maintaining the classification accuracy for *UFC* below a predetermined threshold, different perturbation parameters can be chosen.

Regarding privacy guarantees, the w -event LDP mechanism ensures data indistinguishability [159]. Consequently, within any batch of w records transmitted to the centralized location, it is improbable to discern any precise event or specific value. For example, an attacker might identify a driver in traffic at a particular moment t with a specific speed s , but they cannot infer the data exclusively associated with that driver stored in the data warehouse solely from the information available to them. In the context of classification, the w -event LDP mechanism, as well as the DFT-based perturbation, provide limited privacy guarantees due to the inherent characteristics of the classification task. Since at least two classifiers (UFC and PBC) are assessed, each with opposing objectives, the perturbation process must obscure specific features essential for user classification while retaining other attributes relevant to non-user classification, including inter-feature correlations. Nonetheless, in the presented scenario, the privacy level is assured by minimizing the accuracy $\mathcal{A}'_p(\theta)$ of the *PBC*.

The conducted experiments on driving and HAR datasets validated the presented approach and determined appropriate values for perturbation size (ϵ_T and σ_T), for the clustering parameter ρ_I , and for the cluster privacy budget distribution parameters α_p and α_{up} . Furthermore, experiments showed that it is possible to balance privacy and utility effectively in scenarios involving classifiers for time series data. Moreover, the case of an advanced adversarial model where the attacker has access to both perturbed data and corresponding driver/user annotations is explored. The experimental results demonstrated

that this model achieves improved classification accuracy, suggesting that further research is needed to develop more robust privacy protection mechanisms.

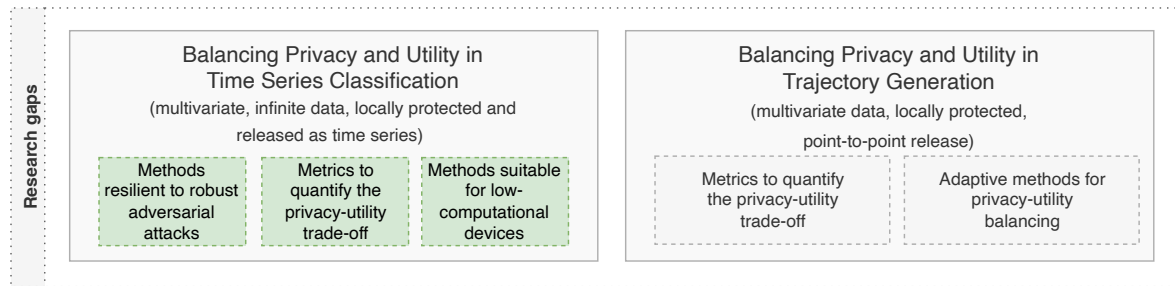


Figure 4.28 The research gaps targeted by the DP-BUP method (highlighted in green).

Further research is necessary to explore generalizing the feature selection and feature importance computation for perturbation across different scenarios. Additionally, it is essential to investigate and compare various perturbation methods to identify the most suitable ones for different types of data and classification tasks, including the classification of perturbed data. The possibility of using multilabel classifiers and scenarios involving multiple PBCs and UFCs should also be investigated. Conducting additional tests on diverse datasets and various types of data will provide a comprehensive evaluation of the effectiveness and robustness of the presented approach in different settings.

In conclusion, the DP-BUP method effectively addresses the identified gaps (Figure 4.28) in achieving a balance between data privacy and utility for protected time series data in the context of TSC. Furthermore, the method demonstrates resilience against robust adversarial attacks and, due to the low resource requirements of perturbation techniques, can be efficiently deployed on devices with limited computational capabilities.

Chapter 5

Balancing Privacy and Utility in Real-Time Location Data Release

This chapter is based on the work presented in the following article:

- **Adrian-Silviu Roman**, Roland Bolboacă, Sophie Cerf, Piroška Haller, and Bogdan Robu. “APUTrajGen: Adaptive Privacy and Utility Preserving Real-Time Synthetic Trajectory Generation”, 2025. [[165](#)]

This chapter presents a novel methodology for protecting location time series data through the synthesis of values intended for public release. In time series data, observations are recorded sequentially over time. Similarly, trajectory data records the path or movement of an object, usually over time and space, with timestamps marking each position. This spatial-temporal nature of trajectory data makes it a specific type of multivariate time series data, where each entry includes both spatial (e.g., latitude, longitude) and temporal (timestamps) components. Location privacy and trajectory data introduce unique privacy challenges. Techniques such as spatial perturbation, synthetic trajectory generation, and Differential Privacy (DP)-based approaches are available. However, the identified research gaps (Figure 2.4) highlight that synthetic data generation methods need improvement to balance the privacy and utility of the resulting trajectory data.

The Adaptive Privacy and Utility Preserving Real-Time Synthetic Trajectory Generation (APU-TrajGen) method is a technique for generating synthetic trajectories that closely resemble real ones. The creation of synthetic trajectories is recognized as a viable approach to safeguarding the privacy of human mobility trajectory data [166]. While the developed method is applicable to any time series data, the focus in this chapter is on protecting location data. Additionally, to balance data privacy with utility, the APU-TrajGen approach presents adaptive perturbation. The mechanism generates data that meets specific utility and privacy objectives. Thus, the synthesized data remains useful for various applications.

5.1 Problem statement and research gaps

Achieving an optimal balance between data privacy and utility within desired parameters remains a complex challenge, as these methods can be difficult to configure appropriately. Moreover, releasing privacy-protecting synthetic trajectories in real-time brings additional challenges as they may not result in continuous trajectories [144] or data collected by a trusted server may still lead to individual user's data leakage during the trajectory data release phase [167].

In this chapter, three specific research queries were investigated:

- How can privacy-protecting synthetic trajectories be generated and released (point-to-point) in real-time to achieve an optimal balance between privacy and utility at specified levels?
- How can this be achieved with low computational load and run in a local environment?
- To what extent do the generated trajectories retain their overall utility?
- How effective is the presented model in safeguarding the privacy of real trajectories?

The existing synthetic trajectory generation approaches with the objective of privacy preservation lack the possibility of real-time point-to-point release. Moreover, they do not

provide mechanisms for balancing the privacy and utility of the requested trajectories taking into account targeted restrictions. In the context of identified research gaps, we propose the following:

- A method for synthesizing or generating trajectories that closely resemble real-world data while maintaining low computational overhead for real-time point-to-point trajectory release.
- A mechanism that effectively balances privacy and utility in location data by adapting protections based on point-to-point or trajectory-to-trajectory metrics.
- A real-time privacy protection mechanism that can be implemented at the device level, eliminating the need for a central trustworthy aggregator and reducing the risk of data breaches from untrustworthy data curators.

Use case: Optimizing taxi station placement. Taxi companies use mobility patterns and trajectory analysis to optimize their services. Objectives may include: (i) optimizing the placement of taxi stations to increase availability and minimize passenger wait times; (ii) identifying the most frequently traveled routes; (iii) detecting anomalies in driver behavior. However, using real-world trajectory data poses privacy risks, as it may contain sensitive location information about individuals' travel habits. The APU-TrajGen framework can be deployed in taxis to provide a privacy-preserving solution by generating synthetic trajectories that are sent to centralized databases for analysis. By training an LSTM model without relying on user-annotated trajectory data, realistic trajectory points are generated. To balance privacy and utility, the method adjusts the granularity of synthetic data based on the distance between real and generated trajectories. The synthetic trajectory data, including start points, stop points, and intersections, can support various mobility analysis applications. For example, identifying areas with a high concentration of starting points can help optimize taxi station placement, improving service efficiency and availability. Additional use cases include optimizing fleet distribution and enhancing urban mobility planning through collaboration with city authorities.

5.2 The APU-TrajGen method

In this section, the APU-TrajGen approach for generating synthetic trajectories for real-time, point-to-point release, is presented, considering the case when only spatial coordinates (latitude, longitude) and timestamps are collected. Importantly, this method does not require prior domain knowledge of human mobility [144] and does not require any user annotated data.

Definition 7. (Trajectory) [97] *A trajectory is defined as a sequence of τ spatial points in chronological order, $\mathcal{T} = (r_1, r_2, \dots, r_p)$, such that each object has a unique trajectory that encapsulates its movement history.*

The objective is to generate a protected trajectory $\mathcal{T}_p = (p_1, p_2, \dots, p_p)$ from the real trajectory $\mathcal{T}_r = (r_1, r_2, \dots, r_p)$ such that $\mathcal{T}_p$ meets specified objectives for both utility and privacy.

5.2.1 LSTM model-based trajectory generation

The proposed trajectory forecasting method is based on an Long Short-Term Memory (LSTM) model with a simplified architecture, with a single hidden layer. LSTM models, introduced by Hochreiter and Schmidhuber [168], were developed to overcome the vanishing and exploding gradient problem inherent in traditional Recurrent Neural Networks (RNNs). Over time, LSTMs have gained widespread adoption and are now a preferred approach for sequence modeling tasks, particularly in time series forecasting.

LSTM models incorporate memory units capable of capturing both short-term and long-term dependencies within time series data. A standard LSTM layer consists of multiple blocks, each containing memory cells and three types of gates: input gate, output gate, and forget gate. Additionally, each block maintains two recurrent connections: the hidden state, which represents short-term memory, and the cell state, which stores long-term memory.

The equations for LSTM [169] at time t are presented below:

$$\mathbf{f}(t) = \text{sigm}(\mathcal{W}_f \mathbf{X}(t) + \mathcal{U}_f \mathbf{h}(t-1) + \mathbf{b}_f), \quad (5.1)$$

$$\mathbf{I}(t) = \text{sigm}(\mathcal{W}_I \mathbf{X}(t) + \mathcal{U}_I \mathbf{h}(t-1) + \mathbf{b}_I), \quad (5.2)$$

$$\overline{\mathbf{C}}(t) = \text{tanh}(\mathcal{W}_C \mathbf{X}(t) + \mathcal{U}_C \mathbf{h}(t-1) + \mathbf{b}_C), \quad (5.3)$$

$$\mathbf{C}(t) = \mathbf{f}(t) \cdot \mathbf{C}(t-1) + \mathbf{I}(t) \cdot \overline{\mathbf{C}}(t), \quad (5.4)$$

$$\mathbf{o}(t) = \text{sigm}(\mathcal{W}_o \mathbf{X}(t) + \mathcal{U}_o \mathbf{h}(t-1) + \mathbf{b}_o), \quad (5.5)$$

$$\mathbf{h}(t) = \mathbf{o}(t) \cdot \text{tanh}(\mathbf{C}(t)), \quad (5.6)$$

$$\hat{y}(t) = \mathcal{P}_{\hat{y}} \mathbf{h}(t) + b_{\hat{y}}. \quad (5.7)$$

where $\mathcal{W}, \mathcal{V}, \mathcal{U}$ and $\mathcal{P}$ denote the weight matrices, $\mathbf{X}$ is the input vector, $\hat{y}(t)$ is the predicted output, $\overline{\mathbf{C}}$ denotes the vector of new candidates for the cell state, $\mathbf{C}$ is the current cell state, $\mathbf{h}$ is the hidden state and b denotes the bias vectors.

In the case of trajectory data, the model is trained in a many-to-many configuration, meaning that for an input sequence of τ observations (e.g., spatial coordinates), τ corresponding predictions are generated as output. During inference, the model operates in two modes depending on a parameter denoted k . For $k = 1$, it performs a single-step prediction, using each observed value as input. For $k > 1$, the model generates predictions iteratively, taking one time-step at a time as input and predicting one time-step ahead (Equations 5.8, 5.9, and 5.10). Additionally, after every k prediction within a trajectory, the model $\mathcal{M}$ reinitializes

with the observed value and continues predicting the following k steps using the preceding $k - 1$ predictions as input.

$$p_{i+1} = \mathcal{M}(r_i; \mathbf{h}_{i-1}; \mathbf{C}_{i-1}), \quad (5.8)$$

$$p_{i+2} = \mathcal{M}(p_{i+1}; \mathbf{h}_i; \mathbf{C}_i) = \mathcal{M}(\mathcal{M}(r_i; \mathbf{h}_{i-1}; \mathbf{C}_{i-1}); \mathbf{h}_i; \mathbf{C}_i), \quad (5.9)$$

$$p_{i+k} = \underbrace{\mathcal{M}(\mathcal{M}(\dots \mathcal{M}(r_i; \mathbf{h}_{i-1}; \mathbf{C}_{i-1}); \mathbf{h}_i, \mathbf{C}_i) \dots; \mathbf{h}_{i+k-3}, \mathbf{C}_{i+k-3})}_{k \text{ times}}; \mathbf{h}_{i+k-2}, \mathbf{C}_{i+k-2}). \quad (5.10)$$

The complete k times architecture and hyperparameters of the LSTM model are as follows: a single LSTM hidden layer composed of 16 hidden units, two features (e.g., trajectory spatial coordinates), and two outputs (e.g., the next coordinates of the trajectory), a learning rate of 0.01, a training sequence length and batch size of 16, 100 training epochs, and a learning rate decay of 0.99 every 50 epochs. The model is trained using stochastic gradient descent, utilizing the Mean Squared Error (MSE) loss function. The remainder of the hyperparameters are utilized with their respective default values. During training, the long- and short-term memory values are reset after each sequence, and during inference, these state values are reset after each trajectory.

5.2.2 Adaptive trajectory data privacy

Let r_i be the tuple containing the longitude and latitude of the actual location for the i -th measurement, and p_{i+1} denote the protected location predicted from r_i . In Equation 5.10, k is the number of locations the defined LSTM model $\mathcal{M}$ is set up to predict starting from a given real location r_i .

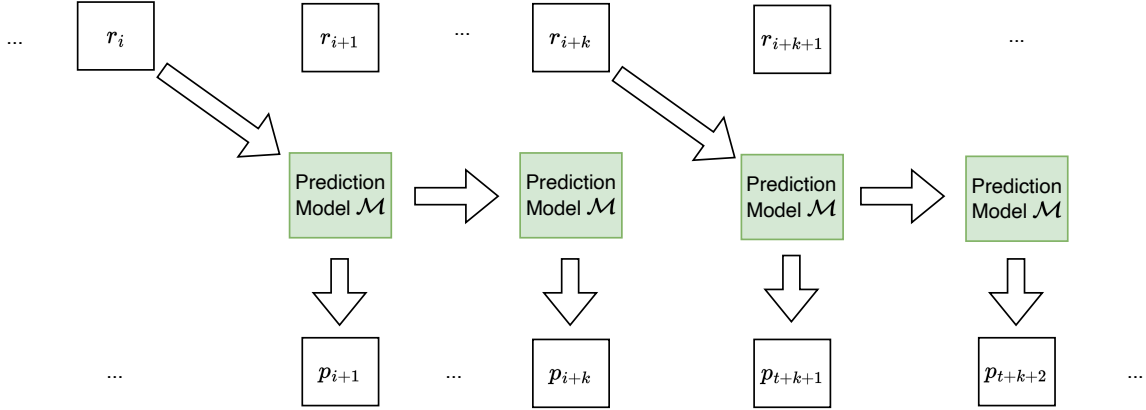


Figure 5.1 The usage of the LSTM-based model to generate protected points.

For the adaptive trajectory protection, a variable value for k in order to initialize the prediction according to Equation 5.8 at desired times is introduced. Thus, at time $i + 1$ generate p_{i+1} from r_i , instead of computing p_{i+1} from p_i (Figure 5.1). Let Utility-Privacy Score (SUP) score, denoted $\mathcal{S}_{UP}$, that measures the instant utility and privacy of the trajectory, and adjust k dynamically to obtain a trajectory that better fits the criteria imposed by the

defined score (Figure 5.2). The assumptions are that a lower k value brings the computed point closer to the real point, thereby enhancing utility, while a higher k value increases the distance between the computed and real points, thereby improving privacy.

The generalized form of $\mathcal{S}_{UP}$ is defined as follows:

$$\mathcal{S}_{UP}(t_i) = \prod_{j=1}^n \frac{d_j(\mathcal{T}_r(t_i), \mathcal{T}_p(t_i)) - a_j}{b_j - a_j}, \quad (5.11)$$

where $\mathcal{T}_r(t_i)$ and $\mathcal{T}_p(t_i)$ denote the real and protected trajectories collected and released up to time t_i ; d_j is a distance metric used to measure the discrepancy between the real and predicted/reported trajectories or as a point-to-point metric, n represents the number of distance metrics considered, and a_j and b_j define the minimum and maximum desired values for d_j , with $d_j \in [a_j, b_j]$.

In the context of this research, $\mathcal{S}_{UP}$ is computed using a single distance metric, specifically the distance between the protected point p_i and the corresponding real location r_i :

$$\mathcal{S}_{UP}(t_i) = \frac{d(r_i, p_i) - \mu_{min}}{\mu_{max} - \mu_{min}}, \quad (5.12)$$

where $d(r_i, p_i)$ is the distance between the computed point p_{i+1} and the real location r_i , μ_{min} and μ_{max} are the targeted minimum and maximum mean distance between the real and protected points. Values μ_{min} and μ_{max} are constraints directly related to the privacy and the utility of the resulted trajectories.

The algorithm starts at $k = 1$. At every point t_i , $\mathcal{S}_{UP}(t_i)$ is computed, and, next value of k is established by taking into account the value of $\mathcal{S}_{UP}(t_i)$:

$$k = \begin{cases} 1, & \text{if } \mathcal{S}_{UP}(t_i) > 1 \text{ and } k = 1, \\ k, & \text{if } \mathcal{S}_{UP}(t_i) \in [0, 1], \\ k + 1, & \text{if } \mathcal{S}_{UP}(t_i) < 0, \\ k - 1, & \text{if } \mathcal{S}_{UP}(t_i) > 1. \end{cases} \quad (5.13)$$

Further, the Equation 5.10 determines the value of p_{i+1} by taking into account the computed value of k .

5.2.3 Privacy and utility evaluation criteria

The performance of the presented model was evaluated using well-established utility measures from previous studies [95, 170, 96], applying these to the Porto dataset used in the experiments. Furthermore, a robust adversarial model designed for trajectory reconstruction is investigated.

5.2.3.1 Utility metrics

The utility metrics are divided into three categories: global-level, trajectory-level, and semantic-level metrics. Global-level metrics, such as *Density error*, *Relative query error*,

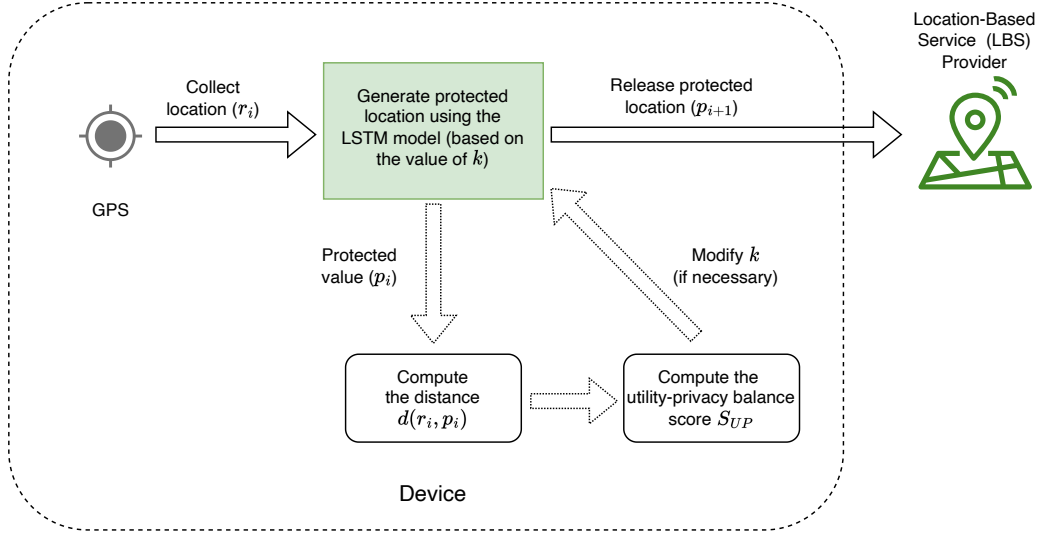


Figure 5.2 The Adaptive Privacy and Utility Preserving Real-Time Synthetic Trajectory Generation (APU-TrajGen) trajectory generation approach.

Hotspot query error, and *Kendall-Tau* coefficient, assess the overall preservation of spatial and density patterns between real and protected datasets, as well as the maintenance of high-density areas (hotspots) [95].

Trajectory-level metrics evaluate individual trajectory characteristics and alignment. Metrics like *Trip error*, *Length error*, and *Diameter error* measure the fidelity of starting and ending points, trajectory lengths, and trajectory spread [96, 97, 98]. Additionally, trajectory similarity is captured through micro-level metrics such as *Hausdorff distance* [99], *Discrete Wavelet Transform (DWT)* [100], and *Edit Distance on Real Sequences (EDR)* [101], providing a detailed assessment of alignment between real and protected individual trajectories.

Semantic-level metrics focus on frequently observed movement patterns in the data. Metrics such as *Pattern F1* and *Pattern error* evaluate the consistency of common travel sequences, ensuring that patterns frequently occurring in real trajectories are similarly represented in the protected data [95]. These metrics offer a complete view of data utility across different levels of trajectory analysis.

To evaluate the capabilities of the adaptive approach to achieve the desired balance between utility and privacy, the average Haversine distance between the protected trajectories $\mathcal{T}_p$ and the trajectories $\mathcal{T}_r$ is computed:

$$\bar{D}(\mathcal{T}_p, \mathcal{T}_r) = \frac{1}{n} \sum_{i=1}^n \bar{d}(\mathcal{T}_{p_i}, \mathcal{T}_{r_i}), \quad (5.14)$$

where n is the number of trajectories, and $\bar{d}(\mathcal{T}_{p_i}, \mathcal{T}_{r_i})$ is the average Haversine pairwise distance between trajectories $\mathcal{T}_{p_i}$ and $\mathcal{T}_{r_i}$.

Additionally, the standard deviation $\sigma_{[\mu_{min}, \mu_{max}]}$ is computed, to measure the dispersion of the average Haversine pairwise distances, $\bar{d}(\mathcal{T}_{p_i}, \mathcal{T}_{r_i})$, relative to the desired interval $[\mu_{min}, \mu_{max}]$.

By calculating the standard deviation, we assess how consistently the distances align with the desired range. A smaller standard deviation shows that the distances are clustered around their mean, suggesting better control over the variability of the generated trajectories. A higher standard deviation indicates greater variability, implying that the distances deviate significantly, making it harder to ensure they remain within the target bounds.

Further, the proportion of times the resulted average Haversine pairwise distance $\bar{d}(\mathcal{T}_{pi}, \mathcal{T}_{ri})$ is in the desired interval (Equation 5.15) is computed. It measures the consistency of the generated trajectories in meeting the specified utility and privacy constraints. A higher proportion indicates that the majority of the distances are within the target range. Conversely, a lower proportion suggests that the distances frequently fall outside the acceptable bounds.

$$P_{[\mu_{min}, \mu_{max}]} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(\mu_{min} \leq \bar{d}(\mathcal{T}_{pi}, \mathcal{T}_{ri}) \leq \mu_{max}), \quad (5.15)$$

where $\mathbb{I}(c)$ is the binary indicator function returning 1 iff the condition c is true, and returns 0 if the condition c is false.

5.2.3.2 The adversarial model

Generally, the privacy of a generated trajectory is defined by the pairwise distance between the original and the protected trajectories [146]. A larger distance indicates a higher level of privacy. Since the model generates trajectories to meet a specified protection level (μ_{min}), the conclusion is that the presented model effectively provides the required protection.

Additionally, the following threat model is investigated. Consider that potential attackers have access to the trained model $\mathcal{M}$, to the set of protected points $\mathcal{T}_p$ and all the parameters (e.g., μ_{min} , μ_{max}) used by the model $\mathcal{M}$. The attacker's objective is to determine a trajectory $\mathcal{T}^*$ close to the real one $\mathcal{T}_r$. The success of the attacking model is quantified by computing the *reconstruction error* [171] defined as:

$$E_R = |\bar{D}(\mathcal{T}^*, \mathcal{T}_r) - \bar{D}(\mathcal{T}_p, \mathcal{T}_r)|. \quad (5.16)$$

5.3 Experimental evaluation

To validate the developed approach, the Porto Taxi dataset [172] was selected and the results were benchmarked against state-of-the-art synthetic trajectory generation methods that have demonstrated strong performance in this dataset.

5.3.1 The location dataset

The Porto Taxi dataset contains 1,710,671 measured GPS trajectories from taxis recorded from 2013/07/01 to 2014/06/30, in Porto, Portugal. Each trajectory includes the trajectory as a series of coordinates, the taxi id, the trajectory id, and a timestamp. Each taxi has multiple

trajectories, each of which is specified by a unique identifier. The coordinates are represented in the POINT (longitude, latitude) format, following the WGS 84 geodetic coordinate system (EPSG: 4326). [173]. The GPS coordinates are sampled at a fixed interval of 15 seconds.

From the dataset, a total of 70% of trajectories is utilized for model training, while the remainder of 30% is utilized for testing. The dataset is normalized using a min/max scaler in the $[0, 1]$ range. The minimum and maximum values for normalization are selected from a specific region of the map. In this scenario, as the outlines of Porto. To facilitate the applications of the privacy mechanisms, post-model predictions are denormalized and restored to the original scales. Each trajectory is trimmed to the furthest multiple of τ to create the training subset. During training, the values of the sequence length of the LSTM model are set to be equal to τ . For model testing, the data points are normalized using the same scaler as above, and no trimming is performed. The testing procedure includes point-by-point and multi-step-ahead forecasts, depending on the current value of k . The first trajectory point is predicted after m steps.

5.3.2 Experiments, results, and discussion

Firstly, the trained model $\mathcal{M}$ was tested with various fixed values of k . Figure 5.3 shows the real trajectory and the computed one for various values of k when the model is reset to every k step. The Haversine distance [174] between the real location r_i and the computed one is shown at each location point.

Table 5.1 presents the results of the utility metrics described in Section 5.2.3.1, comparing them with the best outcomes from state-of-the-art DP-based methods [95, 105] (using $\epsilon = 1.5$, representing the lowest perturbation and highest utility) and the TS-TrajGen approach [144], all evaluated on the Porto dataset. The results indicate that the presented method achieves comparable or superior utility metric values compared to other trajectory generation methods.

Further, the adaptive trajectory generation method detailed in Section 5.2.2, designed to optimize the balance between data utility and privacy, was evaluated. This method establishes the desired privacy-utility trade-off by evaluating the mean distances between corresponding data points (r_i – the original location, and p_i – the protected location). The values μ_{min} and μ_{max} represent the minimum and maximum targeted mean distances between points in the original and protected trajectories. At each collected location, the proposed method computes the mean distance between the points of the real and the protected trajectories and adjusts the parameter k accordingly. As shown in Figure 5.4, the presented approach successfully achieves the intended balance between privacy and utility, as defined by the mean distance between corresponding points. For each trajectory, the distance between points and the value of k are illustrated.

The results presented in Table 5.2 demonstrate that the proposed method yields impressive results for the various tested values of μ_{min} and μ_{max} , with a normal distribution of the distances between $\mathcal{T}_r$ and $\mathcal{T}_p$ (Fig. 5.5). For example, when the desired mean distance between the points of the original and protected trajectories is set between 50 and 150 meters, 87% of

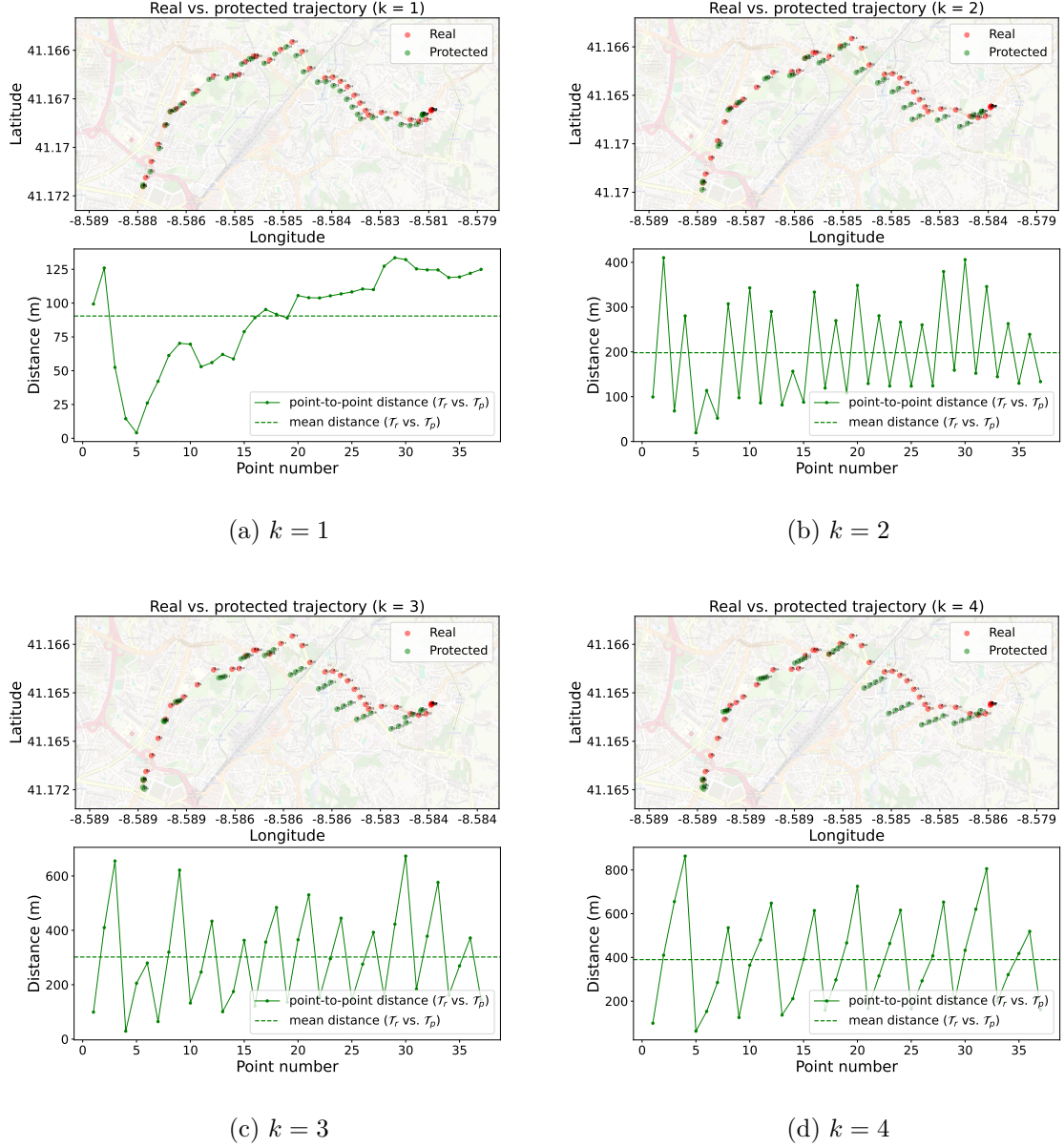


Figure 5.3 Examples of trajectory protection for trajectory id = 1372859766620000467.

the generated trajectories achieve a mean distance within this target range, with $\bar{D} = 120.87$ meters. The computed standard deviations show that distances consistently align with the desired range with acceptable values around the mean interval ($\sigma_{[\mu_{min}, \mu_{max}]}$ increases as the privacy increases). However, while $P_{[\mu_{min}, \mu_{max}]}$ decreases under stronger privacy parameters, the majority of distances still fall within the desired interval, maintaining the balance between privacy and utility.

The robustness of the proposed approach was evaluated against adversarial attacks by simulating the reconstruction of real trajectories $\mathcal{T}_r$ from the published synthetic trajectories

Table 5.1 Computed utility metrics for fixed k on 10,000 points and 207 trajectories.

Metric	APU-TrajGen method					LDPTrace[95], NGRAM[105], TS-TrajGen[144]
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	
Density Error	0.0011	0.0040	0.0523	0.0424	0.0343	0.0069
Relative Query Error	0.0281	0.0617	0.0758	0.0995	0.1042	0.3011
Hotspot Query Error	0	0.0068	0.1645	0.1359	0.0069	0
Kendall-tau	0.8968	0.8873	0.8873	0.8936	0.9000	0.5934
Trip Error	0.0024	0.0111	0.0156	0.0227	0.0191	0.0771
Length Error	0.0045	0.0319	0.0278	0.0275	0.0243	0.0392
Diameter Error	0.0040	0.0050	0.0075	0.0064	0.0085	0.0331
Pattern F1	0.7600	0.6500	0.5200	0.5200	0.5300	0.1900
Pattern Error	0.2629	0.4830	0.5698	0.5620	0.5245	0.6692
Hausdorff	0.1593	0.2577	0.3198	0.4339	0.5255	0.6500
DTW	2.8997	5.2124	7.1777	9.0476	10.795	18.120
EDR	0.0193	0.1304	0.4734	1.000	1.8309	0.5580

Table 5.2 Computed utility metrics for the adaptive approach on 10,000 points and 207 trajectories.

Metric	APU-TrajGen method				LDPTrace[95], NGRAM[105], TS-TrajGen[144]
	$\mu_{min} = 50$ $\mu_{max} = 150$	$\mu_{min} = 100$ $\mu_{max} = 250$	$\mu_{min} = 200$ $\mu_{max} = 350$	$\mu_{min} = 300$ $\mu_{max} = 450$	
Density Error	0.0012	0.0011	0.0263	0.0346	0.0069
Relative Query Error	0.0259	0.0530	0.0851	0.0915	0.3011
Hotspot Query Error	0.0034	0.0034	0.0090	0.0090	0
Kendall-tau	0.9000	0.8904	0.8968	0.7666	0.5934
Trip Error	0.0021	0.0091	0.0152	0.0330	0.0771
Length Error	0.0148	0.0257	0.0132	0.0192	0.0392
Diameter Error	0.0031	0.0062	0.0097	0.0123	0.0331
Pattern F1	0.7299	0.7200	0.5700	0.5500	0.1900
Pattern Error	0.3386	0.3462	0.4613	0.5403	0.6692
Hausdorff	0.3097	0.4484	0.6937	0.8072	0.6500
DTW	4.9609	7.8419	12.286	15.894	18.120
EDR	0.1739	0.9227	2.6618	4.9806	0.5580
$\bar{D}(\mathcal{T}_p, \mathcal{T}_r)$ (m)	120.87	217.98	336.94	432.94	-
$P_{[\mu_{min}, \mu_{max}]}$ (%)	0.8743	0.8067	0.6086	0.5555	-
$\sigma_{[\mu_{min}, \mu_{max}]}$	38.524	80.736	114.21	140.02	-

$\mathcal{T}_p$ using the same model $\mathcal{M}$. In this scenario, attackers are assumed to have full access to the model $\mathcal{M}$, the published synthetic trajectories $\mathcal{T}_p$, and the parameters (μ_{min}, μ_{max}) used during the generation process. The objective of the attackers is to reconstruct a trajectory $\mathcal{T}^*$ that closely approximates the real trajectory $\mathcal{T}_r$. The results, presented in Table 5.3, demonstrate that the mean distance between the reconstructed trajectories $\mathcal{T}^*$ and the real trajectories $\mathcal{T}_r$ remains substantially large, indicating a strong level of privacy protection. Additionally, the

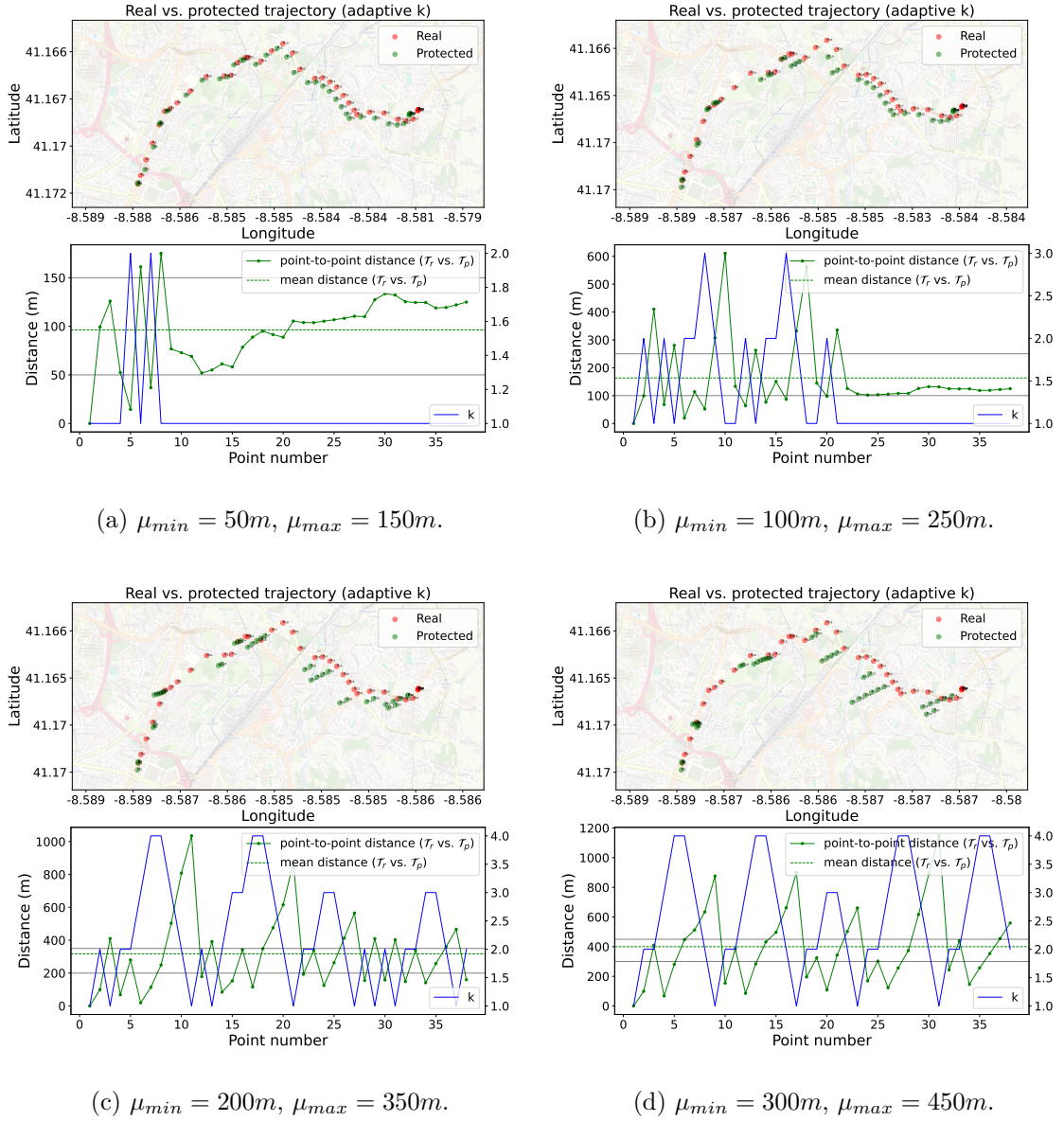


Figure 5.4 Examples of adaptive trajectory protection for trajectory id = 1372859766620000467.

reconstruction error E_R , which measures the discrepancy between the attacker's generated trajectory and the real trajectory, is also consistently high. These findings highlight the robustness of the proposed mechanism in preventing the accurate reconstruction of sensitive trajectory data.

Figure 5.6 further illustrates the trajectories generated by the attacker under varying levels of privacy protection defined by the parameters (μ_{min}, μ_{max}) . It shows that the reconstructed trajectories deviate significantly from the original ones, even when attackers have access to the generating model and detailed knowledge of the privacy mechanism. By dynamically adjusting

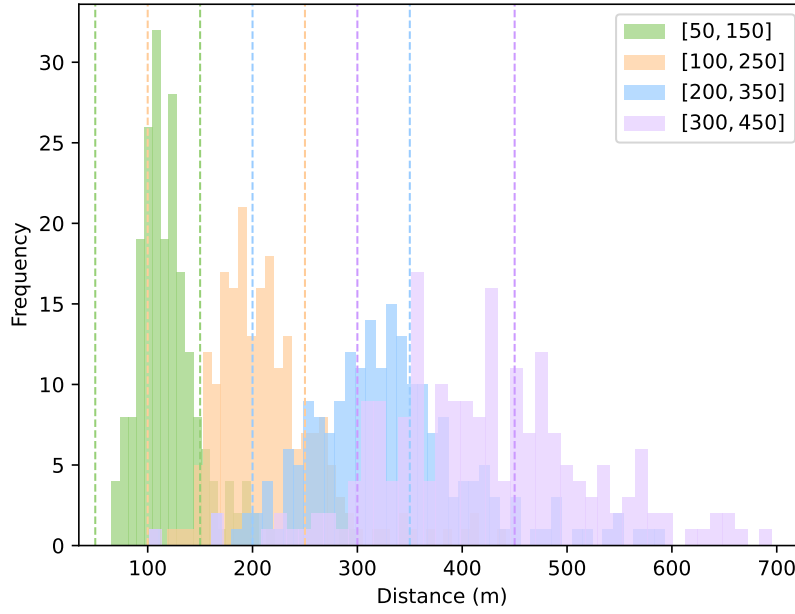


Figure 5.5 The distribution of the distances between $\mathcal{T}_r$ and $\mathcal{T}_p$ for various targeted $[\mu_{min}, \mu_{max}]$.

the generation of synthetic points based on the utility-privacy trade-off, the mechanism creates trajectories that retain their analytical value while safeguarding sensitive information.

Table 5.3 Attack results for the adaptive approach on 10,000 points and 207 trajectories.

Metric	Parameters			
	$\mu_{min} = 50$ $\mu_{max} = 150$	$\mu_{min} = 100$ $\mu_{max} = 250$	$\mu_{min} = 200$ $\mu_{max} = 350$	$\mu_{min} = 300$ $\mu_{max} = 450$
$\bar{D}(\mathcal{T}_p, \mathcal{T}_r)$ (m)	120.87	217.98	336.94	432.94
$\bar{D}(\mathcal{T}^*, \mathcal{T}_r)$ (m)	304.31	522.02	749.71	928.82
E_R (m)	183.44	304.04	412.77	495.88

In terms of utility, the developed model produces results comparable to those of LDPTrace [95], NGRAM [105], and TS-TrajGen [144], using the utility metrics outlined in Section 5.2.3.1, for both adaptive and non-adaptive configurations. From a privacy perspective, the framework is designed to generate trajectories that meet specified privacy objectives, ensuring that even an attacker with access to the model and released trajectories would have limited success in identifying the real trajectory with high accuracy.

Compared to other trajectory generation methods (see Table 5.4), the presented approach allows for the sequential release of generated points while maintaining an optimal balance between privacy and utility, and it operates with low computational overhead. Each point is generated and released as it is collected, requiring no additional processing. However, the deterministic nature of the model exposes a possible vulnerability. An attacker with close approximations of the real coordinates and trajectory length may identify the real trajectory.

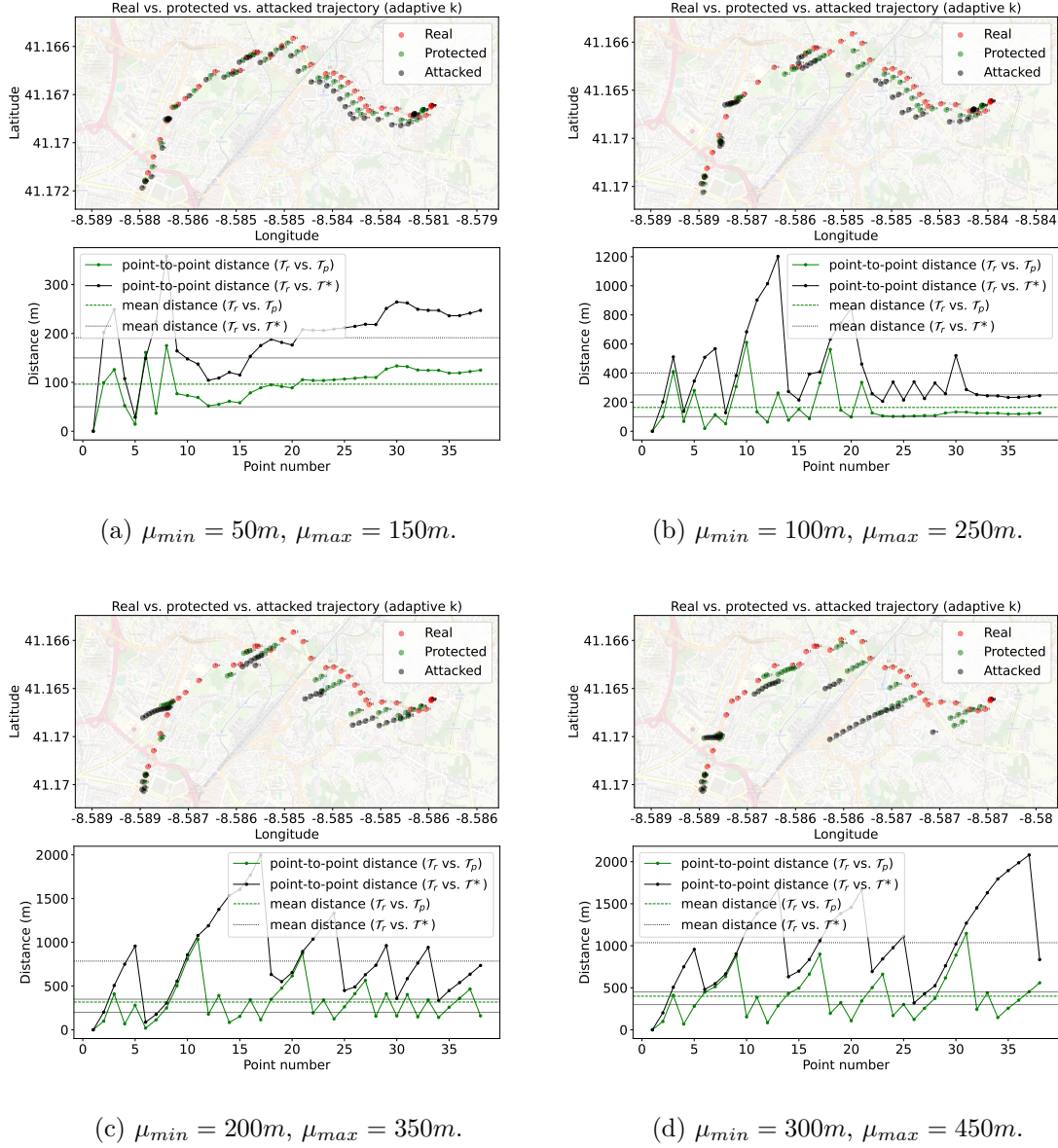


Figure 5.6 Examples of attack results for trajectory id = 1372859766620000467.

A possible solution could decrease the number of released points such that fewer points are generated than those used for trajectory construction.

5.4 Comparison to the state-of-the-art synthetic trajectory approaches

The presented method advances trajectory privacy protection by addressing key limitations in existing approaches, which predominantly rely on DP-based, spatial perturbation, or synthetic data generation techniques. This section compares the method to several state-of-the-art

methods, as summarized in Table 5.4, highlighting how the presented approach uniquely balances privacy and utility in the real-time trajectory data release.

LDPTrace [95] and NGRAM [105] both utilize Local Differential Privacy (LDP) to protect trajectory data by anonymizing each individual trajectory through local perturbations. While these methods effectively preserve user anonymity by ensuring trajectories remain indistinguishable within a larger group, they do not support real-time adaptive privacy-utility balancing. It limits the applicability of the method in dynamic environments where privacy requirements vary in real-time. Furthermore, the perturbation applied to ensure LDP often results in reduced data granularity with the result of limiting the utility for applications that require high spatial and temporal accuracy. In contrast, the APU-TrajGen method includes adaptive privacy-utility balancing, allowing it to dynamically adjust privacy parameters based on current utility needs and privacy risks.

RPTR [146] is another DP-based approach designed specifically for real-time trajectory-based data release. The method focuses on preserving privacy by applying DP perturbations to trajectory data in real-time, which allows for timely data sharing without exposing sensitive information. However, like LDPTrace and NGRAM, RPTR does not incorporate an adaptive privacy-utility balancing mechanism, which can lead to suboptimal data utility when fixed privacy parameters are used in varying application contexts. Additionally, RPTR is limited to trajectory-level privacy with no fine control over individual data points, thus affecting its precision in applications that demand point-to-point data accuracy.

Recent advancements have leveraged deep learning models, including Generative Adversarial Networks (GANs) and LSTM networks, to generate synthetic trajectories that closely resemble real-world data [142, 143, 144, 145]. Methods like TS-TrajGen [144] and LSTM-TrajGAN [145] leverage these models to create realistic, synthetic data, which can protect privacy while maintaining high utility. However, these methods primarily support trajectory-based data release based on Point of Interest (POI) data, limiting their suitability for applications that require point-to-point, real-time data sharing. Additionally, they often apply static privacy parameters that may not reflect varying needs in real-time environments. In contrast, the presented method is designed for real-time, point-to-point data release with adaptive utility-privacy balance. Its point-to-point approach enables granular control over privacy for each trajectory point, making it highly adaptable to changing privacy-utility requirements.

5.5 Summary

In this chapter, the APU-TrajGen method was presented, an adaptive approach for generating privacy-preserving synthetic trajectories to be released in real-time. The method aims to balance privacy and utility by dynamically adjusting the generation of synthetic points using an LSTM-based model while maintaining a low computational overhead suitable for real-time applications. By introducing an adaptive parameter k , the presented model controls the

Table 5.4 Comparison to the state-of-the-art trajectory generator approaches.

Research	Location data	Generation approach	Trajectory release	Adaptive privacy-utility balancing
LDPTrace [95]	Trajectory data (GPS, date, LDP-based time)		Trajectory-based	No
NGRAM [105]	Extended trajectory data (GPS, date, time, POI location)	LDP-based	Trajectory-based	No
RPTR [146]	Trajectory data (GPS, date, DP-based time)		Real-time trajectory-based	No
TS-TrajGen [144]	Trajectory data (GPS, date, Two-stage GAN time)		Trajectory-based	No
LSTM-TrajGAN [145]	Extended trajectory data (GPS, date, time, POI category, POI rating)	LSTM model	Trajectory-based	No
APU-TrajGen (presented in this chapter)	Multiple trajectory attributes (GPS, date, time)	LSTM model	Real-time point-to-point	Yes

distance between real and synthetic trajectory points. and it ensures that the generated data meets predefined privacy and utility thresholds.

The experimental results validated that the APU-TrajGen method is effective in preserving the utility of the synthetic trajectories, as the spatial and temporal patterns necessary for various analytical tasks are retained. The model’s ability to generate synthetic trajectories in a point-by-point manner further enhances its suitability for dynamic environments, such as autonomous driving and real-time monitoring.

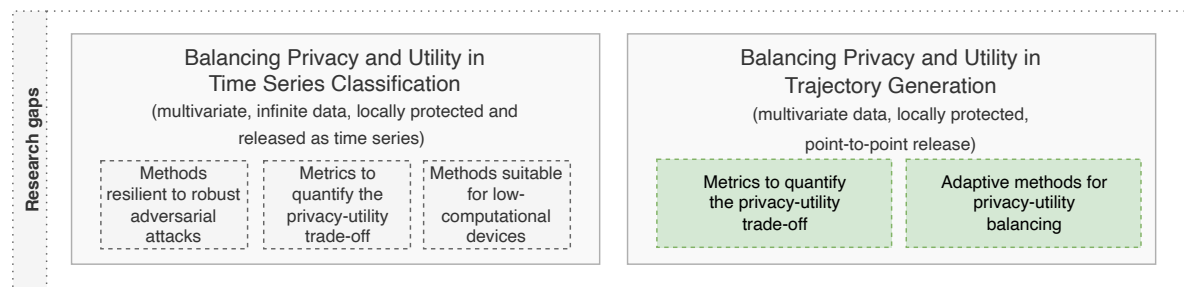


Figure 5.7 The research gaps targeted by the APU-TrajGen method (highlighted in green).

Compared to existing methods like LDPTrace [95] and TS-TrajGen [144], APU-TrajGen achieves a more proper balanced between privacy and utility, offering a higher degree of control over the level of protection while preserving the analytical value of the generated data. The real-time processing capabilities allow for privacy-preserving data release without relying on

a central trusted aggregator. In this way, the risk of data breaches by untrustworthy data curators is significantly reduced.

The APU-TrajGen framework offers a significant contribution to the field of privacy-preserving data release, filling the research gaps (Figure 5.7) between data privacy and utility in real-time applications.

Chapter 6

Conclusions

This chapter provides a summary of the main conclusions and scientific contributions of the thesis, addressing the research objectives established in the introduction. It also highlights the limitations of the proposed methods and suggests possible future research directions.

Achieving a balance between data privacy and utility in the continuous release of time series data presents significant challenges. The research presented in this thesis focuses on developing privacy-preserving mechanisms that address this trade-off, particularly in the context of sensor-generated data. By presenting and evaluating innovative approaches, this work contributes to the advancement of privacy-preserving data analytics and provides practical solutions for complying with privacy regulations – General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), Personal Data Protection Act (PDPA), NIS-2 Directive (NIS-2), and Cyber Resilience Act (CRA). The presented methods are designed to support the data owners and data processors who rely on the utility of protected data, ensuring privacy protection without compromising its utility.

In alignment with the stated research objectives, three novel methods were presented: Feature Selection for Balancing Utility and Privacy in Multivariate Time Series (FS-BUP), Data Perturbation for Balancing Utility and Privacy in Multivariate Time Series (DP-BUP), and APU-TrajGen. These methods were evaluated across diverse scenarios, using powerful adversarial models to test their effectiveness.

6.1 Addressed research questions and objectives

The primary objective of this thesis was to design and develop privacy-preserving data protection methods that achieve an optimal balance between privacy and utility while maintaining computational efficiency for applications with limited resources. The research was guided by three Research Questions (RQs). Through mathematical verification and experimental validation, the following conclusions were drawn.

- **RQ1:** *What are the most effective methods for protecting multivariate time series data in the context of data classification involving a Utility-Focused Classifier (UFC) and a Privacy-Breaking Classifier (PBC)?*

Answer: In the context of Time Series Classification (TSC) for multivariate data, feature selection and feature perturbation were identified as the most effective protection methods. Feature selection ensures that the most sensitive information is never collected, serving as a fundamental privacy-preserving strategy. Feature perturbation further enhances privacy by adjusting the privacy-utility trade-off based on specific requirements. Together, these approaches reduce privacy risks while maintaining classification accuracy.

- **RQ2:** *How can an optimal balance between data privacy and utility be defined and measured?*

Answer: The balance between data privacy and utility is defined in terms of minimizing privacy risks while retaining sufficient information for classification tasks. This trade-off is quantified using privacy and utility measures, such as the Classification Utility-Privacy Balance (BUP) score (for feature perturbation) and the Feature Importance Gap (FIG) score (for feature selection). For synthetic trajectory, Utility-Privacy Score (SUP) allows real-time adjustments of privacy mechanisms. These scores enable optimization, ensuring that privacy-preserving transformations do not significantly decrease data utility in real-world applications.

- **RQ3:** *What approaches can achieve an effective privacy-utility balance across various scenarios, such as protecting multivariate time series data or ensuring location privacy?*

Answer: In TSC for multivariate data, an effective balance is achieved by removing or perturbing features that could lead to user identification. In the same time, features essential for utility classification accuracy are retained. For location privacy, precise data points must be hidden to prevent re-identification, while synthetic trajectory generation remain realistic. The proposed privacy-preserving methods adjust protection levels based on specific privacy-utility requirements and they can be adapted to various scenarios.

The specific research objectives stated in Section 1.2 were addressed through the development of novel methodologies and rigorous validation process.

- **RO1:** *Formulate the problem of ensuring data privacy in the context of TSC.*

Solution: The data privacy challenge in TSC was addressed by introducing the dual-model framework presented in Chapter 3. This framework consists of two competing classifiers: a Utility-Focused Classifier (UFC), which extracts information not related to user identities, and a Privacy-Breaking Classifier (PBC), which attempts to identify users from the data. It enables the evaluation of privacy risks while preserving classification utility.

- **RO2:** *Develop an efficient feature selection method designed for multivariate time series within the context of the presented competing classifiers framework.*

Solution: The FS-BUP method, presented in Chapter 3, was designed and developed within the competing classifiers framework to optimize feature selection, minimizing privacy risks while preserving data utility. It defines the FIG score for ranking features, to enhance selection for privacy preservation. Extensive experiments on diverse datasets validated its effectiveness in achieving a balance between privacy protection and classification accuracy.

- **RO3:** *Present a comprehensive time series data privacy approach that achieves a balance between privacy and utility by selecting suitable perturbation levels for real-world applications.*

Solution: To address **RO3**, the method DP-BUP was presented. The perturbation-based mechanism dynamically adjusts to privacy-utility requirements within the competing classifiers framework. It distributes noise based on feature importance, ensuring flexibility across different perturbation techniques. Additionally, the BUP score was introduced to determine optimal perturbation levels, effectively managing the trade-off between data protection and usability.

- **RO4:** *Extend the concept of privacy-utility balance to location data by presenting a novel privacy-preserving real-time trajectory generation method.*

Solution: To address the research objective **RO4**, the APU-TrajGen method was designed to synthesize realistic trajectories while enabling real-time point-to-point data release. It incorporates a dynamic privacy-utility balancing mechanism, adjusting protections based on trajectory scores at both point-to-point and trajectory-to-trajectory levels. Operating at the device level, APU-TrajGen eliminates reliance on a centralized aggregator, reducing the risk of data breaches while enhancing user privacy control.

- **RO5:** *Validate the effectiveness of these methods through experiments, ensuring their correctness and practical applicability.*

Solution: Extensive experiments on diverse datasets validated the effectiveness of methods FS-BUP and DP-BUP in maintaining classification performance while minimizing privacy risks. Additionally, comprehensive experiments on a widely used location dataset confirmed the validity of the APU-TrajGen method in preserving privacy while maintaining trajectory utility.

6.2 Scientific contributions

This section provides an overview of the key scientific contributions of this thesis. They are presented in detail in their respective chapters, with most having been published as scientific

papers and rigorously reviewed by experts in the field of data privacy. A complete list of published papers is available in the [List of Publications](#) section.

- **Chapter 2:**

- Establishes a strong theoretical foundation for designing privacy-preserving methods in the context of multivariate time series.
- Provides a detailed literature review of existing privacy-preserving methods for time series data, with a focus on approaches applicable to TSC. This analysis identifies research gaps and opportunities for advancing the field. These include the development of privacy-preserving techniques resilient to robust adversarial attacks, methods for deployment on low-computation devices, and solutions for achieving an adjustable balance between privacy and utility.
- Conducts extensive research on the privacy-preserving methods for location data, with a particular focus on synthetic trajectory generation techniques. It provides a scientific foundation for proposing a new solution that balances privacy and utility in trajectory generation.

- **Chapter 3:**

- Designs and develops the FS-BUP method, a novel approach to improve privacy in multivariate time series data by selecting features that balance privacy and utility.
- Presents a model with two competing classifiers: a UFC to preserve data utility and a PBC to measure privacy risks. This dual-model approach offers a unique mechanism for feature selection.
- Formalizes the privacy challenges in TSC by the presentation of the dual-classifier model, allowing for effective privacy protection while retaining classification accuracy.
- Defines the Feature Importance Gap (FIG) as a ranking score to evaluate the relative importance of features for the UFC compared to the PBC, with the objective of selecting the features for privacy optimization.
- Presents a systematic approach for selecting features that prioritize utility for the UFC while reducing the effectiveness of the PBC, achieving a balance that degrades privacy risks without compromising data utility.
- Demonstrates the effectiveness of the feature selection approach through extensive experiments conducted on diverse datasets and scenarios.

- **Chapter 4:**

- Designs and develops the DP-BUP approach, a novel perturbation-based mechanism for time series data that dynamically adapts to the privacy-utility requirements. The same context of the competing classifiers, presented in Chapter 3, is considered.

- Presents a privacy-preserving method independent of perturbation type. This technique applies perturbations to multivariate time series data while distributing noise based on feature importance.
- Defines the Classification Utility-Privacy Balance (BUP) score and a methodology for determining the optimal perturbation levels to achieve the desired privacy-utility trade-off.
- Confirms the effectiveness of the DP-BUP approach through extensive experiments conducted on various datasets and scenarios.

- **Chapter 5:**

- Designs and develops the APU-TrajGen method, an approach for synthesizing trajectories that resemble real-world data while maintaining low computational overhead. The approach enables real-time point-to-point trajectory release.
- Presents a mechanism for dynamically balancing privacy and utility in location data, adapting protections based on trajectory scores evaluated at both the point-to-point and trajectory-to-trajectory levels. The real-time privacy protection mechanism operates at the device level and eliminates the dependence on a centralized aggregator. This approach reduces the risk of data breaches caused by untrustworthy data processors and improves user control over data privacy.
- Defines the Utility-Privacy Score (SUP) score to enable real-time adjustment of the protection model.
- Demonstrates the usefulness of APU-TrajGen method through comprehensive experiments conducted on a popular location dataset.

These contributions highlight important advancements in privacy-preserving techniques for time series and location data, addressing critical challenges such as real-time processing, privacy-utility balancing, and device-level protection.

6.3 Challenges and limitations of the proposed methods

Despite the important advancements presented in this thesis, the proposed methods have limitations. These limitations derive from theoretical assumptions and practical constraints.

The FS-BUP and DP-BUP approaches operate under the assumption that prior knowledge of the UFC and PBC models, or their respective tasks, is available. This knowledge is critical for effectively implementing the privacy mechanisms to balance utility and privacy. However, in real-world scenarios:

- Even though UFCs are known, precise knowledge of the specific PBC models being used may not be available, or these models may evolve over time.

- Multiple utility tasks or adversarial models might exist simultaneously, complicating the optimization process.

These challenges highlight a need for further research into adaptive methods that can generalize across unknown or varying models and tasks.

Further, the DP-BUP approach presents noise to specific features to achieve privacy protection. However, this approach does not account for dependencies or interrelations between features, and perturbing one feature may inadvertently distort related features, potentially reducing overall data utility. Moreover, the approaches assume that the PBC models are specifically designed for user identification. While this assumption is usually true, the method should take into account for adversaries with broader or more diverse objectives. Expanding the framework to address a wider range of adversarial goals would enhance its applicability in real-world scenarios.

The APU-TrajGen method does not require domain knowledge or any user-annotated data to build the LSTM-based model. However, it depends on existing trajectory data, and its usability may be limited in applications where such data is not available. Moreover, the deterministic nature of the proposed trajectory generation model presents a vulnerability. An attacker with close approximations of real coordinates and trajectory length could potentially identify the actual trajectory. The risk is increased in situations where adversaries have access to partial data. One potential solution to this issue is to reduce the number of released points or ensure that fewer points are generated than those needed for trajectory reconstruction. However, this compromise may further diminish the utility of the data, so it requires careful evaluation.

6.4 Future research

The proposed methods (FS-BUP, DP-BUP, APU-TrajGen) have shown impressive results in balancing privacy and utility. However, further research is necessary to address the identified limitations, to improve the efficiency of the methods, and to increase their applicability across a variety of scenarios.

The feature selection and importance computation presented in FS-BUP and DP-BUP should be generalized to accommodate a broader range of scenarios. Future work should also compare different perturbation methods to identify the most effective approaches for various data types and classification tasks, including the classification of perturbed data. Additional tests on diverse datasets will further validate the robustness of these methods.

Future research should focus on developing adaptive mechanisms capable of operating effectively without requiring prior knowledge of specific UFC or PBC models. This generalization would ensure that privacy-preserving methods remain effective even when detailed information about the adversarial or utility-focused models is unavailable or when these models evolve over time. Broadening the adversarial models would make the methods

more robust and applicable to real-world environments where adversaries may have diverse or evolving objectives.

The DP-BUP method does not account for interdependencies between features, leading to potential utility loss when noise is added to one feature that inadvertently distorts others. Incorporating feature correlation modeling into noise addition strategies will minimize utility reduction, particularly for datasets with strong inter-feature correlations. A more detailed analysis of how different levels of noise perturbation affect data utility is necessary. This would provide insights into the process of balancing privacy and utility. Additionally, it would lead to a better selection of optimal perturbation levels for specific tasks.

Future research on APU-TrajGen should address its reliance on existing trajectory datasets, which may limit usability in scenarios where sufficient data is unavailable. Additionally, the deterministic nature of the model presents a privacy risk, as adversaries with partial trajectory knowledge may reconstruct real trajectories. To mitigate the risk, future strategies may need to reduce the number of released points. Releasing fewer points than those used for trajectory construction can increase privacy while maintaining utility. Future research should aim to balance privacy and data utility, increase protection against attacks that reveal hidden patterns, and make the method more effective in situations with limited data.

References

- [1] Theresa Payton and Ted Claypoole. *Privacy in the age of Big data: Recognizing threats, defending your rights, and protecting your family*. Rowman & Littlefield, 2023.
- [2] European Parliament and Council of the European Union. *Regulation (EU) 2023/2854 of the European Parliament and of the Council*. of 13 December 2023 on harmonised rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act). Dec. 13, 2023. URL: <https://eur-lex.europa.eu/eli/reg/2023/2854/oj> (visited on 11/06/2024).
- [3] Shah Khalid Khan, Nirajan Shiwakoti, Peter Stasinopoulos, and Yilun Chen. “Cyber-attacks in the next-generation cars, mitigation techniques, anticipated readiness and future directions”. In: *Accident Analysis & Prevention* 148 (2020), p. 105837.
- [4] Jiwei Tian, Buhong Wang, Rongxiao Guo, Zhen Wang, Kunrui Cao, and Xiaodong Wang. “Adversarial attacks and defenses for deep-learning-based unmanned aerial vehicles”. In: *IEEE Internet of Things Journal* 9.22 (2021), pp. 22399–22409.
- [5] S Thirumalini and Pooja Malemutt. “Investigations on anti-Tampering of diesel particulate filter”. In: *Materials Today: Proceedings* 46 (2021), pp. 4988–4992.
- [6] Robert Shumway and David Stoffer. *Time Series Analysis and Its Applications With R Examples*. Vol. 9. Jan. 2011. ISBN: 978-1-4419-7864-6. DOI: [10.1007/978-1-4419-7865-3](https://doi.org/10.1007/978-1-4419-7865-3).
- [7] Andreas Theissler, Daniel Ulmer, and Ian Dear. “Interactive knowledge discovery in recordings from vehicle tests”. In: *Proc. 33rd FISITA World Automotive Congress. Budapest, Hungary*. 2010.
- [8] David Hallac, Abhijit Sharang, Rainer Stahlmann, Andreas Lamprecht, Markus Huber, Martin Roehder, Jure Leskovec, et al. “Driver identification using automobile sensor data from a single turn”. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2016, pp. 953–958.
- [9] Sakorn Mekruksavanich and Anuchit Jitpattanakul. “Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models”. In: *Electronics* 10.3 (2021), p. 308.
- [10] Franklin Leukam Lako, Paul Lajoie-Mazenc, and Maryline Laurent. “Privacy-Preserving Publication of Time-Series Data in Smart Grid”. In: *Security and Communication Networks* (2021), p. 6643566. ISSN: 1939-0114.
- [11] Zhibo Wang, Wenxin Liu, Xiaoyi Pang, Ju Ren, Zhe Liu, and Yongle Chen. “Towards Pattern-aware Privacy-preserving Real-time Data Collection”. In: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. 2020, pp. 109–118.
- [12] Kun Liu, Chris Giannella, and Hillol Kargupta. “A Survey of Attack Techniques on Privacy-Preserving Data Perturbation Methods”. In: June 2008, pp. 359–381. ISBN: 978-0-387-70991-8. DOI: [10.1007/978-0-387-70992-5](https://doi.org/10.1007/978-0-387-70992-5).
- [13] European Data Protection Board. “Guidelines 1/2020 on processing personal data in the context of connected vehicles and mobility related applications”. In: (2020).

- [14] European Parliament and Council of the European Union. *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). May 4, 2016. URL: <https://data.europa.eu/eli/reg/2016/679/oj> (visited on 04/23/2024).
- [15] *The California Consumer Privacy Act*. June 29, 2018. URL: https://leginfo.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375 (visited on 04/23/2024).
- [16] *Personal Data Protection Act*. Jan. 2, 2013. URL: <https://sso.agc.gov.sg/Act/PDPA2012> (visited on 04/23/2024).
- [17] *Communication from the Commission Commission Guidelines on the application of Article 4 (1) and (2) of Directive (EU) 2022/2555 (NIS 2 Directive) 2023/C 328/02*. Sept. 2023.
- [18] European Commission, Content Directorate-General for Communications Networks, and Technology. *Cyber resilience act – New EU cybersecurity rules ensure more secure hardware and software products*. European Commission, 2022. DOI: [doi/10.2759/543836](https://doi.org/10.2759/543836).
- [19] Manos Katsomallos, Katerina Tzompanaki, and Dimitris Kotzinos. “Privacy, space and time: A survey on privacy-preserving continuous data publishing”. In: *Journal of Spatial Information Science* 2019.19 (2019), pp. 57–103.
- [20] Joseph P. Near and Chiké Abuah. *Programming Differential Privacy*. Vol. 1. 2021. URL: <https://uvm-plaid.github.io/programming-dp/>.
- [21] Cynthia Dwork. “Differential Privacy in New Settings”. In: *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '10. Austin, Texas: Society for Industrial and Applied Mathematics, 2010, pp. 174–183. ISBN: 9780898716986.
- [22] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. “Differentially Private Event Sequences over Infinite Streams”. In: *Proc. VLDB Endow.* 7.12 (2014), pp. 1155–1166. ISSN: 2150-8097. DOI: [10.14778/2732977.2732989](https://doi.org/10.14778/2732977.2732989). URL: <https://doi.org/10.14778/2732977.2732989>.
- [23] Qingqing Ye, Haibo Hu, Ninghui Li, Xiaofeng Meng, Huadi Zheng, and Haotian Yan. “Beyond value perturbation: Local differential privacy in the temporal setting”. In: *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE. 2021, pp. 1–10.
- [24] Zhangbing Li, Baichuan Wang, Jinsheng Li, Yi Hua, and Shaobo Zhang. “Local differential privacy protection for wearable device data”. In: *Plos one* 17.8 (2022), e0272766.
- [25] Adrian-Silviu Roman, Béla Genge, and Piroska Haller. “Balancing Privacy and Utility in Multivariate Time-Series Classification”. In: *Computer Security. ESORICS 2024 International Workshops*. Springer. 2025.
- [26] Mohammad Malekzadeh, Richard Clegg, Andrea Cavallaro, and Hamed Haddadi. “Privacy and utility preserving sensor-data transformations”. In: *Pervasive and Mobile Computing* 63 (2020), p. 101132.
- [27] Xuebin Ren, Chia-Mu Yu, Weiren Yu, Shusen Yang, Xinyu Yang, Julie A McCann, and S Yu Philip. “LoPub: high-dimensional crowdsourced data publication with local differential privacy”. In: *IEEE Transactions on Information Forensics and Security* 13.9 (2018), pp. 2151–2166.
- [28] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. “Mobile sensor data anonymization”. In: *Proceedings of the international conference on internet of things design and implementation*. 2019, pp. 49–58.

- [29] Spiros Papadimitriou, Feifei Li, George Kollios, and Philip S. Yu. “Time Series Compressibility and Privacy”. In: *33rd International Conference on Very Large Data Bases*. VLDB '07. VLDB Endowment, 2007, pp. 459–470. ISBN: 9781595936493.
- [30] Vibhor Rastogi and Suman Nath. “Differentially Private Aggregation of Distributed Time-Series with Transformation and Encryption”. In: *2010 ACM SIGMOD International Conference on Management of Data*. Indianapolis, Indiana, USA, 2010, pp. 735–746. ISBN: 9781450300322.
- [31] Lingjuan Lyu, Xuanli He, Yee Wei Law, and Marimuthu Palaniswami. “Privacy-Preserving Collaborative Deep Learning with Application to Human Activity Recognition”. In: Nov. 2017, pp. 1219–1228. DOI: [10.1145/3132847.3132990](https://doi.org/10.1145/3132847.3132990).
- [32] Tao Wang, Zhigao Zheng, Mubashir Husain Rehmani, Shihong Yao, and Zheng Huo. “Privacy Preservation in Big Data From the Communication Perspective—A Survey”. In: *IEEE Communications Surveys Tutorials* 21.1 (2019), pp. 753–778. DOI: [10.1109/COMST.2018.2865107](https://doi.org/10.1109/COMST.2018.2865107).
- [33] Mengmeng Yang, Taolin Guo, Tianqing Zhu, Ivan Tjuawinata, Jun Zhao, and Kwok-Yan Lam. “Local differential privacy and its applications: A comprehensive survey”. In: *Computer Standards & Interfaces* (2023), p. 103827.
- [34] Gang Xiong, Zhishuai Li, Meihua Zhao, Yu Zhang, Qinghai Miao, Yisheng Lv, and Fei-Yue Wang. “TrajSGAN: A semantic-guiding adversarial network for urban trajectory generation”. In: *IEEE Transactions on Computational Social Systems* 11.2 (2023), pp. 1733–1743.
- [35] Jiahao Ji, Jingyuan Wang, Junjie Wu, Boyang Han, Junbo Zhang, and Yu Zheng. “Precision CityShield against hazardous chemicals threats via location mining and self-supervised learning”. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 3072–3080.
- [36] Houxing Ren, Jingyuan Wang, and Wayne Xin Zhao. “Generative adversarial networks enhanced pre-training for insufficient electronic health records modeling”. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 3810–3818.
- [37] Adrian-Silviu Roman, Béla Genge, and Roland Bolboacă. “Privacy-Oriented Feature Selection for Multivariate Time Series Classification”. In: *Procedia Computer Science* 246 (2024). 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024), pp. 500–509. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2024.09.430>.
- [38] Adrian-Silviu Roman, Béla Genge, Adrian-Vasile Duka, and Piroska Haller. “Privacy-preserving tampering detection in automotive systems”. In: *Electronics* 10.24 (2021), p. 3161.
- [39] Adrian-Silviu Roman. “Evaluating the Privacy and Utility of Time-Series Data Perturbation Algorithms”. In: *Mathematics* 11.5 (2023), p. 1260.
- [40] Adrian-Silviu Roman. “Dynamic Utility-Improving Time Series Data Perturbation Framework”. In: *International Conference Interdisciplinarity in Engineering*. Springer. 2022, pp. 850–861.
- [41] Adrian-Silviu Roman and Bèla Genge. “A Differential Privacy-Based Approach for Protecting Crowdsourced WiFi RSS Data”. In: *2023 IEEE 19th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE. 2023, pp. 255–262.
- [42] L. Sweeney. “k-Anonymity: A Model for Protecting Privacy”. In: *IEEE Security and Privacy* 10 (2002), pp. 1–14.

- [43] Shibnath Mukherjee, Zhiyuan Chen, and Aryya Gangopadhyay. “A privacy-preserving technique for Euclidean distance-based mining algorithms using Fourier-related transforms”. In: *VLDB J.* 15 (Nov. 2006), pp. 293–315. DOI: [10.1007/s00778-006-0010-5](https://doi.org/10.1007/s00778-006-0010-5).
- [44] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity”. In: vol. 2. 2007, pp. 106–115. ISBN: 1-4244-0803-2.
- [45] Fushan Wei, Pandi Vijayakumar, Neeraj Kumar, Ruijie Zhang, and Qingfeng Cheng. “Privacy-preserving implicit authentication protocol using cosine similarity for Internet of Things”. In: *IEEE Internet of Things Journal* 8.7 (2020), pp. 5599–5606.
- [46] Rakesh Agrawal and Ramakrishnan Srikant. “Privacy-Preserving Data Mining”. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’00. Dallas, Texas, USA: Association for Computing Machinery, 2000, pp. 439–450. ISBN: 1581132174. DOI: [10.1145/342009.335438](https://doi.org/10.1145/342009.335438). URL: <https://doi.org/10.1145/342009.335438>.
- [47] Teng Wang, Xuefeng Zhang, Jingyu Feng, and Xinyu Yang. “A comprehensive survey on local differential privacy toward data statistics and analysis”. In: *Sensors* 20.24 (2020), p. 7030.
- [48] Ashis Kumar Chanda, Swapnil Saha, Manziba Akanda Nishi, Md Samiullah, and Chowdhury Farhan Ahmed. “An efficient approach to mine flexible periodic patterns in time series databases”. In: *Engineering Applications of Artificial Intelligence* 44 (2015), pp. 46–63.
- [49] Yulian Mao, Qingqing Ye, Qi Wang, and Haibo Hu. “Differential Privacy for Time Series: A Survey”. In: *Data Engineering* (2023), p. 68.
- [50] Alejandro Pasos Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and Anthony Bagnall. “The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances”. In: *Data Mining and Knowledge Discovery* 35.2 (2021), pp. 401–449.
- [51] Adil Hussain Seh, Mohammad Zarour, Mamdouh Alenezi, Amal Krishna Sarkar, Alka Agrawal, Rajeev Kumar, and Raees Ahmad Khan. “Healthcare data breaches: insights and implications”. In: *Healthcare*. Vol. 8. 2. MDPI. 2020, p. 133.
- [52] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. “Differential Privacy Techniques for Cyber Physical Systems: A Survey”. In: *IEEE Communications Surveys Tutorials* 22.1 (2020), pp. 746–789.
- [53] Manos Katsomaillos, Katerina Tzompanaki, and Dimitris Kotzinos. “Privacy, space and time: A survey on privacy-preserving continuous data publishing”. In: *Journal of Spatial Information Science* 2019.19 (2019), pp. 57–103.
- [54] Tao Wang, Zhigao Zheng, Mubashir Husain Rehmani, Shihong Yao, and Zheng Huo. “Privacy Preservation in Big Data From the Communication Perspective—A Survey”. In: *IEEE Communications Surveys Tutorials* 21.1 (2019), pp. 753–778.
- [55] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. “l-Diversity: Privacy Beyond k-Anonymity”. In: vol. 1. 2006, p. 24.
- [56] Keke Chen and Ling Liu. “Privacy preserving data classification with rotation perturbation”. In: *Fifth IEEE International Conference on Data Mining (ICDM’05)*. 2005. DOI: [10.1109/ICDM.2005.121](https://doi.org/10.1109/ICDM.2005.121).

- [57] Ella Bingham and Heikki Mannila. “Random Projection in Dimensionality Reduction: Applications to Image and Text Data”. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '01. San Francisco, California: Association for Computing Machinery, 2001, pp. 245–250. ISBN: 158113391X. DOI: [10.1145/502512.502546](https://doi.org/10.1145/502512.502546). URL: <https://doi.org/10.1145/502512.502546>.
- [58] Kanishka Bhaduri, Mark D. Stefanski, and Ashok N. Srivastava. “Privacy-Preserving Outlier Detection Through Random Nonlinear Data Distortion”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41.1 (2011), pp. 260–272. DOI: [10.1109/TSMCB.2010.2051540](https://doi.org/10.1109/TSMCB.2010.2051540).
- [59] Cynthia Dwork. “Differential privacy: A survey of results”. In: *International conference on theory and applications of models of computation*. Springer. 2008, pp. 1–19.
- [60] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. “Calibrating noise to sensitivity in private data analysis”. In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.
- [61] Hao Wang and Zhengquan Xu. “CTS-DP: publishing correlated time-series data via differential privacy”. In: *Knowledge-Based Systems* 122 (2017), pp. 167–179.
- [62] Sun-Kyong Hong, Kuldeep Gurjar, Hea-Suk Kim, and Yang-Sae Moon. “A Survey on Privacy Preserving Time-Series Data Mining”. In: 2013.
- [63] Héber Hwang Arcolezi, Jean-François Couchot, Denis Renaud, Bechara Al Bouna, and Xiaokui Xiao. “Differentially private multivariate time series forecasting of aggregated human mobility with deep learning: Input or gradient perturbation?”. In: *Neural Computing and Applications* 34.16 (2022), pp. 13355–13369.
- [64] Cynthia Dwork. “A firm foundation for private data analysis”. In: *Communications of the ACM* 54.1 (2011), pp. 86–95.
- [65] Cynthia Dwork, Aaron Roth, et al. “The algorithmic foundations of differential privacy”. In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.
- [66] Stanley L Warner. “Randomized response: A survey technique for eliminating evasive answer bias”. In: *Journal of the American Statistical Association* 60.309 (1965), pp. 63–69.
- [67] Frank D. McSherry. “Privacy Integrated Queries: An Extensible Platform for Privacy-Preserving Data Analysis”. In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. SIGMOD '09. Providence, Rhode Island, USA: Association for Computing Machinery, 2009, pp. 19–30. ISBN: 9781605585512. DOI: [10.1145/1559845.1559850](https://doi.org/10.1145/1559845.1559850). URL: <https://doi.org/10.1145/1559845.1559850>.
- [68] Jordi Soria-Comas and Josep Domingo-Ferrer. “Big data privacy: challenges to privacy principles and models”. In: *Data science and engineering* 1.1 (2016), pp. 21–28.
- [69] Liyue Fan, Li Xiong, and Vaidy Sunderam. “Differentially private multi-dimensional time series release for traffic monitoring”. In: *Data and Applications Security and Privacy XXVII: 27th Annual IFIP WG 11.3 Conference, DBSec 2013, Newark, NJ, USA, July 15-17, 2013. Proceedings 27*. Springer. 2013, pp. 33–48.
- [70] John C Duchi, Michael I Jordan, and Martin J Wainwright. “Local privacy and statistical minimax rates”. In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE. 2013, pp. 429–438.
- [71] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. “Rappor: Randomized aggregatable privacy-preserving ordinal response”. In: *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 2014, pp. 1054–1067.

- [72] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. “Collecting telemetry data privately”. In: *Advances in Neural Information Processing Systems* 30 (2017).
- [73] Differential Privacy Team. *Learning with Privacy at Scale*. URL: <https://machinelearning.apple.com/research/learning-with-privacy-at-scale> (visited on 12/01/2023).
- [74] Matthew Joseph, Aaron Roth, Jonathan Ullman, and Bo Waggoner. “Local differential privacy for evolving data”. In: *Advances in Neural Information Processing Systems* 31 (2018).
- [75] Xuebin Ren, Liang Shi, Weiren Yu, Shusen Yang, Cong Zhao, and Zongben Xu. “LDP-IDS: Local differential privacy for infinite data streams”. In: *Proceedings of the 2022 international conference on management of data*. 2022, pp. 1064–1077.
- [76] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. “Differential privacy in practice: Expose your epsilons!” In: *Journal of Privacy and Confidentiality* 9.2 (2019).
- [77] Georgios Kellaris and Stavros Papadopoulos. “Practical differential privacy via grouping and smoothing”. In: *Proceedings of the VLDB Endowment* 6.5 (2013), pp. 301–312.
- [78] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. “Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy”. In: *IEEE Transactions on Dependable and Secure Computing* 15.4 (2016), pp. 591–606.
- [79] Ferdinando Fioretto and Pascal Van Hentenryck. “Optstream: Releasing time series privately”. In: *Journal of Artificial Intelligence Research* 65 (2019), pp. 423–456.
- [80] E Oran Brigham. *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1988.
- [81] Shuting Xu and Shuhua Lai. “Fast Fourier transform based data perturbation method for privacy protection”. In: *2007 IEEE Intelligence and Security Informatics*. IEEE. 2007, pp. 221–224.
- [82] Swati Dewangan, Anup Sharma, et al. “Image smoothening and sharpening using frequency domain filtering technique”. In: *Int. J. Emerg. Technol. Eng. Res* 5.4 (2017), pp. 169–174.
- [83] Eunmok Yang, Velmurugan Subbiah Parvathy, P Pandi Selvi, K Shankar, Changho Seo, Gyanendra Prasad Joshi, and Okyeon Yi. “Privacy Preservation in Edge Consumer Electronics by Combining Anomaly Detection with Dynamic Attribute-Based Re-Encryption”. In: *Mathematics* 8.11 (2020), p. 1871.
- [84] Roland Bolboacă. “Adaptive Ensemble Methods for Tampering Detection in Automotive Aftertreatment Systems”. In: *IEEE Access* 10 (2022), pp. 105497–105517.
- [85] Jordi Nin and Vicenç Torra. “Towards the evaluation of time series protection methods”. In: *Information Sciences* 179.11 (2009), pp. 1663–1677.
- [86] Dakshi Agrawal and Charu C. Aggarwal. “On the Design and Quantification of Privacy Preserving Data Mining Algorithms”. In: *20th ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems*. 2001, pp. 247–255.
- [87] Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. “Privacy preserving mining of association rules”. In: *eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 217–228.
- [88] Jeffrey Scott Vitter and Min Wang. “Approximate Computation of Multidimensional Aggregates of Sparse Data Using Wavelets”. In: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*. SIGMOD '99. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1999, pp. 193–204. ISBN: 1581130848. DOI: [10 . 1145 / 304182 . 304199](https://doi.org/10.1145/304182.304199). URL: <https://doi.org/10.1145/304182.304199>.

- [89] Quan Geng and Pramod Viswanath. “The optimal noise-adding mechanism in differential privacy”. In: *IEEE Transactions on Information Theory* 62.2 (2015), pp. 925–951.
- [90] Jordi Soria-Comas and Josep Domingo-Ferrer. “Optimal data-independent noise for differential privacy”. In: *Information Sciences* 250 (2013), pp. 200–214.
- [91] Xiaokui Xiao, Gabriel Bender, Michael Hay, and Johannes Gehrke. “iReduct: Differential privacy with reduced relative errors”. In: *2011 ACM SIGMOD International Conference on Management of data*. 2011, pp. 229–240.
- [92] Xinyu Yang, Xuebin Ren, Jie Lin, and Wei Yu. “On binary decomposition based privacy-preserving aggregation schemes in real-time monitoring systems”. In: *IEEE Transactions on Parallel and Distributed Systems* 27.10 (2016), pp. 2967–2983.
- [93] Gergely Acs, Claude Castelluccia, and Rui Chen. “Differentially private histogram publishing through lossy compression”. In: *2012 IEEE 12th International Conference on Data Mining*. IEEE. 2012, pp. 1–10.
- [94] Tianqing Zhu, Ping Xiong, Gang Li, and Wanlei Zhou. “Correlated differential privacy: Hiding information in non-IID data set”. In: *IEEE Transactions on Information Forensics and Security* 10.2 (2014), pp. 229–242.
- [95] Yuntao Du, Yujia Hu, Zhikun Zhang, Ziquan Fang, Lu Chen, Baihua Zheng, and Yunjun Gao. “Ldptrace: Locally differentially private trajectory synthesis”. In: *Proceedings of the VLDB Endowment* 16.8 (2023), pp. 1897–1909.
- [96] Mehmet Emre Gursoy, Ling Liu, Stacey Truex, Lei Yu, and Wenqi Wei. “Utility-aware synthesis of differentially private and attack-resilient location traces”. In: *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 2018, pp. 196–211.
- [97] Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia Procopiuc, and Divesh Srivastava. “DPT: differentially private trajectory synthesis using hierarchical reference systems”. In: *Proceedings of the VLDB Endowment* 8.11 (2015), pp. 1154–1165.
- [98] Fengmei Jin, Wen Hua, Boyu Ruan, and Xiaofang Zhou. “Frequency-based randomization for guaranteeing differential privacy in spatial trajectories”. In: *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE. 2022, pp. 1727–1739.
- [99] Dong Xie, Feifei Li, and Jeff M Phillips. “Distributed trajectory similarity search”. In: *Proceedings of the VLDB Endowment* 10.11 (2017), pp. 1478–1489.
- [100] Eamonn Keogh and Chotirat Ann Ratanamahatana. “Exact indexing of dynamic time warping”. In: *Knowledge and information systems* 7 (2005), pp. 358–386.
- [101] Lei Chen, M Tamer Özsu, and Vincent Oria. “Robust and fast similarity search for moving object trajectories”. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. 2005, pp. 491–502.
- [102] Xinyu Yang, Teng Wang, Xuebin Ren, and Wei Yu. “Survey on improving data utility in differentially private sequential data publishing”. In: *IEEE Transactions on Big Data* 7.4 (2017), pp. 729–749.
- [103] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. “Differential privacy via wavelet transforms”. In: *IEEE Transactions on knowledge and data engineering* 23.8 (2010), pp. 1200–1214.
- [104] Jingyu Hua, Yue Gao, and Sheng Zhong. “Differentially private publication of general time-serial trajectory data”. In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2015, pp. 549–557.

- [105] Teddy Cunningham, Graham Cormode, Hakan Ferhatosmanoglu, and Divesh Srivastava. “Real-world trajectory sharing with local differential privacy”. In: *arXiv preprint arXiv:2108.02084* (2021).
- [106] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [107] Alexandra Stefan, Vassilis Athitsos, and Gautam Das. “The move-split-merge metric for time series”. In: *IEEE Transactions on Knowledge and Data Engineering* 25.6 (2012), pp. 1425–1438.
- [108] Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei, and Chotirat Ann Ratanamahatana. “Fast time series classification using numerosity reduction”. In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 1033–1040.
- [109] Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. “Learning time-series shapelets”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 392–401.
- [110] Lexiang Ye and Eamonn Keogh. “Time series shapelets: a new primitive for data mining”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2009, pp. 947–956.
- [111] Jason Lines and Anthony Bagnall. “Time series classification with ensembles of elastic distance measures”. In: *Data Mining and Knowledge Discovery* 29 (2015), pp. 565–592.
- [112] Houtao Deng, George Runger, Eugene Tuv, and Martyanov Vladimir. “A time series forest for classification and feature extraction”. In: *Information Sciences* 239 (2013), pp. 142–153.
- [113] Mustafa Gokce Baydogan, George Runger, and Eugene Tuv. “A bag-of-features framework to classify time series”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.11 (2013), pp. 2796–2802.
- [114] Zhiguang Wang, Weizhong Yan, and Tim Oates. “Time series classification from scratch with deep neural networks: A strong baseline”. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 1578–1585.
- [115] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. “Deep learning for time series classification: a review”. In: *Data mining and knowledge discovery* 33.4 (2019), pp. 917–963.
- [116] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. “Recurrent Neural Networks for Multivariate Time Series with Missing Values”. In: *Scientific Reports* 8.1 (Apr. 2018), p. 6085. ISSN: 2045-2322. DOI: [10.1038/s41598-018-24271-9](https://doi.org/10.1038/s41598-018-24271-9). URL: <https://doi.org/10.1038/s41598-018-24271-9>.
- [117] Zhiwen Xiao, Xin Xu, Huanlai Xing, Shouxi Luo, Penglin Dai, and Dawei Zhan. “RTFN: A robust temporal feature network for time series classification”. In: *Information Sciences* 571 (2021), pp. 65–86. ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2021.04.053>. URL: <https://www.sciencedirect.com/science/article/pii/S0020025521003820>.
- [118] Craig Gentry. *A fully homomorphic encryption scheme*. Stanford university, 2009.
- [119] Yuxuan Xiu, Xinyue Ren, Ting Zhang, Yanyu Chen, Li Jiang, Duo Li, Xingjun Wang, Liang Zhao, and Wai Kin Chan. “Time labeled visibility graph for privacy-preserved physiological time series classification”. In: *2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*. IEEE. 2022, pp. 280–284.
- [120] Jing Li, Xiaohui Kuang, Shujie Lin, Xu Ma, and Yi Tang. “Privacy preservation for machine learning training and classification based on homomorphic encryption schemes”. In: *Information Sciences* 526 (2020), pp. 166–179.

- [121] Dabeeruddin Syed, Shady S Refaat, and Othmane Bouhali. “Privacy preservation of data-driven models in smart grids using homomorphic encryption”. In: *Information* 11.7 (2020), p. 357.
- [122] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. “Deep learning with differential privacy”. In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318.
- [123] Shuo Wang, Carsten Rudolph, Surya Nepal, Marthie Grobler, and Shangyu Chen. “PART-GAN: Privacy-preserving time-series sharing”. In: *Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part I* 29. Springer. 2020, pp. 578–593.
- [124] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. “Communication-efficient learning of deep networks from decentralized data”. In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.
- [125] Zhiyu Liang and Hongzhi Wang. “FedST: Federated Shapelet Transformation for Interpretable Time Series Classification”. In: *arXiv preprint arXiv:2302.10631* (2023).
- [126] Girish Chandrashekar and Ferat Sahin. “A survey on feature selection methods”. In: *Computers & Electrical Engineering* 40.1 (2014), pp. 16–28.
- [127] Tao Zhang, Tianqing Zhu, Ping Xiong, Huan Huo, Zahir Tari, and Wanlei Zhou. “Correlated differential privacy: Feature selection in machine learning”. In: *IEEE Transactions on Industrial Informatics* 16.3 (2019), pp. 2115–2124.
- [128] Mina Sheikhalishahi and Fabio Martinelli. “Privacy-utility feature selection as a privacy mechanism in collaborative data classification”. In: *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE. 2017, pp. 244–249.
- [129] S Srijayanthi and T Sethukarasi. “Design of privacy preserving model based on clustering involved anonymization along with feature selection”. In: *Computers & Security* 126 (2023), p. 103027.
- [130] Yasser Jafer, Stan Matwin, and Marina Sokolova. “Task oriented privacy preserving data publishing using feature selection”. In: *Advances in Artificial Intelligence: Proceedings of the 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014*. Springer. 2014, pp. 143–154.
- [131] Yasser Jafer, Stan Matwin, and Marina Sokolova. “A framework for a privacy-aware feature selection evaluation measure”. In: *2015 13th Annual Conference on Privacy, Security and Trust (PST)*. IEEE. 2015, pp. 62–69.
- [132] Xianjin Fang, Qingkui Zeng, and Gaoming Yang. “Local differential privacy for data streams”. In: *International Conference on Security and Privacy in Digital Economy*. Springer. 2020, pp. 143–160.
- [133] Daniel Kifer and Ashwin Machanavajjhala. “No free lunch in data privacy”. In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 2011, pp. 193–204.
- [134] Chi-Yin Chow and Mohamed F Mokbel. “Trajectory privacy in location-based services and data publication”. In: *ACM Sigkdd Explorations Newsletter* 13.1 (2011), pp. 19–29.
- [135] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. “Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data”. In: *Proceedings of the 26th international conference on world wide web*. 2017, pp. 1241–1250.

- [136] Ben Niu, Qinghua Li, Xiaoyan Zhu, Guohong Cao, and Hui Li. “Achieving k-anonymity in privacy-aware location-based services”. In: *IEEE INFOCOM 2014-IEEE conference on computer communications*. IEEE. 2014, pp. 754–762.
- [137] Marco Gruteser and Dirk Grunwald. “Anonymous usage of location-based services through spatial and temporal cloaking”. In: *Proceedings of the 1st international conference on Mobile systems, applications and services*. 2003, pp. 31–42.
- [138] Dara E Seidl, Piotr Jankowski, and Ming-Hsiang Tsou. “Privacy and spatial pattern preservation in masked GPS trajectory data”. In: *International Journal of Geographical Information Science* 30.4 (2016), pp. 785–800.
- [139] Song Gao, Jinmeng Rao, Xinyi Liu, Yuhao Kang, Qunying Huang, and Joseph App. “Exploring the effectiveness of geomasking techniques for protecting the geoprivacy of Twitter users”. In: *Journal of Spatial Information Science* 19 (2019), pp. 105–129.
- [140] Mei-Po Kwan, Irene Casas, and Ben Schmitz. “Protection of geoprivacy and accuracy of spatial information: How effective are geographical masks?” In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 39.2 (2004), pp. 15–28.
- [141] Kristen H Hampton, Molly K Fitch, William B Allshouse, Irene A Doherty, Dionne C Gesink, Peter A Leone, Marc L Serre, and William C Miller. “Mapping health data: improved privacy protection with donut method geomasking”. In: *American journal of epidemiology* 172.9 (2010), pp. 1062–1069.
- [142] Xi Liu, Hanzhou Chen, and Clio Andris. “trajGANs: Using generative adversarial networks for geo-privacy protection of trajectory data (Vision paper)”. In: *Location privacy and security workshop*. 2018, pp. 1–7.
- [143] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. “Social gan: Socially acceptable trajectories with generative adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2255–2264.
- [144] Wenjun Jiang, Wayne Xin Zhao, Jingyuan Wang, and Jiawei Jiang. “Continuous trajectory generation based on two-stage GAN”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 4. 2023, pp. 4374–4382.
- [145] Jinmeng Rao, Song Gao, Yuhao Kang, and Qunying Huang. “LSTM-TrajGAN: A deep learning approach to trajectory privacy protection”. In: *arXiv preprint arXiv:2006.10521* (2020).
- [146] Zhuo Ma, Tian Zhang, Ximeng Liu, Xinghua Li, and Kui Ren. “Real-time privacy-preserving data release over vehicle trajectory”. In: *IEEE transactions on vehicular technology* 68.8 (2019), pp. 8091–8102.
- [147] Eduardo Romera, Luis M Bergasa, and Roberto Arroyo. “Need data for driver behaviour analysis? Presenting the public UAH-DriveSet”. In: *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*. IEEE. 2016, pp. 387–392.
- [148] Byung Il Kwak, Jiyoung Woo, and Huy Kang Kim. “Know your master: Driver Profiling-based Anti-theft method”. In: *PST 2016*. 2016.
- [149] Attila Reiss. *PAMAP2 Physical Activity Monitoring*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5NW2H>. 2012.
- [150] Leo Breiman. “Random Forests”. In: *Machine Learning* 45 (2001), pp. 5–32.
- [151] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.
- [152] Ron Larson. *Precalculus: a concise course*. Cengage learning, 2013.

- [153] Riccardo Coppola and Maurizio Morisio. “Connected car: technologies, issues, future trends”. In: *ACM Computing Surveys (CSUR)* 49.3 (2016), pp. 1–36.
- [154] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. “LSTM fully convolutional networks for time series classification”. In: *IEEE access* 6 (2017), pp. 1662–1669.
- [155] Abdellah El Mekki, Afaf Bouhoute, and Ismail Berrada. “Improving driver identification for the next-generation of in-vehicle software systems”. In: *IEEE Transactions on Vehicular Technology* 68.8 (2019), pp. 7406–7415.
- [156] Erman Pattuk, Murat Kantarcioglu, Huseyin Ulusoy, and Bradley Malin. “Privacy-aware dynamic feature selection”. In: *2015 IEEE 31st international conference on data engineering*. IEEE. 2015, pp. 78–88.
- [157] Pierre Rougé, Ali Moukadem, Alain Dieterlen, Antoine Boutet, and Carole Frindel. “Anonymizing motion sensor data through time-frequency domain”. In: *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2021, pp. 1–6.
- [158] Maria Francesca Carfora, Fabio Martinelli, Francesco Mercaldo, Vittoria Nardone, Albina Orlando, Antonella Santone, and Gigliola Vaglini. “A “pay-how-you-drive” car insurance approach through cluster analysis”. In: *Soft Computing* 23 (2019), pp. 2863–2875.
- [159] Isabel Wagner and David Eckhoff. “Technical privacy metrics: a systematic survey”. In: *ACM Computing Surveys (CSUR)* 51.3 (2018), pp. 1–38.
- [160] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. “Amplification by shuffling: From local to central differential privacy via anonymity”. In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 2468–2479.
- [161] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.
- [162] Nidhi Dua, Shiva Nand Singh, and Vijay Bhaskar Semwal. “Multi-input CNN-GRU based human activity recognition using wearable sensors”. In: *Computing* 103.7 (2021), pp. 1461–1478.
- [163] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. “RescueDP: Real-time spatio-temporal crowd-sourced data publishing with differential privacy”. In: *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE. 2016, pp. 1–9.
- [164] Ergute Bao, Yin Yang, Xiaokui Xiao, and Bolin Ding. “CGM: an enhanced mechanism for streaming data collection with local differential privacy”. In: *Proceedings of the VLDB Endowment* 14.11 (2021), pp. 2258–2270.
- [165] Adrian-Silviu Roman, Roland Bolboacă, Sophie Cerf, Piroska Haller, and Bogdan Robu. “APUTrajGen: Adaptive Privacy and Utility Preserving Real-Time Synthetic Trajectory Generation”. In: *Under review* (2025).
- [166] Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. “A data mining approach to assess privacy risk in human mobility data”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 9.3 (2017), pp. 1–27.
- [167] Kassem Fawaz and Kang G Shin. “Location privacy protection for smartphone users”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 239–250.
- [168] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

- [169] Roland Bolboacă and Piroska Haller. “Enhanced long short-term memory architectures for chaotic systems modeling: An extensive study on the Lorenz system”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 34.12 (2024).
- [170] Yuzhou Jiang, Emre Yilmaz, and Erman Ayday. “Robust Fingerprint of Privacy-Preserving Location Trajectories”. In: *Proceedings on Privacy Enhancing Technologies* (2023).
- [171] Dan Yin and Qing Yang. “GANs Based Density Distribution Privacy-Preservation on Mobility Data”. In: *Security and Communication Networks* 2018.1 (2018), p. 9203076.
- [172] Luigi Libero Lucio Starace. “Porto taxi trajectories dataset”. In: (May 2020). DOI: [10.6084/m9.figshare.12302165.v1](https://doi.org/10.6084/m9.figshare.12302165.v1).
- [173] Kang-Tsung Chang. *Introduction to geographic information systems*. Vol. 4. McGraw-Hill Boston, 2008.
- [174] Igor Bisio, Fabio Lavagetto, Mario Marchese, and Andrea Sciarrone. “GPS/HPS-and Wi-Fi fingerprint-based location recognition for check-in applications over smartphones in cloud-based LBSs”. In: *IEEE transactions on Multimedia* 15.4 (2013), pp. 858–869.