UNIVERSITÉ DU LUXEMBOURG

PhD-FSTM-2025-87
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 14/07/2025 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Nesryne MEJRI
Born on 27 April 1993 in Ariana, Tunisia

UNSUPERVISED ANOMALY DETECTION FOR TYPE-AGNOSTIC AND CROSS-DOMAIN DEEPFAKE DETECTION

Dissertation defence committee

Dr Djamila AOUADA, dissertation supervisor
Associate Professor, Université du Luxembourg

Dr Domenico BIANCULLI, Chairman
Associate Professor, Université du Luxembourg

Dr Christian RIESS, Member
Professor, Friedrich-Alexander Universität, Germany

Dr Touradj EBRAHIMI, Member
Professor, École Polytechnique Fédérale de Lausanne, Switzerland

Dr Mohamed OURDANE, Member
Head of Cybersecurity department, POST, Luxembourg

Acknowledgments

No words could describe the gratitude I bear towards those with whom I have shared this journey, nor the heaviness in my heart for potentially parting ways with them, as this ride nears its end.

First and foremost, I would like to express my most heartfelt gratitude to my supervisors, Prof. Dr. Djamila Aouada and Prof. Dr. Enjie Ghorbel, with whom I have shared the most fulfilling parts of this journey and who will always have a special place in my life and in my heart. I can proudly say that they have both become my role models, not only for their sharp insight, rigorous mentorship, and endless support, but also as inspiring and strong women whose example has deeply shaped me, both personally and professionally. The same goes without saying for Dr. Anis Kacem, to whom I owe my sincerest gratitude for his suggestions, openness, touch of craziness, and continuous support throughout this thesis. I feel truly fortunate to have worked directly with the three pillars of CVI2. I will always remain grateful and admiring of their competence, resilience, honesty, patience, and unwavering support.

I extend my earnest appreciation to the defense committee members of my thesis, Prof. Dr. Domenico Bianculli, Prof. Dr. Touradj Ebrahimi, Dr. Mohamed Ourdane, and Prof. Dr.-Ing. Christian Riess for kindly accepting to evaluate the work submitted in this thesis, and therefore for dedicating their precious time and expertise reviewing it. I would also like to sincerely thank Prof Dr. Yves Le Traon, a member of the comité d'encadrement de thése (CET), for his involvement and valuable time and feedback throughout this work. A special thank you goes to Dr. Thanos Athanasiadis for his guidance on the path to doctoral studies

and his advice to join CVI2 when I first settled in Luxembourg.

I am also deeply grateful to my friends and colleagues, with whom I shared fun experiences, including outreach events, casual after-work gatherings, and road trips, such as Ahmet Serdar Karadeniz, Emilija Pashoska, Inder Pal Singh, Pavel Chernakov, Romain Hermary, and Vincent Gaudilliere. You were amazing to work with and to be around. I have enjoyed every absurd discussion we've ever had.

Overall, I am glad to have met and worked directly or indirectly with some of the current and the former members of CVI2, including Kostas, Mohamed Adel, Albert, Miguel, Laura, Michelle, Astrid, Dimitrios, Samet, Eya, Jose, Katarina, Parsa, Arun, Nassim, Peyman, Carl, Kseniya, Nidhal, Dan and Cosmin.

I would also like to acknowledge the funding sources, including Fonds National de la Recherche (FNR) and POST Luxembourg, for their generous financial support. The works presented in this thesis are funded under the projects: BRIDGES2021/IS/16353350/FaKeDeTeR, and UNFAKE ref.16763798.

Last but not least, I would like to thank my partner, Luc, who stood by me bravely throughout this entire journey. Your constant encouragement and unwavering support have helped me come this far. Thank you ever so much for your unconditional love. Finally, I extend my gratitude to my family and to my Tunisian friends Ryen, Hamza, Emir, Mehdi, Mohamed Ali, Raed, and Chawki. A very special acknowledgment goes to Haypha, whose suggestion to move to Luxembourg set me on the path that brought me to where I am today.

To everyone mentioned above and to those whose names I may have inadvertently overlooked, I thank you from the bottom of my heart. I am and will always be thankful for every experience and opportunity I have shared with you, or that came to me because of you. May your lives, and the lives of your loved ones, be long and filled with happiness.

Index

1 Introduction **3**

1.1 Motivation and Scope 5

1.2 Challenges 6

1.2.1 Forgery-related generalization for deepfake detection 6

1.2.2 Forgery-unrelated generalization for deepfake detection 7

1.3 Objectives and Contributions 8

1.3.1 Toward Type-Agnostic Unsupervised Deepfake Detection 8

1.3.2 Exploring Unsupervised Time-Series Anomaly Detection for Video Deepfakes Detection 9

1.3.3 Integrating Spatial Priors for Lightweight Unsupervised Temporal Deepfake Localization 10

1.3.4 Bridging Domain Gaps in Semantic Anomaly Detection using Unsupervised Domain Adaptation 11

1.4 Publications 12

1.5 Thesis Outline 13

2 Background **15**

2.1 Self-supervised Learning 15

2.1.1 The paradigm of Self-Supervised Learning 15

2.1.2 Context-based pretext tasks 16

2.1.3 Contrastive pretext tasks 18

2.1.4 Downstream task: Binary Classification 20
2.2 Unsupervised Anomaly Detection 21
2.3 Unsupervised Domain Adaptation for classification 24
2.4 Moment-matching Domain Adaptation 25
2.4.1 Adversarial Domain Adaptation 26

3 UNTAG: Learning Generic Features for Unsupervised Type-Agnostic Deepfake Detection 28
3.1 Introduction 29
3.2 Related works 32
3.3 A new paradigm for type-agnostic deepfake detection 33
3.4 UNTAG: Unsupervised type-agnostic deepfake detection 36
3.4.1 Self-supervised learning of generic and discriminative type-agnostic features 36
3.4.2 Unsupervised one-class classification with Gaussian Mixture Models . 39
3.5 Experiments 39
3.5.1 Experimental Protocol 39
3.5.2 Results 45
3.6 Conclusion 49

4 Unsupervised Anomaly Detection in Time Series: An Extensive Evaluation and Analysis of State-of-the-art Methods 51
4.1 Introduction 52
4.2 Preliminaries 57
4.2.1 Types of anomalies 57
4.2.2 Paradigms for anomaly detection in times-series 61
4.3 State-of-the-art on time-series anomaly detection 64
4.3.1 Clustering-based methods 64
4.3.2 Density-estimation methods 65
4.3.3 Distance-based methods 65

4.3.4 Reconstruction-based methods . . . 66
4.3.5 Forecasting-based methods . . . 66
4.3.6 Hybrid methods . . . 67
4.4 Datasets and Evaluation protocol . . . 67
4.4.1 Datasets . . . 67
4.4.2 Evaluation criteria . . . 69
4.4.3 Post-processing and Pre-processing . . . 73
4.4.4 Evaluated methods . . . 74
4.5 Results . . . 77
4.5.1 Performance using standard metrics . . . 77
4.5.2 Performance using revisited metrics for time-series . . . 80
4.5.3 Model stability . . . 82
4.5.4 Model size and memory consumption . . . 84
4.5.5 Generalization to different types of anomalies . . . 84
4.5.6 Discussion . . . 85
4.6 Conclusion . . . 88

5 Facial Region-Based Ensembling for Unsupervised Temporal Deepfake Localization 89

5.1 Introduction . . . 89
5.2 Unsupervised Anomaly Detection in Time Series for Deepfake Localization using Geometric Representations . . . 91
5.3 Facial region-based ensembling for unsupervised deepfake localization . . . 94
5.4 Experiments . . . 95
5.4.1 Experimental settings . . . 95
5.4.2 Results . . . 97
5.4.3 Discussion . . . 101
5.5 Conclusion . . . 102

6 When Unsupervised Domain Adaptation meets One-class Anomaly Detection: Addressing the Two-fold Unsupervised Curse by Leveraging Anomaly Scarcity 104

6.1 Introduction . . . 105
6.2 Related Works: Anomaly detection under domain shift . . . 108
6.3 The Two-fold Unsupervised Curse . . . 110
6.4 Rare Anomalies to the Rescue . . . 111
6.5 Methodology . . . 113
6.6 Experimental Results . . . 115
6.6.1 Experimental Setting . . . 115
6.6.2 Comparison against State-of-the-art. . . . 118
6.6.3 Additional Experiments . . . 119
6.7 Limitations and Future Work . . . 126
6.8 Conclusion . . . 127

7 Conclusion 130

7.1 Summary . . . 130
7.2 Future work . . . 132
7.2.1 Domain-adaptive unsupervised deepfake detection . . . 133
7.2.2 Vision-language models for explainable deepfake detection . . . 134
7.2.3 Extension to content-agnostic forgery detection . . . 134

List of Abbreviations

3D CNN 3D convolutional neural network.

AD Anomaly Detection.

AE autoencoder.

BCE Binary Cross Entropy.

CL Contrastive Learning.

DL Deep Learning.

DNN deep neural network.

FAM Facial Attribute Manipulation.

FR Facial Reenactment.

FS Face Swap.

FSF Fully Synthetic Face.

GMM Gaussian Mixture Model.

KDE Kernel Density Estimation.

MAE Mean Absolute Error.

MIM Masked Image Modeling.

MMD Maximum Mean Discrepancy.

MSE Mean Squared Error.

OC-SVM One-Class Support Vector Machines.

PCA Principal Component Analysis.

SSL Self-Supervised Learning.

SVDD Support Vector Data Description.

TL Transfer Learning.

TSAD Time Series Anomaly Detection.

List of Figures

1.1 Deepfake detection formulated as a binary classification task. 4
1.2 Examples of artifacts exhibited by different types of deepfakes depicting abnormalities affecting the semantics, the shapes and textures of the (a) background, (b) the eyebrows, (c) the eyes, (d) the nose, and (e) the mouth regions. 5

2.1 Illustration of a typical Self-Supervised Learning (SSL) framework. In Stage 1, the model is first pretrained on unlabeled data using an adequate pretext task. Then, in Stage 2, the model is fine-tuned on labeled data from a different downstream task. 16
2.2 Examples of transformations applied in the context of different pretext tasks, namely (a) Rotation prediction [48], (b) Jigsaw solving [50] and (c) colorization [58]. 18
2.3 Illustration of the contrastive pretext task of SimCLR [54], note that the figures are taken from [54]. 19

2.4 Various anomaly detection (AD) methods produce distinct types of decision functions, and their potential limitations. The decision regions are indicated by white and red for normal and anomalous areas, respectively. One-class classification techniques generally learn a discriminative boundary that separates normal from anomalous instances, but could be too loose leaving anomalies undetected (i). In contrast, probabilistic methods estimate the data distribution with the potential issue of possibly underfitting (or overfitting) the tails of a distribution (ii). Finally, reconstruction-based models aim to capture the intrinsic geometric structure of the data, such as a manifold or representative prototypes with the risk that Manifold or prototype structure artifacts leading to a good reconstruction of anomalies (iii). Figure from [63]. 22
2.5 Illustration of domain shift in unsupervised domain adaptation: the model is trained on labeled source data from the photo domain and evaluated on unlabeled target data from the cartoon domain, highlighting the challenge of generalizing across visually distinct domains despite the shared label space between both domains. 24
2.6 Illustration of MK-MMD domain alignment using multiple kernels. 26
2.7 Illustration of the DANN architecture [81]. 27

3.1 (1) The focus of most state-of-the-art deepfake detectors in terms of intra-type and inter-type generalization versus (2) the focus of the proposed method. (a) and (b) refer to the forgery types and datasets, respectively. FR, FS, FSF, and FAM refer to Facial Reenactment, Face Swap, Fully Synthetic Faces, and Facial Attribute Manipulation, respectively. Other is for unknown and stacked forgery types. 30

3.2 The proposed UNTAG framework is based on: (a) First, a self-supervised feature learning is considered: a pretext task learns implicitly artifact-sensitive region features by predicting the manipulated regions if any in an image; and (b) Second, an unsupervised generative one-class classifier is estimated using the self-supervised features of real images. 31
3.3 The transformations generated by R-splicer given a real image from ForgeryNet [26] 37
3.4 Common artifacts and their corresponding deepfake types. Images are randomly samples from StyleGAN2 [98] and FF++ [23]. 44
3.5 Heatmap of the Mean AUC scores, summarizing the overall inter-type generalization of UNTAG and the selected baselines. 45
3.6 t-Distributed Stochastic Neighbor Embedding (t-SNE) [127] visualizations of the real and fake embeddings for OC-FakeDect [96] and UNTAG. Darker points represent real samples, while lighter points correspond to fake samples. 50

4.1 Examples of the five different types of outliers proposed in [146]. 58
4.2 Overview of the different paradigms for anomaly detection in time-series: in contrast to clustering and probabilistic approaches, distance-based, reconstruction-based, and forecasting-based approaches take into account the temporal aspect. 61
4.3 An example of time-series from the UCR dataset, where the discord was calculated using DAMP [168]. 62
4.4 The evaluation process of (a) point-based anomalies versus (b) range-based anomalies. Range-based anomalies are characterized by partial overlap(s) with the ground truth. A more accurate evaluation for time-series should quantify the overlap in terms of **size**, **position**, and **cardinality**. 72

4.5 Application of the Point Adjustment (PA) on a given time-series: the Ground Truth (GT), the original prediction (Pred) and the prediction after Point Adjustment (PA) are reported. In this example, the performance of the algorithm without and with point adjustment is, respectively: Precision $= 0.75$, Recall $= 0.2$, F1-Score $= 0.32$, and Precision $= 0.92$, Recall $= 0.79$, F1-Score $= 0.85$. Best viewed in colors. 74
4.6 Average F1-Score on the five datasets comparing the non-PA and PA protocols. The non-hatched and hatched bars correspond to the mean F1-Score with and without Point Adjustment (PA), respectively. The vertical black line represents the standard deviation over five runs. 79
4.7 The mean performance per method on all datasets using the range-based metrics of [144], with different location biases. 80
4.8 Relation between the number of the parameters of the model and the number of features in the considered dataset. 83
4.9 The ratio of true anomalies detected for each tested method when varying the anomaly types. All methods succeeded in partially detecting each anomaly type, except MTAD-GAT which was unable to detect any collective trend anomaly. 83

5.1 Comparison of landmark displacement statistics extracted from fake and real videos in ForgeryNet [26]. 90
5.2 Overview of the proposed facial region-focused ensembling. (a) Extraction of facial landmark sequences to be used for training several individual Autoencoders (AEs), each trained on a specific facial region trajectories (i.e., nose and mouth). (b) Inference from the per-facial regions Autoencoders (AEs) and aggregation of the individual results via a voting strategy to produce the finale frame prediction. 92
5.3 Landmark detection fails on blurry images, despite the subject being real. . . 102

6.1 **Illustration of the two-fold unsupervised curse**: **(a)** The decision boundary learned from the source set without any adaptation does not allow generalization to the target domain. **(b)** Direct alignment of the unlabeled target with the one-class source features leads to the confusion of normal and abnormal samples. 106
6.2 **Comparison of our setting with previous works**: **(a)** supervised source anomaly detection with supervised domain adaptation [226], **(b)** unsupervised one-class source anomaly detection with few-shot domain adaptation [228, 230, 229], **(c) our considered setting**: unsupervised one-class source anomaly detection with unsupervised domain adaptation. 109
6.3 **Our Solution**: The top branch uses a trainable feature extractor with a DSVDD objective for one-class source data. The bottom branch clusters the features using a frozen CLIP visual encoder to identify the dominant feature cluster and align it with normal source features. ● are normals and ★ are anomalies . 113
6.4 Assessing the validity of anomaly scarcity assumption. 119
6.5 K-Means[245] components variation on VisDA. 122
6.6 (a) Sensitivity analysis. (b) AUC comparison for different visual encoders. . . 127
6.7 Histogram of anomaly scores for all classes of VisDA [236] (x-axis: anomaly score, and y-axis: count). 129

List of Tables

3.1 AUC (%) and Accuracy (Acc. in %) of UNTAG compared to the selected baselines on five different datasets. The best results are highlighted in **bold**. The second best results are underlined. The sub-blocks from top to bottom show supervised, self-supervised, and unsupervised methods, respectively. . 41

3.2 AUC (%) and Accuracy (Acc. (%)) on Face Swap (FS) deepfake generation methods. 42

3.3 AUC (%) and Accuracy (Acc. (%)) for Face Reenactment (FR) deepfake generation methods. 42

3.4 AUC (%) and accuracy (Acc. (%)) for Face Attribute Manipulation (FAM) deepfake generation methods. 43

3.5 AUC (%) and Accuracy (Acc. (%))for Fully Synthetic Faces (FSF) deepfake generation methods. 43

3.6 AUC (%) and Accuracy (Acc. (%)) for combined manipulations involving Face Swap (FS), Face Reenactment (FR), Fully Synthetic Faces (FSF), and Face Attribute Manipulation (FAM). 44

3.7 Specifications of the selected datasets, including the number of training and testing samples used in our experiments. An equivalent number of real images is selected from ForgeryNet [26] to avoid identity leakage [32, 122]. . . 44

3.8 Ablation on the role of each component in UNTAG, namely the pretext task as a direct binary classifier and the GMM as a direct One-Class Classification (OCC) as a Deepfake detector. 45

3.9 Performance of UNTAG under different backbones (Stage 1) when combined with a GMM as the one-class classifier in Stage 2. 46
3.10 Performance of UNTAG using a ResNet18 at Stage 1 while considering various one-class classifiers in Stage 2. 46
3.11 Mediapipe [124] landmark IDs per region. † denotes the full face region used to splice the background region. 47
3.12 The AUC performance of UNTAG using different number of Gaussian components for classification. Bold results highlight the best performance. 48
3.13 AUC (%) of background splicing on different datasets including diffusion, animals and portraits. 49

4.1 Comparison of existing evaluation studies of anomaly detection in time-series: we specify which of the following aspects were taken into account: (1) standard performance metrics which correspond to the precision, recall, and F1-score; (2) revisited performance metrics extending the precision, recall, and F1-score to time-series introduced by Tatbul et al. [144] ; (3) network size; (4) consideration of ML approaches in the comparison; (5) evaluation of recent deep learning techniques; (6) analysis with respect to the types of anomalies; and (7) use of a unified experimental protocol. Note that by “partial", we mean that the authors briefly discussed the concept without necessarily producing any related comparison or results in their study. 53
4.2 Summary of the five datasets considered in the experiments. The percentage of anomalies in the testing set is reported. 69
4.3 Paradigm type and nature of evaluated methods 69

4.4 Results in terms of traditional performance metrics of evaluated state-of-the-art methods (precision P, recall R, F1-score) on the 5 considered datasets **without Point Adjustment (PA)**. The experiments have been performed 5 times for each algorithm and dataset. The mean and standard deviation are reported. The bold and underlined results correspond to the first and second-best F1-Score, respectively. 75
4.5 Results in terms of traditional performance metrics of evaluated state-of-the-art methods (precision (P), recall (R), F1-score (F1)) on the 5 considered datasets **with Point Adjustment (PA)**. The experiments have been performed 5 times for each algorithm and dataset. The mean and standard deviation are reported. The bold and underlined results correspond to the first and second-best F1-Score, respectively. 76
4.6 Number of parameters and model size (MB) of the trained models on different datasets 84
4.7 The *flat-bias* performance (in %) of the tested methods on the five benchmarks using the metrics proposed by Tatbul et al. [144]. The average and the standard deviation of five runs are reported. 85
4.8 The *front-bias* performance (in %) of the tested methods on the five benchmarks using the metrics proposed by Tatbul et al. [144]. The average and the standard deviation of five runs are reported. 86
4.9 The *middle-bias* performance (in %) of the tested methods on the five benchmarks using the metrics proposed by Tatbul et al. [144]. The average and the standard deviation of five runs are reported. 86
4.10 The *back-bias* performance (in) %) of the tested methods on the five benchmarks using the metrics proposed by Tatbul et al. [144]. The average and the standard deviation of five runs are reported. 87

5.1 Results in terms of standard performance metrics on ForgeryNet under the PA protocol. 97

5.2 Results in terms of standard performance metrics on ForgeryNet under the non-PA protocol. 98
5.3 Results in terms of range-based metrics (t-Precision, t-Recall and t-F1-score) proposed in [144] on ForgeryNet under the non-PA protocol. 99
5.4 Results using range-based metrics proposed in [144] on ForgeryNet under the PA protocol. 100
5.5 Results using individual facial regions on ForgeryNet in terms of standard performance metrics under the non-PA protocol. 100
5.6 Feature combination versus ensembling strategy of the three most relevant facial regions under the non-PA protocol. Experiments are performed on ForgeryNet. 101
5.7 Number of model parameters 102

6.1 Ten-run average and standard deviation of AUC (%) on the Office datasets [235, 234]. 116
6.2 AUC (%) on the target domain of our UDA anomaly detector on VisDA [236] compared **with various adaptation paradigms** (from zero-shot, i.e., pre-trained Visual encoders, few-shot, to supervised, i.e., Oracle). 118
6.3 Ablation on the components of the proposed method. 120
6.4 Clustering ablation. GMM and K-means use 10 components for VisDA and 2 for other datasets. $k = 2$ for VisDA and $k = 1$ for the remaining datasets. . . . 122
6.5 Performance in terms of AUC (%) using different domain adaptation losses on VisDA, Office31 and OfficeHome. 123
6.6 AUC (%) of Domain Generalization (DG) for anomaly detection, trained ***ONLY*** on the source domain Photo (Ph.) and tested on unseen domains. DA means Domain Adaptation. 123
6.7 Our UDA approach on two anomaly detection methods [70, 247]. ZS and Src mean Zero-shot and Source only. 123

6.8 AUC performance on VisDA [236] of different trainable feature extractors using our method, in comparison against the source-only-trained model. 124
6.9 VisDA performance with different visual encoders. **w/o CLIP** means the ϕ is self-trained. 125
6.10 Anomaly detection performance on the VisDA [236] dataset for the setting f:R50 + ψ:CLIP-ViT-B32 using additional metrics, such as Accuracy (Acc.), Balanced Accuracy (B.acc.), Precision (P), and Recall (R). 126

Abstract

Deepfakes are visual media created using deep learning models to partially manipulate or fully synthesize human faces. They cover a variety of forgery methods, broadly categorized into four types: face swaps, facial reenactments, facial attribute manipulations, and synthetic face generation. Their increasing realism in recent years has raised concerns regarding their misuse, thereby creating an urgent need for reliable deepfake detection techniques. Hence, various deepfake detection methods have been proposed. They are predominantly based on Deep Neural Networks trained in a supervised manner. As a result, these methods are often prone to generalization issues; hindering their applicability in real-world settings. Although promising, their performance degrades considerably when encountering unseen images/videos that differ significantly from the training data. This drop can be attributed to two different sources of variation in the visual data: forgery diversity and environmental variability.

Herein, this thesis aims to enhance the robustness of deep-fake detectors to these two complementary sources of variation by reformulating deepfake detection as an unsupervised anomaly detection (UAD) task. This formulation eliminates the need for annotated fake data and reduces the dependence on specific types of forgeries.

In the first part of the manuscript, we focus on improving the generalization of deepfake detectors to unseen forgeries. While some research works have attempted to improve this aspect, they have mainly targeted blending-based artifacts typically induced by several face-swaps generation techniques. As a consequence, they usually show degraded performance when dealing with non-blending-based deepfakes, including diffusion-based face-swaps and

other deepfake types characterized by inherently different artifacts. To address this issue, we propose a self-supervised framework that allows extracting features from different artifact-prone regions. This self-supervision mechanism is then coupled with a one-class classifier that models the feature distribution of real data only, thereby avoiding overfitting specific types of deepfake artifacts. This idea has then been extended to temporal deepfake localization, where the goal is to spot specific frames in an unsegmented stream that have undergone deepfake manipulations. Experiments performed on several benchmarks have demonstrated the improved generalization capabilities of the proposed methods.

In the second part of this work, we also address the lack of robustness to environmental changes. Deep learning models, including deepfake detectors, are often challenged by the domain shift issue, a phenomenon frequently caused by uncontrolled variations in lighting, resolution, or background. These variations are typically unrelated to forgeries and can further compromise the detection performance of deepfake detectors. In the literature, this problem has often been mitigated by adopting an Unsupervised Domain Adaptation (UDA) approach. Nevertheless, existing UDA techniques are primarily designed for binary and multi-class classification, while being incompatible with the core proposed paradigm for deepfake detection, namely, unsupervised anomaly detection. Indeed, UDA for UDA is an ill-posed problem due to what we call the two-fold unsupervised curse. To overcome this issue, we propose to take advantage of the scarcity of anomalies and rely on a clustering technique to isolate a predominant cluster to be used for the alignment step. This pioneering work has been validated on anomaly detection benchmarks, showing great potential for enhancing the generalization of deepfake detectors.

Chapter 1

Introduction

Visual forgery in photographs is not a recent phenomenon. Throughout history, photographs have been deliberately manipulated to alter facts, influence the public's opinion, or spread misinformation [1]. Forgery users recognized the powerful impact of the saying "seeing is believing" and leveraged it to their advantage, despite the effort and expertise required to produce these manipulations.

Technological advancements, particularly the development of affordable digital cameras and personal computers, have enabled easy access to digital photographs and visual forgery, respectively. Software such as Photoshop made photo editing easier, yet human expertise remained essential. Consequently, detecting these manipulations has been of great interest in the field of Computer Vision, particularly in Multimedia Forensics. Traditional approaches often rely on physics-based or geometrical models [2, 3], or on the analysis of artifacts such as compression inconsistencies [4] and sensor noise patterns [5], to determine the authenticity of a given photo.

As Generative Deep Learning (DL) grew more popular in recent years, visual facial forgeries became more realistic and even more accessible. Examples of DL-powered tools that completely eliminated the need for expert intervention include DeepfaceLab [6] and Faceswap [7]. This has led to what is known today as "Deepfakes", which can be defined as digital content (images, audios, videos) that is partially or entirely synthesized by a deep learning algorithm. Deepfakes cover a broad scope of facial manipulations, that range from

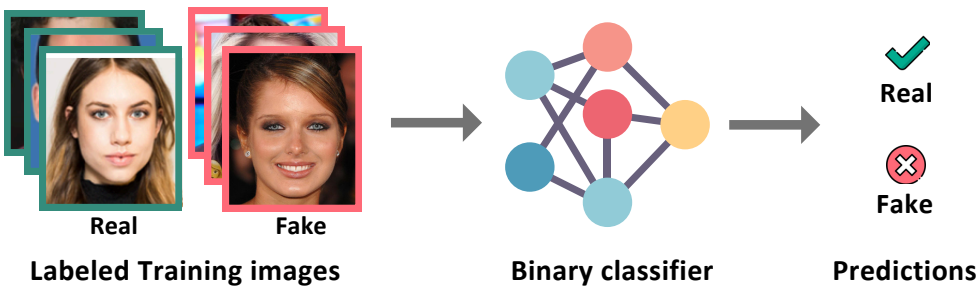
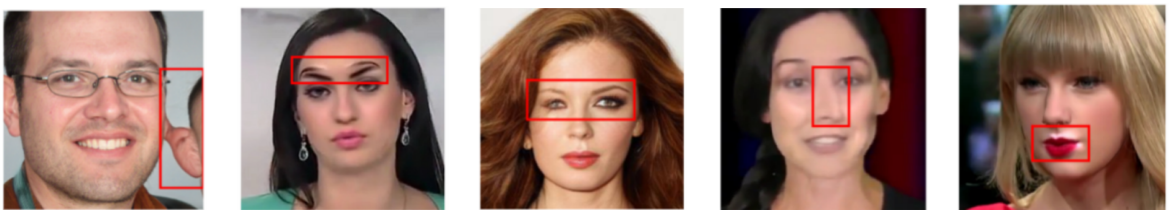


Figure 1.1: Deepfake detection formulated as a binary classification task.

generating entirely synthetic personas to applying subtle edits such as altering eye color. Despite the creative applications they introduced in education [8], entertainment [9], marketing [10], and filmmaking [11], deepfakes also present risks that cannot be overlooked. In fact, they can facilitate malicious activities such as identity theft [12], political misinformation [13, 12], fraud [14], and the spread of non-consensual content [15, 16], which can undermine the public's trust in digital media and negatively impact individuals, organizations, and even entire nations.

Given these threats, numerous deepfake detection techniques have been introduced in the literature, with a clear predominance of deep learning-based methods. These approaches typically formulate deepfake detection as a supervised binary classification problem, where the goal is to distinguish between genuine and forged content. These detectors are generally trained on large annotated datasets, where the training samples are labeled as real or fake, as shown in Figure 1.1. This supervised setup encourages the detector to identify small inconsistencies unique to the fake training data, in comparison to real visuals. Indeed, deepfakes cannot fully reflect the behavior and appearance of a genuine person, leaving detectable traces that detectors can learn to identify. These inconsistencies, known as *artifacts*, vary significantly across different deepfake types and generation methods. For instance, they can occur in the spatial domain, such as inconsistencies in colors, lighting, or object shapes (shown in Figure 1.2), in the temporal domain as jittery motions (i.e., head and mouth movements), and in the frequency domain as abnormal noise patterns and high-frequency anomalies.

In this chapter, we begin by presenting the motivation and the scope of this work. Next, we introduce our primary objectives and contributions to the topic of deepfake detection. Finally, we conclude this chapter by listing the publications resulting from the investigations of this thesis.



(a) Deformation (b) Odd texture (c) Asymmetry (d) Misplacement (e) Discoloration

Figure 1.2: Examples of artifacts exhibited by different types of deepfakes depicting abnormalities affecting the semantics, the shapes and textures of the (a) background, (b) the eyebrows, (c) the eyes, (d) the nose, and (e) the mouth regions.

1.1 Motivation and Scope

The detection performance of most existing detectors is generally satisfactory when evaluated on deepfake data drawn from the same distribution as the training samples. However, this performance deteriorates significantly when exposed to unseen data settings. This limitation, known as lack of generalization, can be attributed to the use of deep neural networks (DNNs), which often struggle to generalize beyond their training conditions [17]. This limitation restricts their real-world applicability and highlights the need for more robust detection strategies. To address this issue, tremendous efforts have been made to improve the robustness of deepfake detection methods to unseen face swap generation methods [18, 19, 20]. These approaches mainly try to model blending artifacts that are common to multiple face-swapping techniques. Nevertheless, it is worth noting that more recent face swap methods, such as diffusion-based approaches, do not incorporate such traces. These cues are also absent in other types of deepfakes, exhibiting completely different inconsistencies. As a direct consequence, existing detection methods are still failing to detect unseen deepfakes regardless of their nature or type. Moreover, as they rely on deep learning frameworks,

state-of-the-art deepfake detection methods are also prone to the domain shift problem, where training and testing data are acquired under different environmental settings.

Therefore, our goal is to improve the generalization capabilities of deepfake detectors to get closer to real-world requirements.

1.2 Challenges

In this thesis, we argue that enhancing the generalization capabilities of deepfake detectors necessitates categorizing the source of unseen information within the raw data to apply suitable strategies. More specifically, we posit that the lack of generalization of deepfake detectors originates from both forgery-related and forgery-unrelated factors, which occur at the artifact level and the domain level, respectively. Artifact-level generalization issues can be associated with the detector overfitting visual cues specific to forgery methods encountered during training. In contrast, domain-level generalization issues refer to the model's sensitivity to variations in factors unrelated to the forgery itself, such as lighting, resolution, or subject identity.

Although this categorization enables a more structured way of approaching the generalization problem, tackling each source remains challenging. Therefore, in the subsequent subsections, we discuss the challenges associated with addressing the artifact and the domain-level generalization issues, respectively.

1.2.1 Forgery-related generalization for deepfake detection

Significant efforts have been made to improve generalization at the artifact level. These works often train DNN architectures, such as XceptionNet [21] and EfficientNet [22], in a supervised manner on large-scale annotated datasets [23, 24, 25, 26]. In particular, recent studies extensively rely on blending artifact modeling [20, 27, 28, 19, 28, 29, 18, 30] typically observed in face-swapping forgeries. However, since these methods are supervised, they tend to overfit the training artifacts. As a result, they are ineffective on forgeries where these artifacts are absent. In fact, deepfakes are not limited to face swap forgeries. They

are broadly categorized into four types: Face Swap (FS), Facial Reenactment (FR), Facial Attribute Manipulation (FAM), and Fully Synthetic Face (FSF) [31], each associated with different artifacts. This diversity poses a significant challenge in forgery-related generalization, as artifacts are often difficult to model: they can be subtle, and their characteristics vary considerably across deepfake types. They may occur in localized facial regions or be spread across an entire image, and they can exist within the spatial, temporal, or frequency domains, or in a combination of them. In many cases, they are non-interpretable, resembling noise or manifesting as physical or anatomical inconsistencies. This complicates defining, interpreting, and modeling them reliably, especially since they are continuously evolving with the development of novel deepfake generation methods.

On a practical level, artifact variability makes it difficult to build representative datasets, as this would require exhaustive collection, annotation, and continuous updates to incorporate new deepfakes with novel artifacts.

The inherent variability of artifacts and the limited representativeness of annotated training datasets, therefore, reveal the fundamental limitations of supervised learning in building a unified and generalizable solution across different deepfake types.

1.2.2 Forgery-unrelated generalization for deepfake detection

As discussed earlier, deepfake detectors suffer from domain-level generalization issues, which can mainly be attributed to their reliance on deep learning models. In fact, detectors can overfit domain-specific characteristics, such as lighting, pose, or background, which are unrelated to actual forgeries and artifacts [32, 33]. As a result, their performance degrades significantly when these domain attributes differ between the training and testing sets. While identifying and labeling domain-specific attributes might seem like a viable solution, it quickly becomes impractical, as deepfakes often appear in highly diverse settings, and exhaustive labeling would require significant time and resources.

This challenge highlights the need to take into account the robustness of deepfake detectors to unseen domains, as even a cross-type generalizable deepfake detector may still fail in the presence of domain shift [34].

1.3 Objectives and Contributions

This thesis aims to enhance the generalization of deepfake detectors at both the artifact and domain levels. To address the limited generalization across deepfake types, we propose reformulating the problem as an unsupervised anomaly detection task. This formulation enables treating deepfakes as anomalies regardless of their type and artifacts. Then, we focus on strengthening this formulation to improve robustness to unseen domains. The following subsections summarize the works developed during this thesis, outlining how these contributions help achieve our objectives.

1.3.1 Toward Type-Agnostic Unsupervised Deepfake Detection

Our first contribution, entitled UNTAG, addresses the artifact-level generalization issues of deepfake detectors. It introduces a novel two-stage framework for unsupervised, type-agnostic deepfake detection. Most existing detectors rely on supervised learning and are trained on a limited set of forgery types, typically face swaps and facial reenactments, which restricts their ability to detect unseen deepfakes. To address this, Self-Supervised Learning (SSL) is investigated for enhancing the generalization of deepfake detectors. Training SSL approaches includes two stages: pretraining on a generic pretext task, followed by fine-tuning on labeled deepfake datasets. While SSL techniques achieved improved robustness to novel deepfake generation methods, they still had two key shortcomings. First, the pre-training task is often unrelated to deepfake detection, resulting in the learning of suboptimal feature representations. Second, the fine-tuning stage remains supervised, making these methods prone to overfitting specific artifacts and forgery types.

Given these limitations, we propose a self-supervised approach tailored specifically for deepfake detection. Unlike generic pretraining, our first stage learns features that focus on artifact-prone regions, making the representations more relevant to the task. In the second stage, supervised fine-tuning is replaced by an unsupervised anomaly detection task. In other words, we fit a one-class classifier directly on the learned feature of the genuine image, thus modeling the distribution of real data without requiring any labeled deepfakes. This

reduces the risk of overfitting specific artifacts while enabling type-agnostic deepfake detection. Our results show that UNTAG outperforms supervised, self-supervised, and other unsupervised approaches, achieving better generalization without relying on labeled fake data. This emphasizes the effectiveness of our formulation for fully unsupervised, type-agnostic deepfake detection. UNTAG has been published in [35].

1.3.2 Exploring Unsupervised Time-Series Anomaly Detection for Video Deepfakes Detection

While UNTAG improved type-agnostic deepfake detection, it remains limited to deepfake image data, overlooking the fact that deepfakes frequently occur as videos. Although it conceptually applies to individual video frames, it cannot capture temporal artifacts, which can be indicative of forgeries. Existing video deepfake detectors [36, 37, 38, 39, 40] address this problem through supervised training on annotated deepfake datasets, using 3D convolutional neural networks (3D CNNs) [41] and transformers [42] to model temporal dependencies. Nevertheless, these methods rely on extensive labeled data, making them costly and prone to poor generalization to unseen artifacts and deepfake types. Moreover, they often assume that videos are either entirely real or entirely fake, thereby overlooking partially manipulated videos, which require temporally localizing the manipulated frames. This task of localizing specific forged frames aligns with unsupervised multivariate Time Series Anomaly Detection (TSAD), particularly since videos can be interpreted as multivariate pixel trajectories.

Therefore, to extend the UAD formulation to temporal data, i.e., videos, we conducted a systematic study of unsupervised multivariate TSAD methods to investigate their suitability for video deepfake detection. This has been motivated by the fact that existing TSAD comparative studies often adopt inconsistent evaluation protocols and focus on standard performance metrics, such as precision and recall, which fail to capture the temporal aspect of anomalies. Furthermore, practical aspects such as model stability, computational cost, and robustness across different types of anomalies are frequently neglected. To address these issues, we performed a comprehensive evaluation study of recent unsupervised time-series

anomaly detection techniques, incorporating range-based metrics and a unified protocol to better assess their real-world applicability to tasks such as video-based deepfake detection. This work has been published [43].

1.3.3 Integrating Spatial Priors for Lightweight Unsupervised Temporal Deepfake Localization

Based on our previous evaluation study, we employ the time-series anomaly detection formulation for unsupervised deepfake localization in videos. Since standard deepfake detection techniques are trained only on annotated datasets depicting entirely real or entirely fake videos, they have restricted applicability on untrimmed videos. In other words, these detectors cannot detect forgeries at the frame level in live streams or partially manipulated videos. Attempts to solve this issue are often trained in a supervised manner on multimodal videos, where video and audio inputs are jointly processed to verify their synchronization and localize forged frames. However, these approaches suffer from several drawbacks; first, the supervision can result in poor generalization capabilities. Second, deepfakes typically do not include audio, and even when they do, they are likely to be forged as well. Finally, processing audio-visual data results in large and cumbersome model architectures.

To mitigate the aforementioned issues, we reformulate the task as an unsupervised multivariate time-series anomaly detection problem, enabling type-agnostic deepfake detection. Aside from requiring only real data for training, our approach does not depend on audio data, making it more applicable under realistic scenarios. Furthermore, to avoid large computational requirements, we use geometric facial representations i.e., facial landmarks, as multivariate time-series input instead of directly modeling raw pixel data. Finally, our approach introduces an ensembling strategy for tracking artifact-prone facial regions. Our experiments, conducted on the ForgeryNet dataset [26], highlighted the relevance of our approach, making it the first lightweight and fully unsupervised method for deepfake localization. This work has been published in [44].

1.3.4 Bridging Domain Gaps in Semantic Anomaly Detection using Unsupervised Domain Adaptation

All our previous work established Unsupervised Anomaly Detection (UAD) as a more suitable alternative to supervised learning for generalizable image and video deepfake detection. By training a one-class classifier only on normal data, this formulation alleviates the need for expensive annotated fake data and improves robustness to multiple types of generation methods. However, like most deep learning-based techniques, it remains sensitive to domain shift [34]. For instance, a one-class classifier trained on indoor faces may detect outdoor ones as outliers due to unseen variations in the lighting conditions. Consequently, despite being effective in a single domain, the UAD formulation may fail in the presence of a domain gap, which restricts its applicability to real-world scenarios.

Existing approaches tackling domain shift in visual unsupervised anomaly detection primarily include few-shot domain adaptation methods. These approaches are trained jointly on a one-class source set and a small set of annotated real samples from the target domain, ensuring minimal exposure to the target domain's distribution. Although this strategy may seem adequate, it remains constrained by the need for labeled target-domain data and the risk of insufficient representativeness of the target domain characteristics. Alternatively, in standard multi-class classification tasks, Unsupervised Domain Adaptation (UDA) has shown more promising results against domain shift. By leveraging a suitable alignment objective and a larger, but unlabeled, target domain set, UDA enables implicit feature distribution alignment without requiring any labels. Nevertheless, applying UDA to one-class anomaly detectors is not straightforward, as both tasks are unsupervised, leading to what we define as the two-fold unsupervised curse, an open challenge that has never been addressed in the literature.

To overcome this issue, we propose a novel UDA framework tailored specifically for one-class-based UAD. Our approach assumes that anomalies are rare and leverages clustering techniques to identify target-normal domain data, enabling their alignment with the one-class source real data. Extensive experiments on standard adaptation benchmarks validate the effectiveness of this framework, emphasizing its potential to address UDA with UAD. This

work, which lays the foundation for domain-adaptive unsupervised deepfake detection, is currently under review.

1.4 Publications

JOURNALS

1. **Mejri, N.**, Lopez-Fuentes, L., Roy, K., Chernakov, P., Ghorbel, E. and Aouada, D., 2024. Unsupervised anomaly detection in time-series: An extensive evaluation and analysis of state-of-the-art methods. Expert Systems with Applications, p.124922.

2. **Mejri, N.**, Ghorbel, E., Kacem, A., Chernakov, P., Foteinopoulou, N., and Aouada, D., 2025, Unsupervised Domain Adaptation with One-class Anomaly Detection, under preparation for submission.

CONFERENCES

1. **Mejri, N.**, Ghorbel, E. and Aouada, D., 2023, June. Untag: Learning generic features for unsupervised type-agnostic deepfake detection. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). IEEE.

2. **Mejri, N.**, Chernakov, P., Kuleshova, P., Ghorbel, E. and Aouada, D., 2024, July. Facial Region-Based Ensembling for Unsupervised Temporal Deepfake Localization. In 2024 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). IEEE.

3. **Mejri, N.**, Ghorbel, E., Kacem, A., Chernakov, P., Foteinopoulou, N., and Aouada, D., 2025, When Unsupervised Domain Adaptation meets One-class Anomaly Detection: Addressing the two-fold unsupervised curse by leveraging anomaly scarcity, under review, submitted to the Conference on Neural Information Processing Systems 2025.

PUBLICATIONS NOT INCLUDED IN THIS THESIS

1. Singh, I.P., **Mejri, N.**, Nguyen, V.D., Ghorbel, E. and Aouada, D., 2023. Multi-label deepfake classification. 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP).

2. Nguyen, D., **Mejri, N.**, Singh, I.P., Kuleshova, P., Astrid, M., Kacem, A., Ghorbel, E. and Aouada, D., 2024. LAA-Net: Localized Artifact Attention Network for Quality-Agnostic and Generalizable Deepfake Detection. IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR).

3. Karadeniz, A.S., Mallis, D., **Mejri, N**., Cherenkova, K., Kacem, A. and Aouada, D., 2024. DAVINCI: A Single-Stage Architecture for Constrained CAD Sketch Inference. Proceedings of the British Machine Vision Conference (BMVC).

4. Karadeniz, A.S., Mallis, D., **Mejri, N.**, Cherenkova, K., Kacem, A. and Aouada, D., 2025. Picasso: A feed-forward framework for parametric inference of CAD sketches via rendering self-supervision. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV).

1.5 Thesis Outline

This dissertation is organized as follows:

- **Chapter 2:** This chapter introduces the necessary background for understanding the contributions of this thesis. It focuses primarily on Unsupervised Learning concepts such as Unsupervised Anomaly Detection (UAD), Self-Supervised Learning (SSL), and Unsupervised Domain Adaptation (UDA).

- **Chapter 3:** This chapter introduces UNTAG, a two-stage framework for type-agnostic deepfake detection which combines a self-supervision mechanism with one-class UAD.

- **Chapter 4:** This chapter extends the formulation of UNTAG to the temporal domain, where deepfakes are known for showing notable temporal inconsistencies. The chapter presents a general evaluation study of existing Unsupervised Time Series Anomaly Detection (TSAD) approaches with the aim of assessing their applicability to real-world applications, such as Deepfake Detection.
- **Chapter 5:** This chapter builds on the findings of the previous chapter and tackles the task of Unsupervised Deepfake Temporal Localization, where a facial-region-based ensemble for detecting partially manipulated videos is introduced.
- **Chapter 6:** In this chapter, we investigate the applicability of Unsupervised Domain Adaptation (UDA) for UAD problems and propose a framework for solving the doubly unsupervised nature of the tasks for general image classification.
- **Chapter 7:** This final chapter summarizes the work presented in this thesis and further discusses future perspectives.

Chapter 2

Background

This chapter covers the background required to understand the contributions presented in this thesis. Specifically, we begin by introducing self-supervision, a representation learning paradigm that helps capture discriminative features exclusively from unlabeled data. Second, we introduce unsupervised anomaly detection, the core notion supporting all our contributions, and enabling cross-type deepfake detection. Finally, we introduce Unsupervised Domain Adaptation (UDA) for binary and multiclass image classification.

2.1 Self-supervised Learning

This section introduces the Self-Supervised Learning (SSL) paradigm and outlines two of its prominent subcategories, namely context-based methods and contrastive learning. Subsequently, representative pretext tasks used in SSL, along with their associated downstream tasks, are described.

2.1.1 The paradigm of Self-Supervised Learning

Self-Supervised Learning (SSL), initially introduced in [45], is a subcategory of unsupervised learning [46] that enables learning generic feature representations in contexts where large-scale unlabeled data is available, but whose annotation process can be impractical or costly.

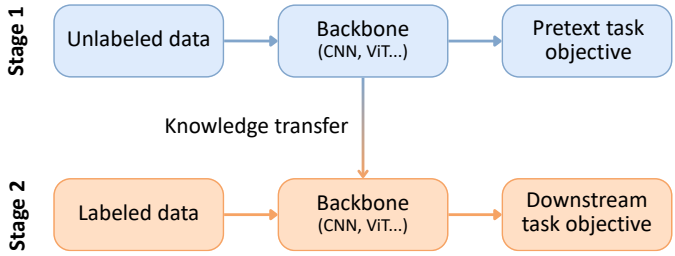


Figure 2.1: Illustration of a typical Self-Supervised Learning (SSL) framework. In Stage 1, the model is first pretrained on unlabeled data using an adequate pretext task. Then, in Stage 2, the model is fine-tuned on labeled data from a different downstream task.

SSL typically follows a two-stage framework consisting of a pretraining phase followed by a fine-tuning phase as depicted in Figure 2.1. The first stage, known as the pretext task, has an auxiliary objective, whose goal is learning transferable features directly from the unlabeled data without needing any human annotations [47]. More specifically, pretext tasks require deriving pseudo-labels deterministically from intrinsic properties of the unlabeled data, such as spatial structure [48, 49, 50], temporal continuity [51, 52, 53], or invariance under various data augmentations [54, 55, 56, 57]. The second stage typically utilizes the learned representations as a starting point for fine-tuning a variety of downstream tasks, such as image classification or segmentation. This often results in faster convergence and performance comparable to supervised learning methods, while reducing the risk of overfitting, particularly when the downstream task training data is limited [46].

According to Gui et al. [46], SSL methods can be classified into three major categories: context-based methods, Contrastive Learning (CL) methods, and Masked Image Modeling (MIM). In the following subsections, we focus primarily on the first two categories, namely context-based and contrastive SSL, as they have been used in the contributions of this thesis.

2.1.2 Context-based pretext tasks

Context-based SSL defines pretext tasks in which the input data is deliberately transformed, and the model is trained to infer a property of the transformation or to recover missing infor-

mation. Since these transformations are predefined and deterministic, they allow associating each transformed data point with a corresponding pseudo-label without requiring manual annotations. Classical pretext task examples include rotation prediction [48], where the input is rotated by fixed angles and the model is trained to classify the applied rotation; jigsaw puzzle solving [50], where the input is divided into shuffled patches and the model predicts their original arrangement; and colorization [58], where the image chrominance channels are removed and the model is trained to restore them from the grayscale input. Representative transformations for each task are illustrated in Figure 2.2.

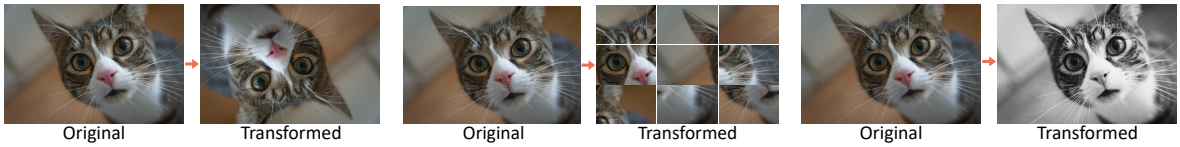
Formally, for a given input space $\mathcal{X}_i$ (e.g., image or video space), and an unlabeled dataset $\mathcal{D} = \{\mathbf{X}\}_{i=1}^{N} \subset \mathcal{X}$, consider a finite set of K transformations $\mathcal{T} = \{T_j\}_{j=1}^{K}$. Each transformation T_j is associated with a task-specific pseudo-label $y_j \in \mathcal{Y}$. In classification-based pretext tasks, such as rotation prediction [48] or jigsaw solving [50], $y_j = j$, corresponding to the index j of the transformation itself, leading to a label space $\mathcal{Y} = \{1, \ldots, K\}$. In regression-based pretext tasks, such as colorization [58], the pseudo-label is directly derived from the original input, such as the image's chrominance channels, and resides in the vector space $\mathcal{Y} = \mathbb{R}^{H \times W \times 2}$. Therefore, the augmented dataset for the pretext task is expressed as:

$$\mathcal{D}^{\mathsf{Aug}} = \{(T_j(\mathbf{X}_i),\ y_{i,j})\}_{i=1,j=1}^{N,K}\,.$$

Let $\zeta_\theta : \mathcal{X} \to \mathcal{Y}$ be a neural network model with parameters θ, which maps each transformed sample of $\mathcal{D}^{\mathsf{Aug}}$ to its corresponding pseudo-label. The training objective of context-based SSL approaches is formalized as:

$$\min_{\theta}\ \frac{1}{NK}\sum_{i=1}^{N}\sum_{j=1}^{K}\mathcal{L}\left(\zeta_\theta(T_j(\mathbf{X})),\ y_j\right),$$

where $\mathcal{L}$ denotes an appropriate loss function. For classification pretext tasks, cross-entropy loss is commonly used, whereas regression tasks typically rely on losses such as Mean Absolute Error (MAE) or Mean Squared Error (MSE) [59].



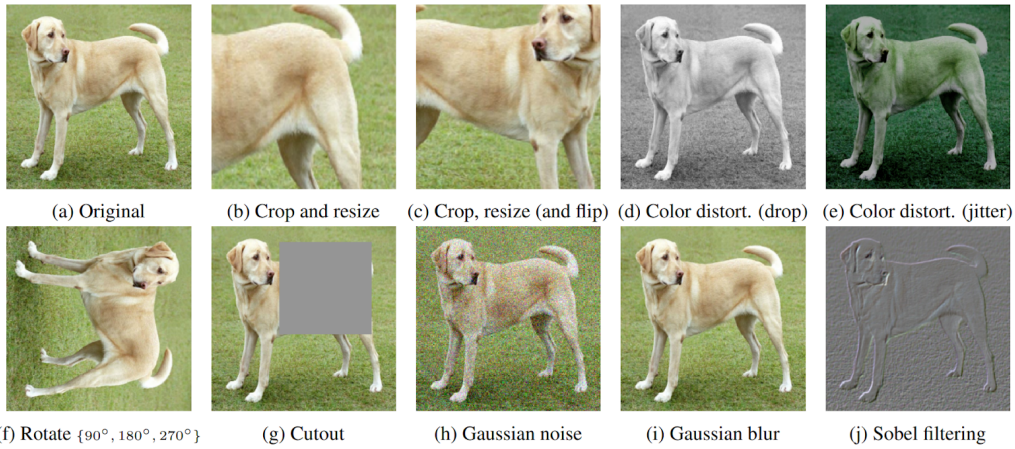
(a) Rotating (b) Spatial shuffling (c) Grayscaling

Figure 2.2: Examples of transformations applied in the context of different pretext tasks, namely (a) Rotation prediction [48], (b) Jigsaw solving [50] and (c) colorization [58].

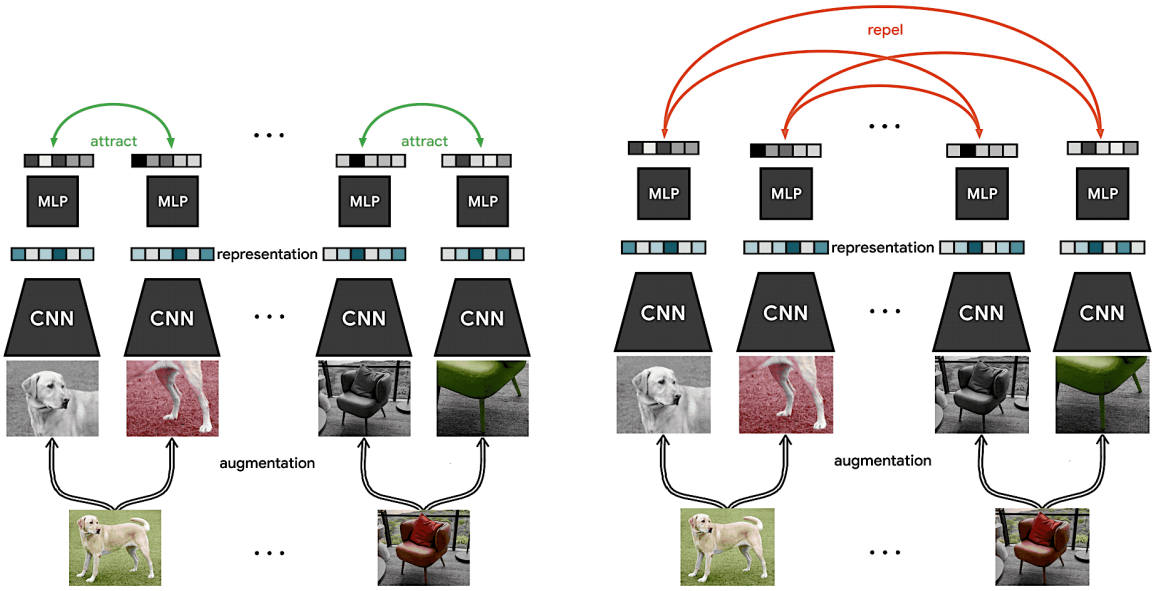
2.1.3 Contrastive pretext tasks

As described in the previous section, context-based self-supervised learning focuses on predicting properties of applied transformations, which makes the model sensitive to their presence. By contrast, contrastive self-supervised learning adopts a fundamentally different strategy: it seeks to learn feature representations that are invariant to such transformations by identifying correspondences between multiple views of the same input. A view refers to a transformed version of the original unlabeled instance as shown in Figure 2.3 (A). Rather than associating each transformed sample with an explicit pseudo-label, contrastive methods construct pairs of input data where different augmented views of the same data point are treated as similar (*positive pairs*), whereas views from different data points are considered dissimilar (*negative pairs*). The core objective is therefore to learn an embedding space in which augmented views of the same instance are mapped to similar representations, while views from different instances are mapped to dissimilar representations. This is illustrated in the contrastive framework of Chen et al. [54] in Figure 2.3 (B)-(C).

Specifically, let $\mathcal{X}$ denote the input space, and let $\mathcal{D} = \{\mathbf{X}_i\}_{i=1}^{N} \subset \mathcal{X}$ be a dataset of N unlabeled samples. Let $\mathcal{T} = \{T_j : \mathcal{X} \to \mathcal{X}\}_{j=1}^{K}$ denote a finite set of stochastic transformations, such as random cropping, color jittering, or horizontal flipping, as illustrated in Figure 2.3 (A). Let $T^{(1)} = T_1^{(1)} \circ T_2^{(1)} \circ \cdots \circ T_K^{(1)}$ and $T^{(2)} = T_1^{(2)} \circ T_2^{(2)} \circ \cdots \circ T_K^{(2)}$ denote two independently sampled compositions of the transformations in $\mathcal{T}$, where each $T_j^{(1)}$ and $T_j^{(2)}$ being the same transformation T_j applied with independently sampled random parameters. For each input $\mathbf{X}_i$, these compositions yield two distinct views $\tilde{\mathbf{X}}_i^{(1)} = T^{(1)}(\mathbf{X}_i)$ and $\tilde{\mathbf{X}}_i^{(2)} = T^{(2)}(\mathbf{X}_i)$. These views are passed through an encoder network $f_{\theta_f} : \mathcal{X} \to \mathbb{R}^d$, which maps each augmented



(A) Illustration of the set $\mathcal{T}$ of data augmentations used in SimCLR [54]. Each view is generated by composing all transformations $T_j \in \mathcal{T}$, where each T_j applied using independently sampled random parameters.



(B) Maximizing the similarity (green arrows) between feature representations of the views generated from the same anchor.

(C) Minimizing the similarity (red arrows) between the feature representations of the views generated from a different anchor.

Figure 2.3: Illustration of the contrastive pretext task of SimCLR [54], note that the figures are taken from [54].

input to a d-dimensional feature vector:

$$\mathbf{z}_i^{(1)} = f_{\theta_f}(\tilde{\mathbf{X}}_i^{(1)}), \qquad \mathbf{z}_i^{(2)} = f_{\theta_f}(\tilde{\mathbf{X}}_i^{(2)}). \tag{2.1}$$

The objective is to minimize the distance between positive pairs, i.e., the views generated from the same image, while pushing apart all other (negative) pairs, as illustrated in Figure 2.3 (B)-(C). This is achieved by using contrastive losses such as the Normalized Temperature-scaled Cross Entropy (NT-Xent) loss [54].

Specifically, given a batch of B unlabeled samples, let $\{\mathbf{z}_i\}_{i=1}^{2B}$ denote the set of representations corresponding to the $2B$ augmented views (two views per sample $\mathbf{X}_i$). Each representation $\mathbf{z}_i$ has a unique positive counterpart $\mathbf{z}_j$, where the index j is defined as:

$$j = \begin{cases} i + B & \text{if } i \leq B \\ i - B & \text{if } i > B \end{cases} \tag{2.2}$$

The NT-Xent loss [54] is then computed as:

$$\mathcal{L}_{\text{NT-Xent}} = \frac{1}{2B} \sum_{i=1}^{2B} -\log \frac{\exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k \neq i]} \exp\left(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau\right)} \tag{2.3}$$

where $\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}$ denotes cosine similarity, and $\tau \in \mathbb{R}^+$ is a temperature parameter. This formulation encourages the encoder f_θ to learn representations that are both transformation-invariant and instance-discriminative, which is beneficial for downstream tasks such as classification or retrieval.

2.1.4 Downstream task: Binary Classification

After pretraining the encoder f_{θ_f} for a given pretext task (i.e., context-based or contrastive), the learned representations can be transferred to a wide variety of downstream tasks, such as binary classification, as commonly done in SSL deepfake detection methods [60, 61, 62].

Let $\mathcal{D} = \{(\mathbf{X}_i, y_i)\}_{i=1}^{N}$ denote a small labeled dataset composed of N samples (i.e., a

deepfake detection dataset), with binary labels $y_i \in \{0, 1\}$ for a given application that can be formulated as a binary classification task. A linear or shallow classifier $g_{\theta_g} : \mathbb{R}^d \to \{0, 1\}$ is trained on top of the frozen or fine-tuned encoder f_{θ_f}, forming the composed model $\zeta = g_{\theta_g} \circ f_{\theta_f}$. The objective is to minimize the Binary Cross Entropy (BCE) loss:

$$\mathcal{L}(\theta_g) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{BCE}}(g_{\theta_g}(f_{\theta_f}(\mathbf{X}_i)), y_i), \tag{2.4}$$

where $\mathcal{L}_{\text{BCE}}$ denotes the standard binary cross-entropy. Empirically, despite the limited size of $\mathcal{D}$, the fine-tuned classifier can achieve satisfactory performance, often comparable to fully supervised models trained on larger labeled datasets, highlighting the effectiveness of SSL pretraining in low-label regimes [46].

2.2 Unsupervised Anomaly Detection

Anomaly Detection (AD) is a fundamental machine learning task whose primary goal is to detect instances that significantly deviate from some notion of normality [63]. Although AD can be framed as a binary classification problem when sufficient labeled anomalies are available, such annotations are often unavailable or costly to obtain in practice. This limitation is particularly relevant in domains such as industrial manufacturing [64] or spacecraft telemetry [65]. For instance, deliberately damaging expensive components in a production line or injecting failures into spacecraft systems is unreasonable, as it would compromise valuable resources, such as hardware or the spacecraft itself.

In such settings, anomaly detection is typically formulated as an unsupervised learning task, where models are trained on unlabeled data composed primarily of normal samples. Depending on how normality is modeled (i.e., what assumptions are made about anomalies), Unsupervised Anomaly Detection (UAD) techniques can be grouped into three major categories: one-class classification, reconstruction-based, and density-based techniques [63], with each category appearing at least once in our contributions:

- One-Class Classification (OCC) is a discriminative modeling paradigm grounded in

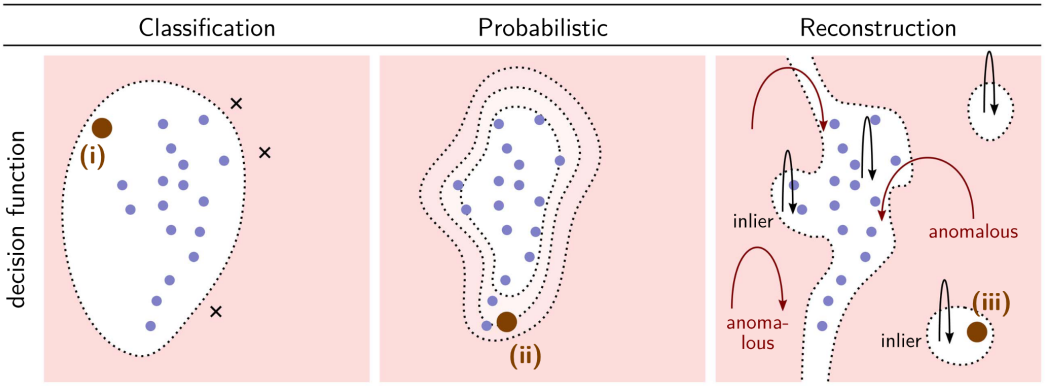


Figure 2.4: Various anomaly detection (AD) methods produce distinct types of decision functions, and their potential limitations. The decision regions are indicated by white and red for normal and anomalous areas, respectively. One-class classification techniques generally learn a discriminative boundary that separates normal from anomalous instances, but could be too loose leaving anomalies undetected (i). In contrast, probabilistic methods estimate the data distribution with the potential issue of possibly underfitting (or overfitting) the tails of a distribution (ii). Finally, reconstruction-based models aim to capture the intrinsic geometric structure of the data, such as a manifold or representative prototypes with the risk that Manifold or prototype structure artifacts leading to a good reconstruction of anomalies (iii). Figure from [63].

the concentration assumption [66, 67], and aiming to learn a decision boundary that compactly encloses the normal data distribution while assigning high anomaly scores to instances outside it. This formulation has motivated several well-established works, including One-Class Support Vector Machines (OC-SVM) [68], Support Vector Data Description (SVDD) [69], and recent deep learning-based methods such as Deep SVDD [70].

- Reconstruction-based approaches are generative techniques that learn to reconstruct normal inputs with a low reconstruction error, under the assumption that anomalies cannot be accurately reconstructed. Representative methods include Principal Component Analysis (PCA) [71], Autoencoders (AEs) [72], and GANs [73].

- Finally, probabilistic strategies assume that normal data occupies high-density regions of an underlying distribution, while anomalies reside in low-density regions. This perspective motivated works such as Gaussian Mixture Model (GMM) [74] and Kernel

Density Estimation (KDE) [75], which estimate the data density and identify anomalies as samples falling below a likelihood threshold.

Formally, let $\mathcal{X}$ denote the input space, and let $\mathcal{D}^n = \{(\mathbf{X}_i, y_i); y_i = 0\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$ be a training set composed of primarily normal samples, where $y_i = 0$ denotes the normal class. Let $f_{\theta_f} : \mathcal{X} \to \mathbb{R}^d$ be a feature extractor, and let $s_{\theta_s} : \mathbb{R}^d \to \mathbb{R}$ be a scoring function. The composed model can be defined as $\zeta_\theta = s_{\theta_s} \circ f_{\theta_f}$ with parameters $\theta = (\theta_f, \theta_s)$. The UAD training objective is then expressed as:

$$\min_{\theta} \ \mathbb{E}_{(\mathbf{X}_i, y_i) \sim \mathcal{D}^n} \left[\mathcal{L} \left(\zeta_\theta(\mathbf{X}_i), y_i = 0 \right) \right] + \lambda_{\text{reg}} \cdot R(f_{\theta_f}, s_{\theta_s}, \theta), \tag{2.5}$$

where $\mathcal{L}$ is a loss function applied to the anomaly score, R is a regularization term, and $\lambda_{\text{reg}} \in \mathbb{R}^+$ balances the influence of the regularization. The choice of R varies depending on the UAD subcategory and may be omitted in some formulations [63]. For instance, Deep SVDD [70], a one-class-based method, employs L2-regularization over network weights to avoid feature collapse [70].

At inference, the model is expected to generalize to the test distribution $\mathcal{D}^{\text{test}}$, which includes both normal and anomalous samples. For each $\mathbf{X}_i \in \mathcal{D}^{\text{test}}$, the model computes an anomaly score $\zeta(\mathbf{X}_i)$. A binary prediction is then obtained by thresholding the score:

$$\hat{y}_i = \begin{cases} 1, & \text{if } \zeta(\mathbf{X}_i) > \delta \quad \text{(anomaly)} \\ 0, & \text{otherwise} \quad \text{(normal)}, \end{cases} \tag{2.6}$$

where $\delta \in \mathbb{R}$ is a threshold based on calibration or evaluation criteria.

This unified UAD formulation accommodates various data modalities, such as images, where the input space is $\mathcal{X} = \mathbb{R}^{h \times w \times c}$, with h, w, and c denoting the image height, width, and number of channels, respectively; as well as time-series, where the input $\mathbf{X}_i$ corresponds to a temporally ordered sequence $\mathbf{X}_i = \{\mathbf{X}_t\}_{1 \le t \le T}$, with $\mathbf{X}_t \in \mathbb{R}^{d'}$ representing a d'-dimensional observation at timestamp t, and T being the length of the sequence.

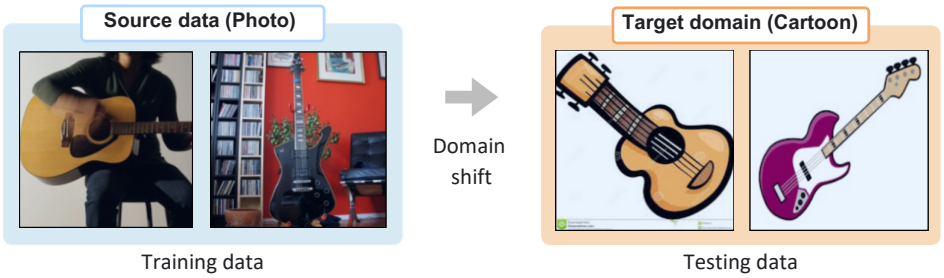


Figure 2.5: Illustration of domain shift in unsupervised domain adaptation: the model is trained on labeled source data from the photo domain and evaluated on unlabeled target data from the cartoon domain, highlighting the challenge of generalizing across visually distinct domains despite the shared label space between both domains.

2.3 Unsupervised Domain Adaptation for classification

Most unsupervised anomaly detection (UAD) methods implicitly assume that the distribution of normal data encountered during training remains unchanged at test time and that the model is expected to generalize to unseen domains without explicit adaptation. However, this assumption is often challenged in practice due to the presence of domain shift [34], which results in a significant degradation of performance under cross-domain settings (see Figure 2.5).

Unsupervised Domain Adaptation (UDA) is a well-established paradigm within Transfer Learning (TL) that mitigates domain shift by transferring knowledge from a labeled source domain to a related but different unlabeled target domain [76]. Specifically, UDA aims to learn domain-invariant feature representations by jointly training a model on labeled source data and unlabeled target data, thus eliminating the need for manual annotation on the target domain. As UDA has not been studied in the context of UAD, the following sections provide a review of representative UDA approaches from general image classification, covering the multi-class settings.

Formally, Let $\mathcal{D}^s = \{(\mathbf{X}_i^s, y_i^s)\}_{i=1}^{N_s}$ be a labeled dataset from a source domain, consisting of N_s samples, where each input $\mathbf{X}_i^s \in \mathbb{R}^{h \times w \times c}$ is an image and its associated label $y_i^s \in \mathcal{Y} = \{1, \ldots, C\}$, for $i = 1, \ldots, N_s$. Let $\mathcal{D}^t = \{\mathbf{X}_i^t\}_{i=1}^{N_t}$ be an unlabeled dataset from a related but distinct target domain, composed of N_t samples, where each $\mathbf{X}_i^t \in \mathbb{R}^{h \times w \times c}$, for $i = 1, \ldots, N_t$.

We assume that the source and target domains share the same label space $\mathcal{Y}$, and that there exists a domain shift between them. The goal of Unsupervised Domain Adaptation (UDA) for image classification is to learn a model $\zeta : \mathbb{R}^{h \times w \times c} \to \mathcal{Y}$ using both $\mathcal{D}^s$ and $\mathcal{D}^t$, such that it generalizes effectively to the target domain. Specifically, the model consists of a domain-invariant feature extractor $f : \mathbb{R}^{h \times w \times c} \to \mathbb{R}^d$, followed by a classifier $g : \mathbb{R}^d \to \mathcal{Y}$, such that $\zeta = g \circ f$.

Given this general setting, the two subsequent sections introduce two major UDA families: moment-matching and adversarial UDA techniques.

2.4 Moment-matching Domain Adaptation

Moment matching approaches constitute a class of UDA techniques whose goal is to learn domain-invariant feature representations by explicitly aligning the distributions of the source and target domains. This is typically achieved by minimizing statistical distances such as the Maximum Mean Discrepancy (MMD) [77], which measures the discrepancy between the empirical means of feature representations in a reproducing kernel Hilbert space (RKHS). Specifically, samples from both domains are mapped into an RKHS via a kernel-induced feature map φ, and the squared distance between their mean embeddings is minimized. A large distance indicates misalignment between the two distributions. Formally, given feature embeddings $f(\mathbf{X}_i^s)$ and $f(\mathbf{X}_j^t)$ extracted by the shared encoder f, the squared MMD is defined as:

$$d^2_{\text{ker}}(\mathcal{D}^s, \mathcal{D}^t) = \left\| \mathbb{E}_{\mathbf{X}^s \sim \mathcal{D}^s}[\varphi(f(\mathbf{X}^s))] - \mathbb{E}_{\mathbf{X}^t \sim \mathcal{D}^t}[\varphi(f(\mathbf{X}^t))] \right\|^2_{\mathcal{H}_{\text{ker}}}, \tag{2.7}$$

where $\mathcal{H}_{\text{ker}}$ is the RKHS associated with the kernel ker. The multiple-kernel MMD (MK-MMD) [78] extends this formulation by combining several kernels to capture features across different layers and scales, as illustrated in Figure 2.6, thereby improving the robustness of distribution alignment. A notable extension is the Joint Maximum Mean Discrepancy (JMMD) [79], which aligns joint distributions across multiple domain-specific layers, encouraging consistency in both marginal and conditional distributions. Another related method

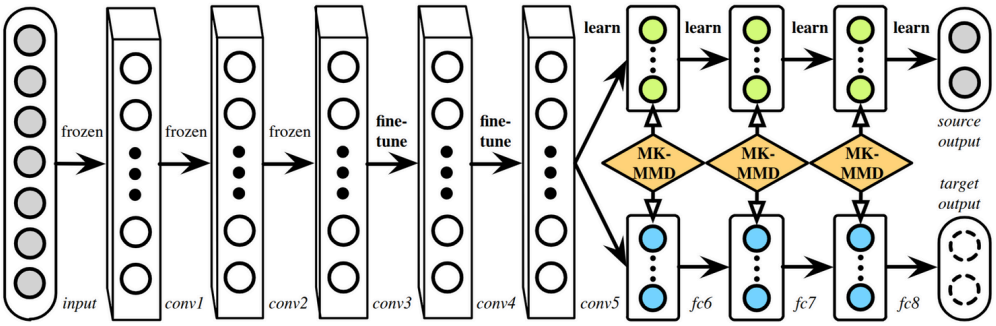


Figure 2.6: Illustration of MK-MMD domain alignment using multiple kernels.

is Correlation Alignment (CORAL) [80], which minimizes the discrepancy between second-order statistics (i.e., covariances) of source and target features, and can be interpreted as analogous to MMD with a second-order polynomial kernel [80].

2.4.1 Adversarial Domain Adaptation

Adversarial UDA constitutes a popular class of domain adaptation methods that aim to learn domain-invariant feature representations by implicitly aligning the source and target distributions. This is typically achieved by introducing an auxiliary trainable module, known as the *domain discriminator*, which distinguishes whether input features originate from the source or target domain.

Formally, a domain discriminator $D : \mathbb{R}^d \to [0, 1]$ is introduced, where $\mathbb{R}^d$ is the feature space. The discriminator is trained to output 1 for source features and 0 for target features. In contrast, the feature extractor f is optimized to produce representations that prevent the discriminator from reliably distinguishing between the two domains. This adversarial setup encourages the learned feature space to be domain-invariant.

The learning process is formulated as a minimax optimization between two objectives: (i) a supervised classification loss on the labeled source data, and (ii) an adversarial loss that aligns the source and target distributions. The classification loss is defined as

$$\mathcal{L}^s(f, g) = \frac{1}{N_s} \sum_{i=1}^{N_s} \ell(g(f(\mathbf{X}_i^s)), y_i^s), \tag{2.8}$$

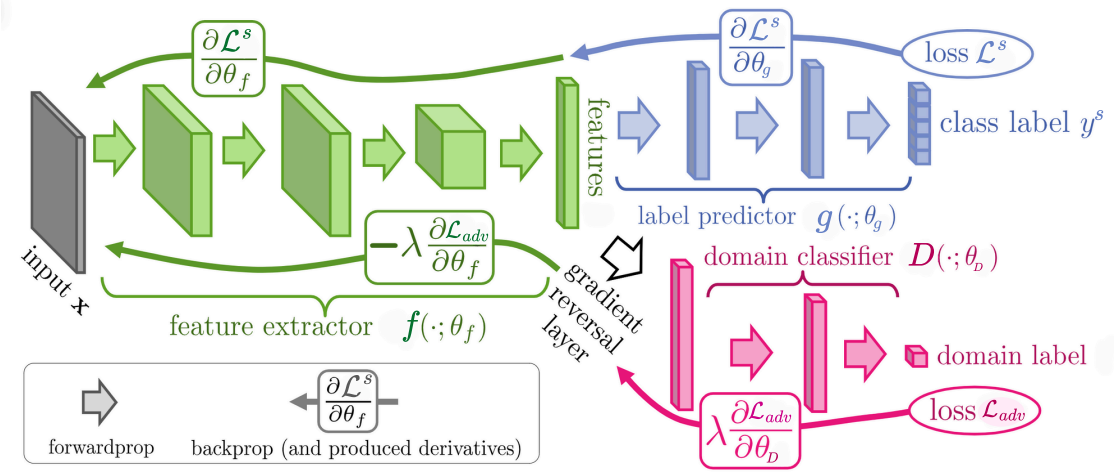


Figure 2.7: Illustration of the DANN architecture [81].

where ℓ denotes the cross-entropy loss. This term promotes discriminative learning on the source domain. The adversarial loss is defined as

$$\mathcal{L}_{\text{adv}}(f, D) = -\frac{1}{N_s}\sum_{i=1}^{N_s} \log D(f(\mathbf{X}_i^s)) - \frac{1}{N_t}\sum_{i=1}^{N_t} \log(1 - D(f(\mathbf{X}_i^t))), \tag{2.9}$$

where D learns to distinguish source from target representations, while f is optimized to make its outputs indistinguishable across domains.

The full objective is given by

$$\min_{\zeta} \max_{D} \; \mathcal{L}^s(\zeta) + \lambda_{\text{adv}}\, \mathcal{L}_{\text{adv}}(f, D), \tag{2.10}$$

where $\lambda_{\text{adv}} > 0$ balances the classification and alignment losses.

This formulation follows the DANN framework [81], as illustrated in Figure 2.7, and enables the model to learn representations that are both class-discriminative on the source domain $\mathcal{D}^s$ and domain-invariant across $\mathcal{D}^s$ and $\mathcal{D}^t$.

Chapter 3

UNTAG: Learning Generic Features for Unsupervised Type-Agnostic Deepfake Detection

This chapter introduces a novel framework for **UN**supervised **T**ype-**AG**nostic deepfake detection referred to as UNTAG. Existing methods are generally trained in a supervised manner at the classification level and usually focus on detecting two types of forgeries at most; therefore, limiting their generalization capability across different types of deepfakes. To handle that, we reformulate the problem of deepfake detection as a one-class classification coupled with a self-supervision feature learning mechanism. Our intuition is that by estimating the distribution of real data in a discriminative feature space, a deepfake can be detected as an outlier regardless of its type. In particular, UNTAG involves two sequential steps. First, deep representations are learned based on a self-supervised mechanism that focuses on manipulated regions. Second, a one-class classifier that detects deepfakes from the learned deep representations is estimated. The results reported on several datasets show the effectiveness of UNTAG. They also highlight the relevance of the proposed new paradigm for unsupervised type-agnostic deepfake detection. The source code is publicly available.

3.1 Introduction

Deepfakes are realistic facial images or videos that are either fully generated or partially altered using a generative Deep Neural Network (DNN) such as Generative Adversarial Networks (GANs) [73, 82]. Over the last years, numerous incidents, including fraud and misinformation [13, 12], have raised concerns about their misuse [83].

Given this threat, several deepfake detection methods have been introduced [84, 23, 85, 86, 87, 88, 89, 90]. Nevertheless, existing approaches remain hardly applicable to real-world scenarios given their lack of *inter-type* generalization. In fact, generalization can be addressed at two levels: (1) At the inter-type level, we mean robustness to unseen types of deepfakes. Possible types of deepfakes are Face Swap (FS), Facial Reenactments (FR), Facial Attribute Manipulations (FAM), and Fully Synthetic Faces (FSF) such as GAN-generated faces; (2) At the *intra-type* level, we mean robustness to unseen forgery methods generating the same type of deepfakes. While intra-type generalization has been extensively studied, the topic of inter-type generalization remains less explored. Figure 3.1 further clarifies the distinction between intra-type and inter-type generalizations. This figure depicts the different types of deepfakes along with dataset examples incorporating them. It also specifies the focus of our chapter compared to most state-of-the-art methods.

Earlier approaches formulate the problem of deepfake detection as an end-to-end binary supervised classification task [23]. More specifically, they mostly learn a DNN model that focuses on image or video cues, known as artifacts. Unfortunately, such methods have shown poor *intra-type generalization* capabilities. This drop in performance might be explained by the fact that fully supervised DNNs tend to overfit the training data, as highlighted in [91, 92, 93, 94]. To overcome these limitations, some approaches have employed a self-supervision strategy for extracting more generic features [28, 60, 95]. Nevertheless, these approaches rely on a supervised classification for detecting deepfakes. Thus, they highly depend on annotated data; hence achieving low inter-type generalization. In other words, they tend to be effective solely in the presence of forgery types encountered at the training phase.

This chapter addresses the under-explored research problem of *type-agnostic deepfake*

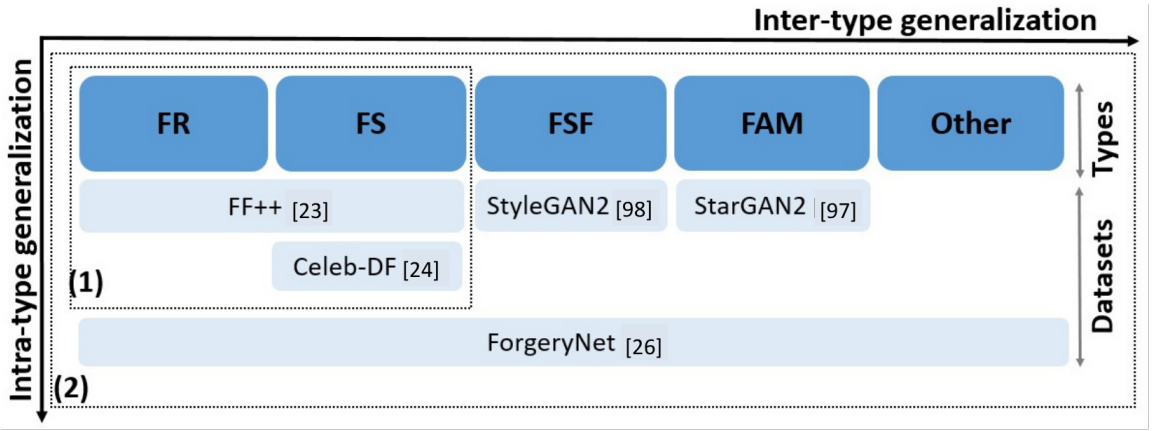
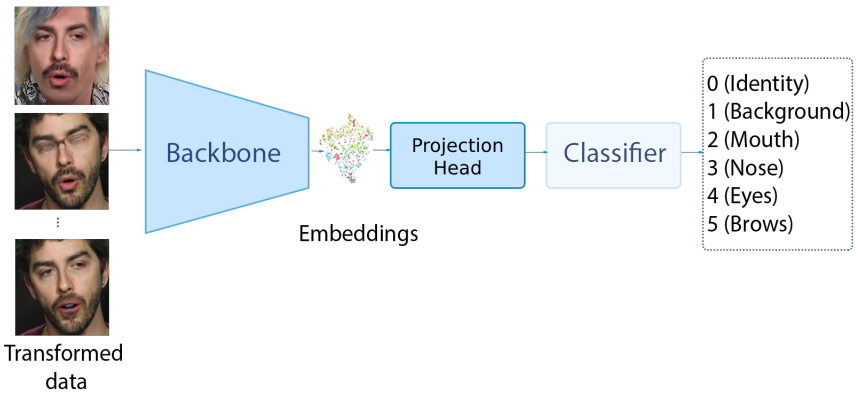


Figure 3.1: (1) The focus of most state-of-the-art deepfake detectors in terms of intra-type and inter-type generalization versus (2) the focus of the proposed method. (a) and (b) refer to the forgery types and datasets, respectively. FR, FS, FSF, and FAM refer to Facial Reenactment, Face Swap, Fully Synthetic Faces, and Facial Attribute Manipulation, respectively. Other is for unknown and stacked forgery types.

detection using unlabeled data. As a solution, we propose to model the distribution of normal images/videos and detect deepfakes as anomalies. Such an approach also prevents the use of costly annotated data. To the best of our knowledge, unsupervised classification for deepfake detection has only been considered in [96] where a Variational Auto-Encoder (VAE) was used to learn the distribution of real data. However, while this approach can be conceptually employed for detecting any types of deepfakes, the authors do not explicitly consider more than two usual types, namely Face Swaps and Facial Reenactments. In Section 3.5, we show experimentally that [96] has poor generalization across different types of deepfakes. Two facts might explain this. First, the generated features are not discriminative enough as the learning process is not implicitly guided to focus on specific artifact-sensitive regions. Second, the Variational Auto-Encoder (VAE) assumes that the latent representations of real data follow a Gaussian distribution which might be too simplistic for modeling the complex distribution of real data. In this chapter, we propose a novel Unsupervised Type-Agnostic deepfake detection (UNTAG) which leverages an appropriate self-supervision mechanism for learning generic yet discriminative features. First, a simple augmentation technique called R-Splicer is introduced. It generates synthetic data by apply-

(a) **Learning self-supervised representations**



(b) **Unsupervised one-class classification supported by self-supervised features.**

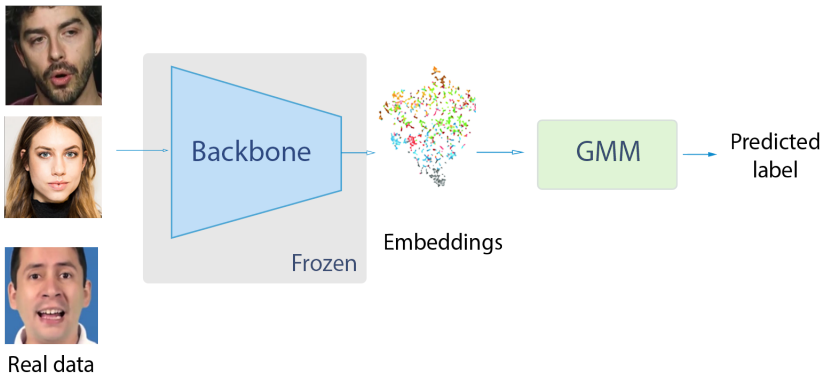


Figure 3.2: The proposed UNTAG framework is based on: (a) First, a self-supervised feature learning is considered: a pretext task learns implicitly artifact-sensitive region features by predicting the manipulated regions if any in an image; and (b) Second, an unsupervised generative one-class classifier is estimated using the self-supervised features of real images.

ing splicing and blending operations on multiple predefined regions in a given image. The selected regions are known to potentially incorporate artifacts for different types of deepfakes. Then, the augmented data are used to train a deep learning network that detects the manipulated regions. Our intuition is that by employing this self-supervision mechanism, the network will implicitly produce features that can target *artifact-sensitive* regions. Second, the feature learning step is followed by an unsupervised one-class generative classifier which estimates the probability density of real data; thus, considering only authentic data during the training phase.

The contributions of this chapter are summarized below: (1) The paradigm of unsupervised type-agnostic deepfake detection is introduced. To the best of our knowledge, no prior work has explicitly formulated it as such; (2) A novel framework called UNTAG for unsupervised type-agnostic deepfake detection is proposed. In this context, an original non-contrastive self-supervised pretext task is specifically tailored for the problem of deepfake detection; (3) A simple augmentation technique termed R-Splicer is proposed and used for training the pretext task; (4) A protocol for evaluating the performance of deepfake detectors under the proposed formulation using three well-known datasets and two generation methods, namely, ForgeryNet [26], FaceForensics++ [23], Celeb-DF [24], StarGAN2 [97], StyleGAN [98, 99] is designed; and (5) An extensive experimental evaluation is carried out. The chapter is structured as follows: Section 3.2 describes the state-of-the-art of self-supervised and unsupervised deepfake detection. Section 3.3 formulates the new paradigm of type-agnostic deepfake detection using a one-class classifier. The proposed method called UNTAG is detailed in Section 3.4. The experiments' results and limitations are given in Section 3.5. Finally, Section 3.6 concludes this work.

3.2 Related works

Earlier deepfake detection methods mostly train a Convolutional Neural Network (CNN) in an end-to-end manner to detect generic [86, 21, 22] or specific artifacts [100, 101, 94]. Despite their performance in constrained settings, it has been shown in multiple references

that they struggle to generalize to unseen deepfake generation methods [91, 92, 93, 94]. To overcome this issue, a growing focus is being given to self-supervision [102, 20, 103, 27, 62, 104, 61, 60].

Leveraging self-supervision for deepfake detection. Instead of training a network in an end-to-end manner, self-supervised methods guide the learning of feature representations by considering a contrastive loss or an auxiliary task known as the pretext task. Auxiliary task-based approaches typically involve the classification of augmented data which simulates a targeted inconsistency [102, 20, 103, 27, 61, 60, 28, 95]. Contrastive learning approaches [104, 62] aim at minimizing/maximizing the similarity/dissimilarity between an instance and its augmentations. Although these techniques leverage self-supervision, the final classification remains supervised unlike UNTAG which uses a one-class classifier. As a consequence, these techniques do not generalize effectively to unseen types of deepfakes and require the availability of large annotated datasets.

Unsupervised deepfake detection. While existing deepfake detection methods rely on a binary classification task, Khalid et al. [96] introduced a one-class classifier architecture for detecting deepfakes as anomalies. This means that only real data are considered during the training phase. While this idea seems promising for achieving inter-type generalization, the authors attempted to detect only two kinds of deepfakes, namely, face-swaps and facial reenactments. As discussed in the introduction, this method lacks inter-type generalization capabilities probably due to: (1) the low discriminative power of learned features; and (2) the constraining assumption that real embeddings follows a Gaussian distribution. This claim is experimentally supported in Section 3.5.

3.3 A new paradigm for type-agnostic deepfake detection

Let $\mathcal{D} = (\mathcal{I}, \mathcal{L})$ be a dataset composed of N images $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^{N}$ and their corresponding labels $\mathcal{L} = \{l_i\}_{i=1}^{N}$. $\mathbf{I}_i \in \mathbb{R}^{c \times w \times h}$, $l_i \in [\![0, 1]\!]$ and c, w and h are the number of channels, the width and the height of the image $\mathbf{I}_i$, respectively. $\mathcal{I}$ is defined by $\mathcal{I} = \mathcal{I}^R \cup \mathcal{I}^F$ where $\mathcal{I}^R$ and $\mathcal{I}^F$ are the subsets of real and fake images, respectively. For all $i \in [\![1, N]\!]$ and $\mathbf{I}_i \in \mathcal{I}$,

the label $l_i = \mathbb{1}_{\mathbf{I}_i \in \mathcal{I}^F}$, with $\mathbb{1}$ being an indicator function. $\mathcal{I}^F$ is assumed to contain all types of deepfakes, i.e., $\mathcal{I}^F = \bigcup_{j=1}^{n} \mathcal{I}^{F_j}$, with $\mathcal{I}^{F_j}$ being the set of images belonging to a particular type of deepfakes, and n the total number of forgery types. The ultimate goal of deepfake detection is to learn a function $f(.)$ such that,

$$\forall\, i \in [\![1, N]\!] \text{ and } \mathbf{I}_i \in \mathcal{I}, f(\mathbf{I}_i) = l_i. \tag{3.1}$$

Fully supervised approaches. These methods aim at learning a feature extractor function $f_\theta(.)$ parametrized by the weights θ of a neural network such that,

$$\forall\, i \in [\![1, N]\!] \text{ and } \mathbf{I}_i \in \tilde{\mathcal{I}},\ \sigma(f_\theta(\mathbf{I}_i)) = l_i, \tag{3.2}$$

where $\sigma(.)$ is an activation function used for classifying the input image as real or fake. Note that only a subset of $\mathcal{D}$ denoted by $\tilde{\mathcal{D}} = (\tilde{\mathcal{I}}, \tilde{\mathcal{L}}) \subset \mathcal{D}$ is used to train the model. In fact, supervised methods focus mostly on one to two types of deepfakes, e.g. face-swaps and facial reenactment. In other words, $\tilde{\mathcal{I}} = \mathcal{I}^R \cup \mathcal{I}^{F'}$, with $\mathcal{I}^{F'} = \bigcup_{j=1}^{n_1} \mathcal{I}^{F_{s(j)}} \subset \mathcal{I}^F$ being the subset containing the fake images of the considered forgery types. Each type is denoted by $F_{s(j)}$, for $s(j) \in [\![1, n]\!]$ a sampling function indicating the deepfake type and $n_1 < n$ the number of deepfake types in $\mathcal{I}^{F'}$. Focusing on a subset of fakes makes the inter-type generalization difficult. In addition to that, since f_θ is learned in an end-to-end manner, supervised techniques usually do not generalize to unseen generation algorithms producing similar types of deepfakes F' [20, 105].

Self-supervision based on an auxiliary task for deepfake detection. For an enhanced intra-type generalization, self-supervised techniques, usually decouple the learning process into two stages. A first stage extracts rich representations by considering an auxiliary task that is generally unrelated to the detection task. The second stage discriminates between real and fake images based on the extracted representations. Formally, the aim of self-supervised approaches is to learn two functions f_{θ_1} and f_{θ_2} parametrized by the neural

network weights θ_1 and θ_2, respectively, such that,

$$\forall\ i \in [\![1, N]\!] \text{ and } \mathbf{I}_i \in \tilde{\mathcal{I}},\ f_{\theta_2}(f_{\theta_1}(\mathbf{I}_i)) = l_i. \tag{3.3}$$

Typically, the estimation of f_{θ_1} involves only the set of real images $\mathcal{I}^R$. The latter is extended to a set of transformed images $\mathcal{I}^{Aug}$ associated with pseudo-labels $\mathcal{L}^{Aug}$, forming $\mathcal{D}^{Aug} = (\mathcal{I}^{Aug}, \mathcal{L}^{Aug})$. The dataset $\mathcal{D}^{Aug}$ is used to perform the auxiliary task. Hence, f_{θ_2} maps latent embeddings resulting from the auxiliary task to their corresponding labels. For the second phase, the subset $\tilde{\mathcal{D}}$ is used in a supervised fashion. Although self-supervised mechanisms improve *intra-type generalization* aspect, these approaches still rely on annotated data and are therefore not generic across different types of deepfakes.

Leveraging self-supervision for one-class type-agnostic deepfake detection. In this chapter, we propose to address the problem of unsupervised type-agnostic deepfake detection. However, unlike [96], our goal is to learn discriminative type-agnostic features, while modelling more accurately the distribution of real data. For that purpose, we propose to decouple the feature learning from the final classification as in Eq. (3.3). First, a self-supervised strategy tailored to the task of type-agnostic deepfake detection is leveraged for estimating f_{θ_1}. However, instead of learning a binary classifier during the second stage, the embeddings $f_{\theta_1}(\mathbf{I})$ generated from real samples $\mathbf{I} \in \mathcal{I}^R$ are assumed to follow a multivariate, Gaussian mixture distribution, such that $f_{\theta_1}(\mathbf{I}) \propto p(f_{\theta_1}(\mathbf{I})|l = 0)$ and l is the label of $\mathbf{I}$. The probability density $p(f_{\theta_1}(\mathbf{I})|l = 0)$ is defined as,

$$p(f_{\theta_1}(\mathbf{I})|l = 0) = \sum_{i=1}^{K} \Phi_i\, \mathcal{N}(f_{\theta_1}(\mathbf{I})|(^{-}_i, \mathbf{\Sigma}_i), l = 0). \tag{3.4}$$

Note that $\sum_{i=1}^{K} \Phi_i = 1$, K is the number of Gaussian components and Φ_i is the weight of the component i. This assumption is in line with the concentration hypothesis [106], which suggests that the embeddings of real and fake data are respectively assumed to be concentrated and non-concentrated in the feature space. At this stage, unlike previous self-supervised methods, no annotated data is used. Hence, the problem can be seen as a one-

class classification since only real images are taken into account for training. As real data is unlikely to be noise-free, we refer to this formulation as an unsupervised task. The function f_{θ_2} allows the discrimination between real latent features and any non-real embeddings and is computed as follows,

$$f_{\theta_2}(f_{\theta_1}(\mathbf{I}))) = 1 - \mathbb{1}_{[-L(\theta_2|f_{\theta_1}(\mathbf{I}))>\tau)]}, \tag{3.5}$$

where $L(\theta_2|f_{\theta_1}(\mathbf{I})) = -\text{Log}(p(f_{\theta_1}(\mathbf{I})|l = 0)$ is the log-likelihood given the parameter $\theta_2 = (^{-}{}_i, \mathbf{\Sigma}_i)_{i\in[\![1,K]\!]}$ and $\tau > 0$ is a predefined threshold.

3.4 UNTAG: Unsupervised type-agnostic deepfake detection

This section describes the proposed approach called UNsupervised Type-AGnostic (UNTAG) deepfake detection. As formulated in Eq. (3.4) and Eq. (3.5), a framework involving an auxiliary and a downstream task is proposed for detecting deepfakes. In particular, a pretext task for learning type-agnostic artifact-sensitive features is proposed in order to estimate f_{θ_1} (3.4.1). Hereafter, a final unsupervised classification process allows estimating f_{θ_2} based on a one-class Gaussian Mixture Model (GMM) (3.4.2). An overview of UNTAG is given in Figure 3.2.

3.4.1 Self-supervised learning of generic and discriminative type-agnostic features

Inspired by [48, 107, 108], our objective is to learn discriminative and generic representations from a set of transformed images. For that purpose, a projection head [54] is adopted as it has shown great performance [54, 109, 108]. A projection head is a multilayer perceptron (MLP) appended to the backbone network just after its pooling layer. It is used during training and discarded at inference time.

More concretely, given a dataset of transformed images and their generated pseudo-labels $\mathcal{D}^{Aug}$, the pretext task learns a composition of two functions f_{θ_1} and f_{θ_3}. The function

f_{θ_3}, parametrized by θ_3 corresponds to the projection head. The two mappings are learned in an end-to-end manner such that, given an input image $\mathbf{I}_m \in \mathcal{I}^{Aug}$ and its associated pseudo-label $l_m \in [\![0, k]\!]$,

$$f_{\theta_3} \circ f_{\theta_1}(\mathbf{I}_m) = l_m. \tag{3.6}$$

$f_{\theta_1}(\mathbf{I}_m)$ denotes the features extracted by the backbone network and $f_{\theta_3}(f_{\theta_1}(\mathbf{I}_m))$ refers to the predicted pseudo-label. It is done by minimizing the following loss denoted by R_p,

$$R_p = \mathop{\mathbb{E}}_{\mathbf{I}_m \sim \pi_{\mathcal{X}}} \left[\mathbb{H}(l_m, L_{f_{\theta_3} \circ f_{\theta_1}}(l|\mathbf{I}_m) \right], \tag{3.7}$$

where $\approx_{\mathcal{X}}$ is the distribution of the augmented training data, $\mathbb{H}$ is the cross-entropy loss,

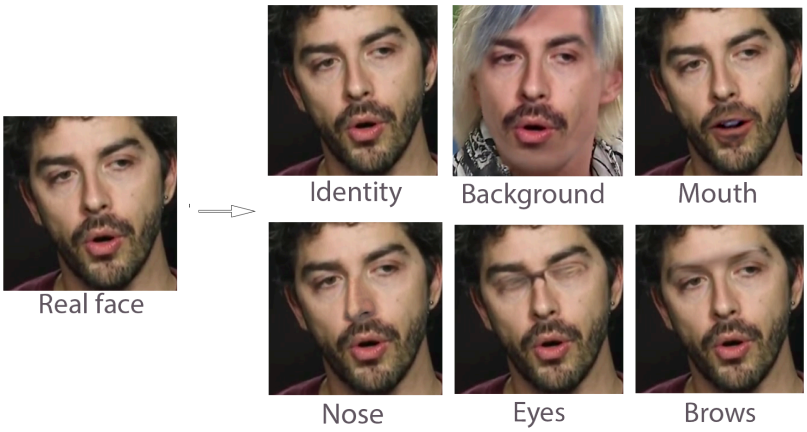


Figure 3.3: The transformations generated by R-splicer given a real image from ForgeryNet [26]

and $L_{f_{\theta_3} \circ f_{\theta_1}}(l_m|\mathbf{I}_m) = L_{f_{\theta_3}}(l_m|f_{\theta_1}(\mathbf{I}_m))$ is the likelihood of being in the presence of label l_m given the extracted image embeddings $f(\mathbf{I}_m)$. Overall, the key idea consists in applying suitable transformations to images for learning discriminative representations. In line with this, a tailored data augmentation called R-splicer is proposed in order to generate $\mathcal{D}^{Aug}$.

R-Splicer. Augmenting real data by generating *pseudo-fake* images is a common practice in the deepfake detection literature [20, 103, 27, 19, 28]. Such methods simulate characteristic face-swap artifacts using simplistic operations [20, 103, 27, 19]. These augmentation strategies coupled with self-supervision have significantly boosted the intra-type generaliza-

tion capabilities of deepfake detectors. In particular, they mostly focus on creating synthetic blending or warping artifacts located in the boundaries of the facial area. As a result, these approaches struggle to achieve inter-type generalization as experimentally demonstrated in [19] on GAN-generated images. In contrast, our method tries to be more generic by mimicking different deepfake types. It simulates artifacts not only in the boundaries of the facial area, but also in the background and in more localized facial regions. Hence, it achieves good inter-type generalization, as shown in Section 3.5.

Our approach is based on two key observations about deepfakes: (1) Most methods [103, 20, 27, 19, 28] assume that artifacts lie only in the facial area. However, in practice, this does not always hold. For example, inconsistencies may occur in the person's hair or background in GAN-generated faces, and (2) deepfakes may exhibit fine-grained local and global inconsistencies, or both simultaneously. For example, fully synthetic faces are global forgeries whereas facial reenactments are fine-grained. These observations suggest that the model should target different regions of the image and take into consideration local and global artifacts. In line with these observations, the introduced R-Splicer applies splicing operations on a predefined set of facial and non-facial regions. In total, k ($k = 5$) regions are manipulated as depicted in Figure 3.3. In this chapter, the choice of regions is heuristically made by taking into account three elements: (1) areas in which artifacts are more likely to appear in different types of deepfakes; (2) areas with high-level semantics; and (3) simplicity of the splicing operation. For example, other regions such as the ear have been excluded, as it is often occluded and difficult to detect. More regions might be included in future works. This is in line with the recently introduced methods [28], which aim at manipulating different regions. Nevertheless, this work differs from ours as the manipulations are partly supervised with annotated images by considering the cross-entropy loss.
Formally, a spliced image $\mathbf{I}_m$ is defined as,

$$\mathbf{I}_m^{(j)} = \mathbf{M}_i \odot \mathbf{I}_d^{(j)} + (\mathbf{J}_{w \times h} - \mathbf{M}_i) \odot \mathbf{I}_r^{(j)}, \text{ for } j \in [\![1, c]\!], \tag{3.8}$$

where j is the channel index, $\mathbf{M}_i$ is a grayscale mask associated with the predetermined

i^{th} region, $\mathbf{I}_r$ is the target image to be manipulated and $\mathbf{I}_d$ is the image transferring its region of interest, $\mathbf{J}_{w\times h}$ is the all-ones matrix of size $w \times h$ and $\odot$ is the element-wise multiplication. Therefore, using a set of n_r real images belonging to $\mathcal{I}^R$, the dataset $\mathcal{D}^{Aug} = (\mathcal{I}^{Aug}, \mathcal{L}^{Aug})$ is built by applying on each image all the predefined splicing operations denoted by $\mathcal{T} = \{\mathcal{T}_j\}_{j=0}^{k}$, where k is the number of candidate manipulation regions. This means that $\mathcal{I}^{Aug} = \bigcup_{i=0}^{n_r}(\bigcup_{j=0}^{k} \mathcal{T}_j(\mathbf{I}_i))$ with $\mathcal{T}_0$ being the identity transformation such that $\mathcal{T}_0(\mathbf{I}) = \mathbf{I}$ for $\mathbf{I} \in \mathcal{I}^R$ and $\mathcal{T}_j$ for $j \neq 0$ a function that splices the j^{th} region and replaces it with the same region from another image. The generated labels $\mathcal{L}^{Aug} = \bigcup_{i=0}^{n_r}(\bigcup_{j=0}^{k} j)$ correspond to the manipulated regions. A technical explanation of the details of R-Splicer is provided in Algorithm 1.

3.4.2 Unsupervised one-class classification with Gaussian Mixture Models

For detecting deepfakes, we finally propose the use of a generative one-class classifier denoted by f_{θ_2} in Eq. (3.5). The network trained for the pretext task is frozen for extracting features that are fed to the one-class classifier. More precisely, a GMM is first fitted using only real data embeddings, as presented in Eq. (3.4). The GMM parameters, denoted by θ_2 in Eq. (3.5), are estimated using the Expectation-Maximization (EM) method [110]. Then, at inference, the GMM discriminates between embeddings extracted from real images and non-authentic ones as shown in Eq. (3.5).

3.5 Experiments

3.5.1 Experimental Protocol

Baselines. We propose to compare UNTAG to six representative baselines: Two supervised deepfake detection approaches called (1) **DFD-HF** [111] and **Xception-Ext** is an Xception [21] that we re-adapt and combine with our pretext task feature extractor pretrained on R-spliced data. It is finetuned using the protocol of [23]. (3) a self-supervised deepfake detection method termed **DSP-FWA** [103]; (4) to the best of our knowledge, the only fully

Algorithm 1: R-Splicer

Input: A set $\mathcal{I}^R$ of N^R real images, where $N^R < N$, k = 5 regions
An index-to-region mapping $\{\{$**0** : No Transform$\}, \{$**1** : Background$\}, \{$**2** : Mouth$\}$, $\{$**3** : Nose$\}, \{$**4** : $\{$Eyes$\}, \{$**5** : Brows$\}\}$
A predefined set of facial landmark IDs for each region
Output: An augmented dataset $\mathcal{D}^{Aug} = (\mathcal{I}^{Aug}, \mathcal{L}^{Aug})$

$\mathcal{I}^{Aug} \leftarrow [\,]$ `// Initialize empty image list`
$\mathcal{L}^{Aug} \leftarrow [\,]$ `// Initialize empty label list`

for $i \leftarrow 1$ ***to*** N^R **do**
  `// Face processing`
  $\boldsymbol{I}_i \leftarrow$ Load i^{th} image from $\mathcal{I}^R$
  $\boldsymbol{I}_i \leftarrow$ cropped Region-Of-Interest `// the face in` $\boldsymbol{I}_i$

  for $region\_index \leftarrow 0$ ***to*** k **do**
    `// Lookup` $region\_index$ `in the index-to-region mapping`
    if $region\_index == 0$ **then**
      $\boldsymbol{I}_m \leftarrow \boldsymbol{I}_i$ `// No transformation applied`
    end
    else
      Retrieve facial landmark IDs for associated with $region\_index$
      Sample 20 face candidates $\{\boldsymbol{I}_c^1, ..., \boldsymbol{I}_c^{20}\}$ from $\mathcal{I}^R$
      Compute head poses for all 20 face candidates
      Select $\boldsymbol{I}_c$ with the closest head pose to $\boldsymbol{I}_i$

      Compute landmarks $\mathbf{lmk}_i$ for $\boldsymbol{I}_i$ `// original image`
      Compute landmarks $\mathbf{lmk}_c$ for $\mathbf{I}_c$ `// splicing candidate`

      Warp $\mathbf{lmk}_c$ to match $\mathbf{lmk}_i$
      Generate splicing mask $\boldsymbol{M}$
      Compute$(\boldsymbol{I}_m)_{j=3}$ `// Blend region onto` $\mathbf{I}_i$ `following Eq(8)`

    end
    $l_m \leftarrow region\_index$ `// Assign region_index label`
    $\mathcal{I}^{Aug} \leftarrow$ append$(\mathcal{I}^{Aug}, \boldsymbol{I}_m)$
    $\mathcal{L}^{Aug} \leftarrow$ append$(\mathcal{L}^{Aug}, l_m)$
  end
end

Method \ Dataset	Celeb-DF [24]		FF++ [23]		StyleGAN [98]		StarGAN [97]		ForgeryNet [26]		Mean AUC
	AUC	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**	
Xception-Ext [21]	50.29	50.57	48.58	50.00	55.35	44.50	46.99	50.00	49.84	50.16	50.21
DFD-HF [111]	43.12	50.70	51.21	50.75	50.66	52.35	76.99	50.75	43.10	50.32	52.96
DFD-HF-OC	25.40	50.00	31.06	50.00	59.87	37.36	51.50	43.33	37.66	49.95	43.57
DSP-FWA [103]	49.47	49.50	53.65	53.36	63.57	63.10	50.76	50.81	51.65	51.40	53.65
DSP-FWA-OC	52.61	52.60	72.00	71.79	50.93	50.62	54.30	54.35	57.20	57.13	60.33
SimCLR [54]	43.06	56.22	51.44	59.72	37.97	56.46	15.31	50.40	54.18	57.23	41.50
RotNet [48]	72.05	69.75	<u>75.28</u>	<u>70.71</u>	<u>59.26</u>	60.87	34.58	56.64	51.82	53.84	62.38
OC-FakeDect [96]	<u>74.10</u>	<u>69.95</u>	54.16	54.27	49.84	<u>65.82</u>	<u>41.35</u>	<u>76.50</u>	<u>63.81</u>	<u>60.32</u>	57.90
UNTAG (Ours)	**74.71**	**70.64**	**81.81**	**75.61**	**82.81**	**76.87**	**91.14**	**87.30**	**77.02**	**70.70**	80.03

Table 3.1: AUC (%) and Accuracy (Acc. in %) of UNTAG compared to the selected baselines on five different datasets. The best results are highlighted in **bold**. The second best results are <u>underlined</u>. The sub-blocks from top to bottom show supervised, self-supervised, and unsupervised methods, respectively.

unsupervised deepfake detection called **OC-FakeDect** [96]; (5) **SimCLR** [54] a contrastive self-supervised generic approach, and (6) a generic unsupervised one-class classification method entitled **RotNet** [48] supported by a non-contrastive pretext task. It generates image features in a similar way to UNTAG. For evaluating DFD-HF [111] and DSP-FWA [103] in terms of both classification and quality of features, two variants are proposed: the first one is based on a supervised classifier, while the second employs a GMM-based unsupervised classification similar to UNTAG. More specifically, we first consider the two original models and attempt to directly classify images as fake or real. In the second variant, the classification layer is discarded and a GMM is fitted to detect deepfakes in an unsupervised manner. In this case, the two methods are denoted by **DFD-HF-OC** and **DSP-FWA-OC**, respectively. This allows carrying out a fair comparison with our approach, while evaluating the relevance of the unsupervised setting for type-agnostic deepfake detection.

Datasets and Experimental Settings. In the remainder of this chapter, we refer to **f**ace-**s**waps as **FS**, **f**acial **r**eenactments as **FR**, **f**acial **a**ttribute **m**anipulations as **FAM**, **f**ully **s**ynthetic **f**aces (GAN-generated faces) as **FSF**, the combination of the four aforementioned types as **Multi**, and the combination of the four aforementioned types with stacked manipulations as

Method / Subset	SimCLR [54]		RotNet [48]		OC-FakeDect [96]		UNTAG (Ours)	
	AUC	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**
Celeb-DF [24]	43.06	56.22	72.05	69.75	74.10	69.95	**74.71**	**70.64**
FF++ DF [23]	52.30	58.71	73.67	70.70	48.31	52.79	**83.45**	**76.14**
FF++ FS [23]	55.89	60.96	74.03	70.36	48.15	52.57	**76.64**	**72.65**
FF++ FaceShifter [112]	50.22	59.41	74.61	70.60	68.54	58.68	**85.19**	**80.13**
BlendFace [26]	50.10	56.91	44.47	51.73	54.02	52.18	**71.17**	**66.06**
FaceShifter [112]	61.22	60.26	53.11	54.75	63.85	60.87	**73.94**	**67.95**
DeepFakes [6]	55.73	59.56	59.88	59.05	66.41	63.30	**83.45**	**76.14**
MMReplacement [26]	52.94	58.14	58.06	57.81	63.94	61.20	**78.91**	**72.29**
Mean AUC	52.68±4.97		63.74±10.74		60.92±9.01		78.43±4.82	

Table 3.2: AUC (%) and Accuracy (Acc. (%)) on Face Swap (FS) deepfake generation methods.

Method / Subset	SimCLR [54]		RotNet [48]		OC-FakeDect [96]		UNTAG (Ours)	
	AUC	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**
FF++ NT [113]	59.58	62.26	75.08	71.42	50.95	53.16	**76.57**	**72.09**
FF++ F2F [114]	61.46	61.73	75.51	71.41	50.65	53.50	**76.93**	**72.97**
ATVGNet [115]	53.93	59.14	49.73	55.26	62.25	59.29	**75.18**	**69.57**
FOMotion [116]	56.67	59.20	46.87	52.90	69.21	64.89	**75.78**	**69.70**
TalkingHead [117]	54.86	59.07	47.22	52.30	66.78	61.17	**75.03**	**69.39**
Mean AUC	57.30±2.84		58.88±13.44		59.97±7.81		75.90±0.75	

Table 3.3: AUC (%) and Accuracy (Acc. (%)) for Face Reenactment (FR) deepfake generation methods.

Multi+, respectively. Figure 3.4 presents examples of artifacts that commonly appear in different facial regions, along with the deepfake types they are associated with. Three well-known datasets are considered for the experiments, namely, ForgeryNet [26] (Multi+), FaceForensics++ [23] (FS, FR), and Celeb-DF [24] (FS). In addition, we generate two datasets using StarGAN2 [97] (FAM) and StyleGAN2 [98] (FSF). ForgeryNet [26] is a recently introduced dataset. Compared to other datasets, it has the advantage to include all types of deepfakes. Table 3.7 provides further details regarding the dataset statistics and our protocol. During testing, balanced sets of about 2000 samples are utilized. Forged data is randomly sampled from forgery datasets, whereas real data is randomly sampled from the

Method / Subset	SimCLR [54]		RotNet [48]		OC-FakeDect [96]		UNTAG (Ours)	
	AUC	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**
StarGAN2* [97]	15.31	50.40	34.58	56.64	41.35	76.50	**91.14**	**87.30**
SC-FEGAN [118]	61.57	59.73	52.00	53.69	59.26	56.51	**70.16**	**65.78**
MaskGAN [119]	53.05	55.63	49.44	51.42	61.46	59.25	**75.73**	**69.52**
StarGAN2 [97]	54.97	56.96	56.15	56.28	67.55	64.01	**79.18**	**71.92**
Mean AUC	46.23±18.13		48.04±8.13		57.40±9.75		79.05±7.69	

Table 3.4: AUC (%) and accuracy (Acc. (%)) for Face Attribute Manipulation (FAM) deepfake generation methods.

Method / Subset	SimCLR [54]		RotNet [48]		OC-FakeDect [96]		UNTAG (Ours)	
	AUC	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**	**AUC**	**Acc.**
StyleGAN2* [98]	31.42	55.05	61.08	62.00	53.03	66.30	**80.82**	**74.65**
StyleGAN3 [99]	56.62	61.05	52.36	59.74	50.47	65.30	**85.08**	**78.46**
StyleGAN2 [98]	61.19	60.34	49.23	51.60	62.45	59.25	**78.22**	**71.67**
Mean AUC	49.74±13.09		54.22±5.01		55.31±5.15		81.37±2.83	

Table 3.5: AUC (%) and Accuracy (Acc. (%))for Fully Synthetic Faces (FSF) deepfake generation methods.

ForgeryNet validation set [26]. An exception is made for the StyleGAN3 [99]: about 300 samples are utilized[1]. Additionally, ForgeryNet takes into account more than 5400 subjects in contrast to previous benchmarks such as Celeb-DF which consider only 59 individuals. Hence, mixing real data from ForgeryNet [26] with fake data from the targeted datasets ensure differentiating between the identity leakage [94, 33] phenomenon and actual high deepfake detection performance. In the experiments, we report the Area Under the ROC Curve (AUC) and the Accuracy (Acc.)

Implementation details. Since we train our model using real data only, we randomly sample a subset of $142,371$ real images from the ForgeryNet training set. First, the 256×256 images are extracted with MediaPipe [124]. The details of landmarks of each region are provided in Table 3.11. R-Splicer generates from real data $20,406$ images annotated with

[1]This is mainly due to the fact that the authors released only this amount of curated data at the preparation of this work.

Subset \ Method	Types	SimCLR		RotNet		OC-FakeDect		UNTAG (Ours)	
		AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.
FS-GAN [120]	FS+FR	56.82	59.23	56.76	57.19	58.65	56.74	**80.94**	**74.74**
DiscoFaceGAN [121]	FSF+FAM	55.15	58.56	64.65	62.39	71.72	67.22	**80.97**	**74.45**
StarGAN2+BlendFace [26]	FAM+FS	40.93	53.81	45.74	51.78	60.03	57.63	**71.40**	**67.24**
StarGAN2+Deepfakes [26]	FAM+FS	59.54	60.20	58.42	58.06	70.38	67.80	**82.56**	**75.40**
Mean AUC		53.11±7.20		56.39±6.82		65.19±5.89		78.97±4.42	

Table 3.6: AUC (%) and Accuracy (Acc. (%)) for combined manipulations involving Face Swap (FS), Face Reenactment (FR), Fully Synthetic Faces (FSF), and Face Attribute Manipulation (FAM).

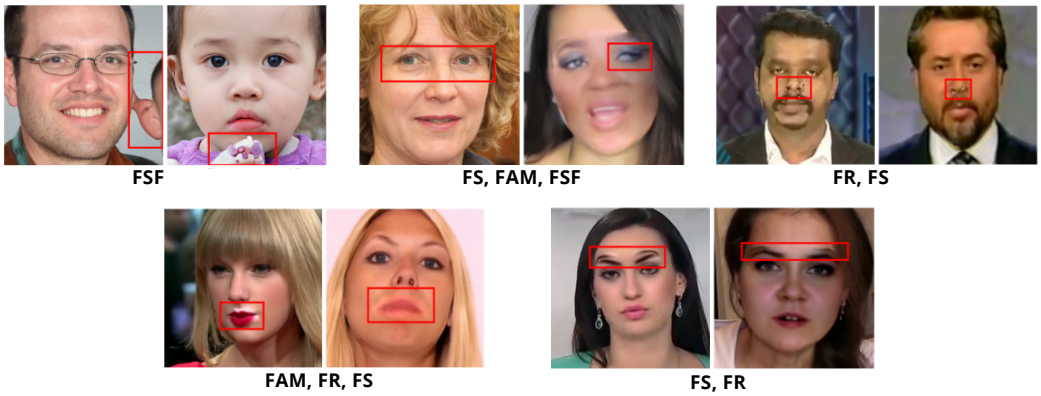


Figure 3.4: Common artifacts and their corresponding deepfake types. Images are randomly samples from StyleGAN2 [98] and FF++ [23].

Dataset	Types	Approaches	Subjects	Videos		Images		Samples Used	
				Real	Fake	Real	Fake	Train	Test
FF++ [23]	FS, FR	5	-	1,000	5,000	-	-	0	2,200
Celeb-DF [24]	FS	1	59	590	5,639	-	-	0	1,221
StyleGAN3 [99]	FSF	1	386	-	-	-	386	0	534
StarGAN2 [97]	FAM	1	-	-	-	-	-	0	1,000
ForgeryNet [26]	Multi+	15	5400+	99,630	121,617	1,438,201	1,457,861	142,371	2,200

Table 3.7: Specifications of the selected datasets, including the number of training and testing samples used in our experiments. An equivalent number of real images is selected from ForgeryNet [26] to avoid identity leakage [32, 122].

the manipulated region. For the pretext task, the annotated images are used to fine-tune a ResNet-18 [123] for 107 iterations with a learning rate of 3.10^{-5}. UNTAG is implemented using Pytorch [125] and trained with an NVIDIA Titan V GPU. Basic data augmentation op-

w/ pretext task	w/ GMM	Celeb-DF [24]		FF++ [23]		StyleGAN [98]		StarGAN [97]		ForgeryNet [26]	
		AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.
✓	✗	62.51	61.17	54.53	53.75	45.64	50.28	18.17	50.10	55.29	57.73
✗	✓	27.43	53.55	54.85	43.21	69.75	73.99	65.46	70.42	24.82	51.20
✓	✓	**74.71**	**70.64**	**81.81**	**75.61**	**82.81**	**76.87**	**91.14**	**87.30**	**77.02**	**70.70**

Table 3.8: Ablation on the role of each component in UNTAG, namely the pretext task as a direct binary classifier and the GMM as a direct One-Class Classification (OCC) as a Deepfake detector.

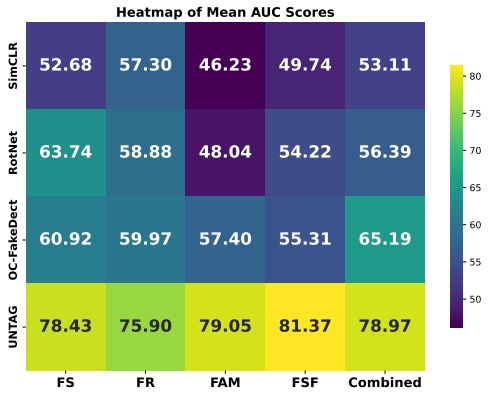


Figure 3.5: Heatmap of the Mean AUC scores, summarizing the overall inter-type generalization of UNTAG and the selected baselines.

erations such as random horizontal flipping and random gray-scaling are applied. A GMM model with 3 components is fitted using Scikitlearn [126]. The number of Gaussian components is empirically fixed. The batch size is fixed to 32 and 512 for the first and second steps, respectively.

3.5.2 Results

Comparison with the baselines. Table 3.1 reports the obtained results on the five considered datasets. UNTAG clearly outperforms state-of-the-art methods in all the datasets. Overall, unsupervised classification-based methods like SimCLR [54], RotNet [48], OC-FakeDetect [96] and UNTAG are more effective for learning features that are robust to different types of forgeries. In contrast, methods that are learned in a supervised manner seem to not be suitable for type-agnostic deepfake detection. Xception-Ext, DFD-HF [111]

	Celeb-DF [24]		FF++ [23]		StyleGAN [98]		StarGAN2 [97]		ForgeryNet [26]	
Feature Extractor	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.
EfficientNet-B0 [22]	59.00	62.83	70.76	68.34	55.82	63.30	79.56	75.30	52.07	56.64
ResNet34 [123]	71.81	69.77	**84.22**	**77.36**	78.21	76.02	**95.36**	**90.50**	56.34	58.30
ResNet18 [123]	**74.71**	70.64	81.81	75.61	**82.81**	**76.87**	91.14	87.30	**77.02**	**70.70**

Table 3.9: Performance of UNTAG under different backbones (Stage 1) when combined with a GMM as the one-class classifier in Stage 2.

	Celeb-DF [24]		FF++ [23]		StyleGAN [98]		StarGAN2 [97]		ForgeryNet [26]	
One-class Classifier	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.
OC-SVM [68]	19.91	**75.02**	29.02	62.09	27.52	57.86	24.91	62.30	21.89	70.45
KDE [75]	74.27	70.40	74.54	70.02	70.45	68.63	80.24	71.90	**78.11**	**72.00**
GMM [74]	**74.71**	70.64	**81.81**	**75.61**	**82.81**	**76.87**	**91.14**	**87.30**	77.02	70.70

Table 3.10: Performance of UNTAG using a ResNet18 at Stage 1 while considering various one-class classifiers in Stage 2.

and DSP-FWA [103] have a significantly lower performance than unsupervised methods. In fact, despite the fact that DFD-HF [111] achieves an AUC of 91.63% and an accuracy of 83.81% using the original protocol of [111], changing the testing set impacts its performance to a large extent. This suggests that the model has learned the identity of the subjects rather than subject-independent features. Similarly, the supervised Xception classifier which was extended with UNTAG's pretext task is not capable of effectively detecting deepfakes. Another observation can be made regarding self-supervision: DSP-FWA [103] achieves noticeably higher performance than DFD-HF [111] and Xception-Ext [21], even when the classification is supervised. Finally, the irrelevance of the features generated by DFD-HF [111] is confirmed when observing its unsupervised variant. In fact, the performance drops importantly when using DFD-HF-OC, in contrast to DSP-FWA-OC, which learns from simulated warping artifacts.

Framework study. We report in Tables 3.9 and 3.10 variants of UNTAG using different backbones and various one-class classifiers. We observe that most variants are competitive overall and support the proposed framework, except for OC-SVM, which shows very low performance on some datasets. This might be due to the deterministic nature of OC-SVM;

Region	Mediapipe [124] Landmark identifiers (IDs)
Face†	[108, 68, 143, 213, 210, 208, 426, 430, 433, 372, 298, 337]
Eyebrows	[9, 68, 156, 124, 53, 52, 8, 282, 283, 353, 333, 298]
Eyes	[8, 222, 224, 35, 230, 6, 450, 265, 445, 442]
Nose	[193, 203, 164, 423, 417]
Mouth	[164, 165, 212, 200, 432, 391]

Table 3.11: Mediapipe [124] landmark IDs per region. † denotes the full face region used to splice the background region.

therefore, supporting the use of a probabilistic classifier. We note that overall, the best performance is achieved using the combination (Resnet18 + GMM) considered in the chapter.

Evaluation of the generalization capabilities. Tables 3.2, 3.3, 3.4, 3.5, and 3.6 show the detailed performance of our method compared to SimCLR [54], RotNet [48] and OC-FakeDect [96]. The performance obtained for each manipulation in the considered datasets is reported separately. The results show that UNTAG also outperforms SimCLR [54] RotNet [48] and OC-FakeDect [96] regardless of the considered manipulation type. This success could be explained by the relevance of the proposed self-supervision task for deepfake detection. In fact, the self-supervision employed by RotNet [48] which is based on rotation predictions are less suitable for type-agnostic deepfake detection. Similarly, SimCLR [10], which is a contrastive-based self-supervision approach, achieves lower generalization performance than UNTAG. This is further confirmed in Figure 3.5. Additionally, Figure 3.6 showing the t-Distributed Stochastic Neighbor Embedding (t-SNE) visualizations of the real and fake embeddings produced by OC-FakeDect [96] and UNTAG and its variants confirm that UNTAG yields overall more generic representations than OC-FakeDect [96]. Indeed, real and fake embeddings seem to be more easily separable for all the UNTAG variants as compared to OC-FakeDect [96]. Also, the t-SNE visualization embeddings of UNTAG suggests that real data are for most datasets distributed over several small clusters; thus highlighting the relevance of the GMM-based modeling.

Ablation Study. The role of each UNTAG's component, namely the proposed pretext task and one-class classification, is investigated. In Table 3.8, the ablation study results are reported. First, we consider the pretext task as a standalone classifier. To this end, the pretext

task network is retrained as a binary classifier detecting spliced and non-spliced images and used for detecting deepfakes at inference. The results show that the network is only sensitive to face swaps as in Celeb-DF [24], but performs poorly on GAN-generated images. Second, instead of fitting the GMM model with real-image embeddings, we directly use the set of $142,371$ authentic images to estimate the model parameters. The results show that the GMM can distinguish between real and GAN-generated images, suggesting that these images have inherently different generation processes. These experiments show that our pretext task and one-class classification are complementary and justifies their use for type-agnostic deepfake detection.

Dataset	Forgery Type	Number of Gaussians		
		1	2	3
Celeb-DF[24]	FS	73.89	73.87	**74.71**
FF++[23]	FS,FR	81.03	81.01	**81.81**
StyleGAN[98, 99]	FSF	81.91	82.68	**82.81**
StarGAN2[97]	FAM	82.76	89.41	**91.14**
ForgeryNet[26]	Multi+	76.17	76.53	**77.02**
Mean AUC		79.15	80.70	**81.50**

Table 3.12: The AUC performance of UNTAG using different number of Gaussian components for classification. Bold results highlight the best performance.

Number of Gaussian Components. In Table 3.12, we vary the number of GMM components and report the AUC of UNTAG, accordingly. The best performance is obtained when using 3 components. Thus, all the results are reported in the chapter using 3 components.

Ablation on the contribution of the background and extension to non-natural image domains. Table 3.13 reports UNTAG's accuracy with and without background splicing on in-domains and out-of-domain data. The results generally show a slight improvement, especially on diffusion [128] (DDPM) Table 3.13 reports the testing results on images generated using diffusion models [128], animal faces [97] (AFHQ) and portraits [129] (Metfaces). Two observations can be made: (1) UNTAG achieves promising and consistent performance on diffusion models, which is reasonable as artifacts in fully synthetic images occur in the back-

	In-Domain (ID)					Out-of-Domain (OOD)		
Splicing	Celeb-DF	FF++	StyleGAN	StarGAN2	ForgeryNet	diffusion	AFHQ	Metfaces
w/o bg	70.41	75.50	76.77	**87.60**	70.59	73.50	55.52	54.69
w/ bg	**70.64**	**75.61**	**76.87**	87.30	**70.70**	**76.86**	**55.65**	**54.85**

Table 3.13: AUC (%) of background splicing on different datasets including diffusion, animals and portraits.

ground as well. (2) Since our model was trained on real human faces, our method does not generalize to other datasets with a different semantic viewpoint, even when the subjects are human portraits [129].

Limitations. Despite the relevance of the learned representations as compared to the considered baselines, the t-SNE visualization of the fake and real embeddings produced by UNTAG demonstrates some limitations Figure 3.6. Indeed, the embeddings are overall globally separable, but there still exists a significant overlap between them. The clusters of real and fake data remain too close, therefore, it would be interesting to incorporate in a future work a contrastive loss to enhance the distinction between the two distributions.

3.6 Conclusion

In this chapter, the problem of deepfake detection has been formulated as an unsupervised type-agnostic problem. A solution termed UNTAG using a one-class classifier and a self-supervision mechanism has been proposed. In particular, a novel auxiliary task specifically tailored for deepfake detection has been introduced. It aims at learning discriminative features by detecting manipulated regions with a simple splicing-blending technique. Finally, a GMM is fitted to the learned representations of the real data. As a result, deepfakes can be detected as anomalies regardless of their types without using any data annotation. UNTAG achieves an encouraging inter-type generalization capabilities, while only relying on real data for training.

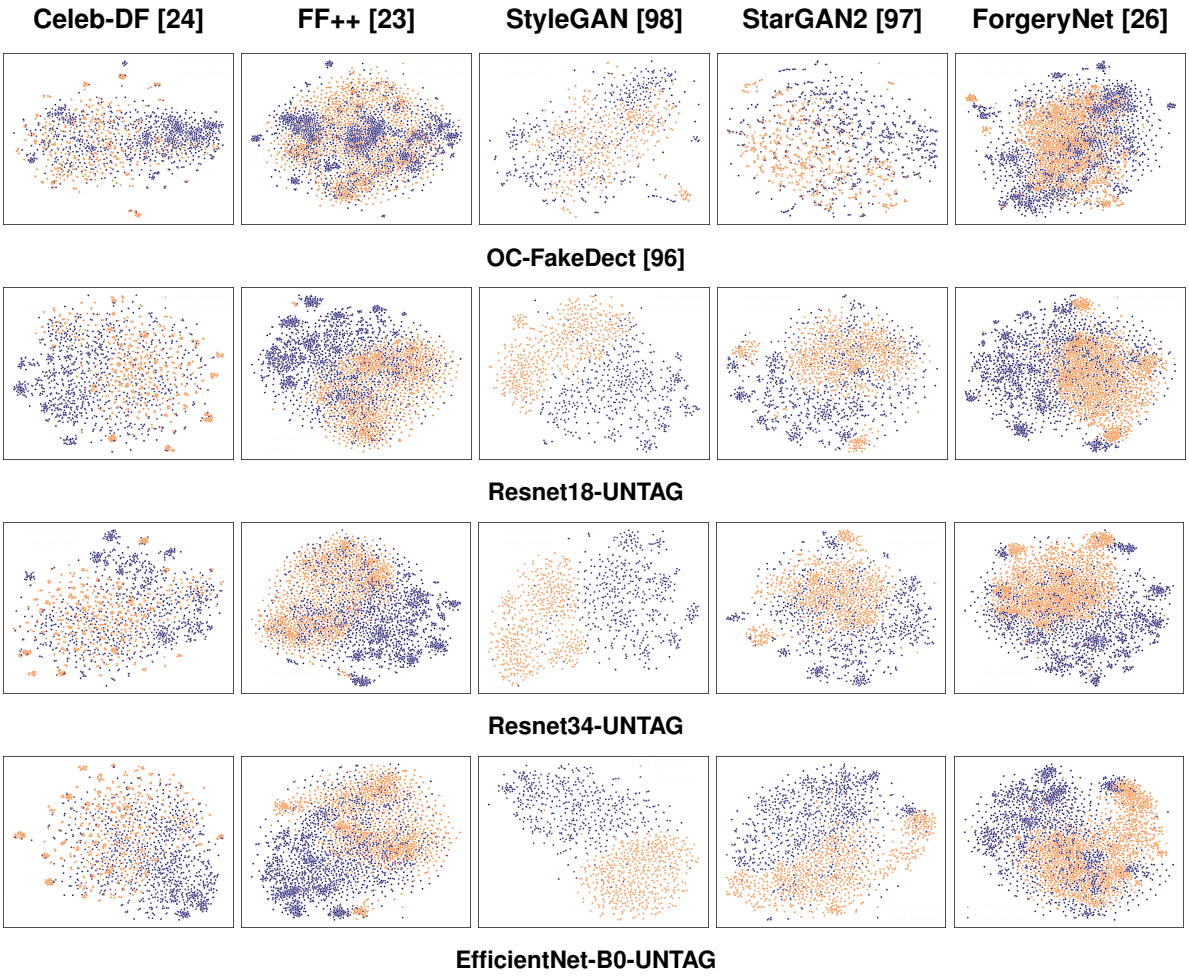


Figure 3.6: t-Distributed Stochastic Neighbor Embedding (t-SNE) [127] visualizations of the real and fake embeddings for OC-FakeDect [96] and UNTAG. Darker points represent real samples, while lighter points correspond to fake samples.

Chapter 4

Unsupervised Anomaly Detection in Time Series: An Extensive Evaluation and Analysis of State-of-the-art Methods

The previous chapter introduced UNTAG, a method enabling type-agnostic deepfake detection at the image level. Nevertheless, in realistic scenarios, deepfakes are not limited to static images; they can also be videos. This motivates our investigation of unsupervised anomaly detection techniques applied to temporally structured data. In fact, videos can be interpreted as multivariate time Series, making it natural to examine the existing literature on unsupervised multivariate Time-series Anomaly Detection (TSAD) techniques. Since these approaches have not been applied to deepfake detection in the past, assessing their maturity and understanding which underlying paradigms are best suited for a real-world task like deepfake detection becomes a necessary step before considering their applicability to this context.

To this end, this chapter presents an evaluation of recent unsupervised multivariate time-series anomaly detection methods. Although the unsupervised multivariate TSAD literature

is extensive, a comprehensive evaluation study taking into account real-world constraints is still needed. Existing studies often evaluate primarily on standard metrics such as precision, recall, and F1-score, overlooking other important assessment aspects such as (i) considering performance metrics specifically tailored for time-series data, (ii) analyzing the model size and stability, (iii) evaluating across different anomaly types, and (iv) considering a clear and a unified experimental protocol. The overall objective of this work is, therefore, to gain insights into the suitability of these approaches for real-world tasks such as deepfake detection.

4.1 Introduction

A multivariate time-series corresponds to a temporally ordered set of variables. This mathematical representation has been used in countless domains, such as finance, health, and biomechanics. Designing methods for automatically analyzing time-series (e.g., forecasting, classification, anomaly detection) has been widely investigated by researchers [130, 131]. A particular focus is given to anomaly detection in time-series [132, 133]. In general, an anomaly or outlier can be defined as an observation or sample that does not follow an expected pattern. The popularity of anomaly detection in time-series is probably due to its interest in numerous industrial contexts. As an example, one can mention the detection of faulty sensors [134], fraudulent bank transactions [135], and pathologies in medical data [136, 137].

In the literature, some attempts have been made to develop supervised and semi-supervised approaches [138, 139]. Although supervised techniques may achieve higher detection performance on anomalies seen during training, they usually risk overfitting those anomalies, resulting in poor generalization to novel outliers. Semi-supervised approaches offer a more flexible solution leveraging both labeled and unlabeled data [140]. However, despite being promising [141, 140], these methods still rely on a certain amount of annotated data, which can be constraining. Hence, the task of time-series anomaly detection is usually formulated as an unsupervised problem [142]. In fact, since anomalies occur rarely, annotating data

Study	Standard eval. metrics	Range-based eval. metrics [144]	Net. size	Eval. of recent DL methods	Comp. w/ ML methods	Per Anomaly type analysis	Unified exp. protocol	Model stability
[145]	Yes	No	No	Yes	No	No	No	No
[146]	Yes	No	No	No	Yes	Yes	Yes	No
[133]	No	No	No	No	No	No	No	No
[147]	Yes	No	No	Yes	No	No	Yes	No
[148]	Yes	Partial	Partial	No	Yes	Yes	Partial	No
[149]	Yes	No	No	Yes	Yes	No	No	Yes
[150]	Yes	No	No	Yes	No	No	Partial	No
[151]	Yes	No	Partial	Yes	No	No	Partial	No
[152]	Yes	Partial	No	No	Yes	Yes	No	Yes
[153]	No	No	Partial	Yes	No	No	No	No
[154]	No	Partial	Partial	Yes	Yes	No	Partial	No
[155]	Yes	No	Partial	Yes	Yes	No	No	No
Ours	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 4.1: Comparison of existing evaluation studies of anomaly detection in time-series: we specify which of the following aspects were taken into account: (1) standard performance metrics which correspond to the precision, recall, and F1-score; (2) revisited performance metrics extending the precision, recall, and F1-score to time-series introduced by Tatbul et al. [144] ; (3) network size; (4) consideration of ML approaches in the comparison; (5) evaluation of recent deep learning techniques; (6) analysis with respect to the types of anomalies; and (7) use of a unified experimental protocol. Note that by “partial", we mean that the authors briefly discussed the concept without necessarily producing any related comparison or results in their study.

becomes challenging and costly. This makes unsupervised learning more adequate despite being exposed to additional challenges such as the lack of explicit guidance and complex hyper-parameter tuning [143]. In this article, we focus on the topic of unsupervised anomaly detection in time-series.

Earlier methods of anomaly detection in time-series mostly employed traditional Machine Learning (ML) [156, 157] and auto-regressive [158, 159] techniques. However, as discussed in [145], these approaches are mainly subject to *the curse of dimensionality*. In other words, their performance drops in the presence of high-dimensional time-series.

To address this, motivated by the tremendous advances in Deep Learning (DL), massive efforts have been recently made to design suitable Deep Neural Network (DNN) architectures [160, 161, 142]. These DL-based approaches have achieved impressive performance in terms of *standard performance metrics* (precision, recall, and F1-score). Nevertheless, despite their promising results, their suitability in a realistic industrial context still needs further investigation. For that purpose, it is timely to propose an extensive comparison of recent unsupervised DL techniques that consider the following aspects:

(i) Model size and model stability: Existing methods overlook the *model size* and the *model stability*, which are important indicators of the scalability and the performance stability. By a stable model, we mean a model which has stable performance under different training trials.

(ii) Unified experimental protocol: There is no clear experimental protocol for evaluating state-of-the-art methods. As a consequence, it can be noted that the reported experimental values vary considerably from one chapter to another. For instance, as highlighted by Kim et al. [147], a peculiar evaluation protocol called Point Adjustment (PA) introduced by Xu et al. [161] is often used [162, 142], while it is ignored in other cases [163, 164].

(iii) Performance metrics for time-series: As discussed by Tatbul et al. [144], the used standard performance metrics (precision, recall, and F1-score) might not be entirely adequate for evaluating time-series anomaly detectors. These metrics were initially designed for time-independent predictions and not for range-based ones. As an alternative, [144] extended these metrics to time-series. However, it can be noted that current state-of-the-art methods do not consider these relatively novel evaluation criteria.

(iv) Experimental analysis with respect to the anomaly type: a detailed experimental evaluation with respect to the type of anomaly is missing in the state-of-the-art. Significant efforts have been dedicated to rigorously defining the different possible types of outliers in time-series [145, 146]. However, no detailed experimental analysis has been carried out in that direction.

(v) Comparison against ML methods: Similar to the works of [133, 149], we emphasize the importance of comparing traditional ML strategies to DL approaches. Recent stud-

ies [153, 151, 150] tend to focus on DL-based works, often overlooking ML techniques. Our findings are consistent with [133, 149], and indicate that ML methods remain relevant for the task of unsupervised anomaly detection in time-series.

In the literature, some survey studies were proposed for unsupervised time-series anomaly detection [153, 154]. They primarily focus on presenting recent approaches, their relevant applications and their challenges and limitations. Some other works [133, 152] have conducted experiments to identify the flaws of current benchmark datasets and scoring functions, proposing new datasets and issuing recommendations for practitioners. While few other evaluation studies focused on experimentally comparing recent anomaly detection algorithms [155, 149, 145, 150, 147, 146, 151, 152, 148]. Our work belongs to this latest category. For instance, [145] present a brief comparison of recent DL algorithms in terms of precision, recall, and F1-score but neglect the model size and model stability. We can also mention the work of [146], where a new taxonomy for time-series outliers is proposed. Then, based on that, a methodology to generate synthetic datasets is suggested. They finally compare nine different algorithms according to outlier types but they do not include the latest DL algorithms. Nevertheless, similar to [145], they only focus on classical evaluation criteria, omitting range-based evaluation, model stability and sizes. Furthermore, [147] present a rigorous evaluation of recent DL techniques by questioning the Point Adjustment protocol. Nevertheless, the model size and model stability, as well as the performance metrics for time-series are not considered. [148] propose a large-scale evaluation study of existing anomaly detection methods, thereby assessing the overall progress made in this field. Nevertheless, they do not investigate the conceptual differences and limitations of different types of approaches. In addition, recent state-of-the-art deep learning methods published in top-tier venues such as [162, 160] are not considered. Last but not least, while they attempt to readapt the AUC using the recently introduced range-based metrics [144], they do not report the range-based precision, recall, and F1-score that are essential for an in-depth comparative study of existing methods, which is the core objective of the present chapter. The work presented by [151] compares DL techniques with and without the Point Adjustment protocol under different federated learning settings using classical metrics only.

Similarly, [150] provide a comprehensive review of DL-based anomaly detection for time-series, detailing fundamental principles, applications, and guidelines for practitioners. They compare several DL-based approaches using classical metrics but do not consider practical aspects like model size and stability. The study of [152] proposes a benchmark for evaluating univariate anomaly detection methods, mostly targeting classical ML approaches including only few DL methods and without adopting a unified protocol. Furthermore, [149] report a per-benchmark analysis between conventional, ML-based, and DL-based approaches, accounting for model stability but lacking range-based evaluation, per-anomaly type analysis, and a clear unified protocol. Lastly, [155] focus on evaluating multivariate techniques without reporting any range-based performance, or per-anomaly type analysis.

Hence, in this survey, we provide a comprehensive evaluation study of recent state-of-the-art algorithms by taking into account all the mentioned aspects $(\mathbf{i})$ to $(\mathbf{v})$. As summarized in Table 4.1, an analysis using standard performance metrics, as well as the novel performance metrics proposed by Tatbul et al. [144] is performed. In addition, the number of parameters of DL-based approaches is reported as it directly impacts the memory consumption and the model scalability. Moreover, experiments according to the nature of anomalies are carried out using the taxonomy that was recently introduced by Lai et al. [146]. Lastly, a unified experimental protocol is used to compare existing methods. In short, this work aims to provide a comprehensive evaluation of numerous paradigm-representative time-series anomaly detection techniques, including recent deep learning methods, for a better assessment of their practical relevance. For that purpose, additional aspects are considered in complement to the traditional performance metrics, such as employing a unified experimental protocol, using range-based performance metrics, analyzing the performance based on the type of anomalies, and studying the model size and stability. The aim of this study is to help the community understand the advantages and limitations of state-of-the-art techniques from a broader applicative perspective and lay the foundations for better experimental evaluation practices.

The remainder of this chapter is organized as follows. Section 4.2 presents preliminaries necessary for the understating of this chapter. Section 4.3 reviews state-of-the-art time-

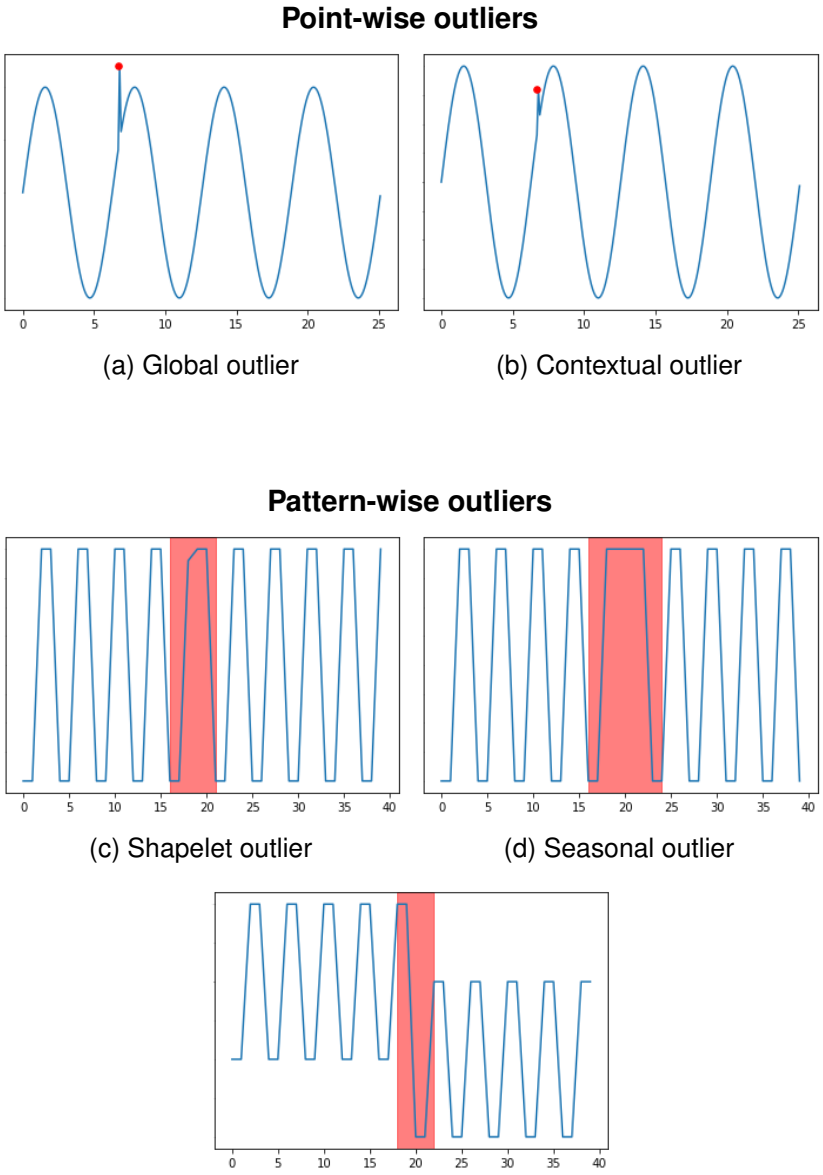
series anomaly detection methods. Section 4.4 describes the used datasets and details the evaluation protocol considered in the experiments. Section 4.5 presents and analyzes the results. Finally, Section 4.6 concludes this work.

4.2 Preliminaries

A time-series is a temporally ordered set of n variables which can be denoted by $X = \{X_t\}_{1 \leq t \leq N}$ where $X_t \in \mathbb{R}^n$ refers to the n-dimensional vector of variables at an instant t. Note that the time-series is univariate if $n = 1$, and is multivariate otherwise ($n > 1$). This section reviews the necessary background for a better understanding of this survey. Specifically, we start by recalling the different types of time-series anomalies according to the taxonomy of [146]. Then, we present the usual paradigms employed for anomaly detection in time-series.

4.2.1 Types of anomalies

As discussed in [145], anomalies in time-series can generally be classified into three main categories, namely, *point*, *contextual*, and *collective* anomalies. However, unlike point anomalies, the definitions of contextual and collective ones are more ambiguous in the state-of-the-art, as stated by Lai et al. [146]. Indeed, they are heavily impacted by the application context. For instance, [165] defines contextual anomalies as small temporal segments formed by neighboring points, while [166] considers them as seasonal points (occurring periodically). Lai et al. have recently refined the definition of outlier types [146]. They distinguish between *point-wise outliers* and *pattern-wise outliers*. The former is formed by *global* and *contextual outliers* while the latter is composed of *shapelet*, *seasonal*, and *trend outliers*. In the following, the taxonomy proposed by Lai et al. [146], which is central to our analysis, is recalled.



(a) Global outlier (b) Contextual outlier

(c) Shapelet outlier (d) Seasonal outlier

(e) Trend outlier

Figure 4.1: Examples of the five different types of outliers proposed in [146].

Point-wise anomalies

Point-wise outliers are local anomalies occurring on individual time stamps. Let $X = \{X_t\}_{1 \leq t \leq N}$ be a multivariate time-series and $\hat{X}_t$ the expected value of X_t at an instant t according to a regression model. Given a well-chosen threshold $\delta > 0$, an anomaly at an instant t can be formally defined by,

$$\|X_t - \hat{X}_t\| > \delta, \tag{4.1}$$

where $\|.\|$ defines an L_p norm.

Global outliers. They can be seen as point-wise anomalies which importantly deviate from the rest of the points in a time-series. They usually correspond to spikes in the time-series, as shown in Figure 4.1a. In this case, the threshold δ can be formulated as,

$$\delta = \lambda\sigma(X), \tag{4.2}$$

where $\sigma(.)$ refers to the standard deviation operator and $\lambda \in \mathbb{R}^{+*}$.

Contextual outliers. They refer to individual points which differ significantly from their neighbors. The latter are often small glitches in the time-series as illustrated in Figure 4.1b. The threshold can be defined as,

$$\delta = \lambda\sigma(X_{t-k:t+k}), \tag{4.3}$$

where $X_{t-k:t+k} = \{X_{t-k}, X_{t-k+1}, ..., X_{t+k}\}$ is the signal corresponding to the temporal window centered on t. The function $\sigma(.)$ refers to the standard deviation operator and $\lambda \in \mathbb{R}^{+*}$.

Pattern-wise anomalies

Pattern-wise anomalies refer to anomalous sub-sequences which typically showcase discords or irregularities. These anomalies are defined by Lai et al. [146] by modeling a time-series X with spectral structural analysis [167] as follows,

$$X = \rho(2\pi\omega T) + \tau(T), \tag{4.4}$$

such that $\rho(2\pi\omega T) = \sum_k [A\sin(2\pi\omega_k T) + B\cos(2\pi\omega_k T)]$ corresponds to the base *shapelet* function which can be interpreted as the characteristic shape of X. The *seasonality*, which describes a pattern occurring at specific regular intervals in a time-series, is modeled with $w = \{w_1, w_2, ..., w_k\}$. Finally, a trend function denoted by τ defines the global direction of X. In particular, a sub-sequence $X_{i:j}$ of a time-series X with $1 \leq i < j \leq N$ can be formulated using a shapelet function such that,

$$X_{i:j} = \rho(2\pi\omega T_{i,j}) + \tau(T_{i,j}), \tag{4.5}$$

with ρ, ω, τ, and $T_{i,j}$ respectively being the shape, the seasonality, the trend, and the time-stamps of the sub-sequence. The analysis of the shapelet, the seasonality as well as the trend functions allow distinguishing the three following outliers:

Shapelet outliers. They represent the anomalous sub-sequences enclosing shapelets that are different from the expected ones, as shown in Figure 4.1c. The following condition can be used to define shapelet outliers as follows,

$$d_\rho(\rho(.), \hat{\rho}(.)) > \delta, \tag{4.6}$$

with d_ρ being a dissimilarity measure computed between two sets of shapelets. $\hat{\rho}(.)$ corresponds to the expected shapelets in a given sub-sequence and δ is the threshold.

Seasonal outliers. They can be defined as sub-sequences with unexpected seasonalities with respect to the full sequence, as illustrated in Figure 4.1d.

$$d_\omega(\omega, \hat{\omega}) > \delta, \tag{4.7}$$

with d_ω being a dissimilarity measure between two seasonality, $\hat{\omega}$ being the expected seasonality in the sub-sequence, and δ being the threshold.

Trend outliers. They refer to sub-sequences with an importantly altered trend. Consequently, a shift in the mean data can be observed, as shown in Figure 4.1e. Mathematically,

trend outliers can be defined by,

$$d_\tau(\tau, \hat{\tau}) > \delta \tag{4.8}$$

where d_τ is a dissimilarity measure computed between two trends, $\hat{\tau}$ is the expected trend of the sub-sequence, and δ is the threshold.

4.2.2 Paradigms for anomaly detection in times-series

Existing anomaly detection methods in time-series mainly employ five different paradigms, namely, clustering-based, density estimation-based, distance-based, reconstruction-based and forecasting-based methods.

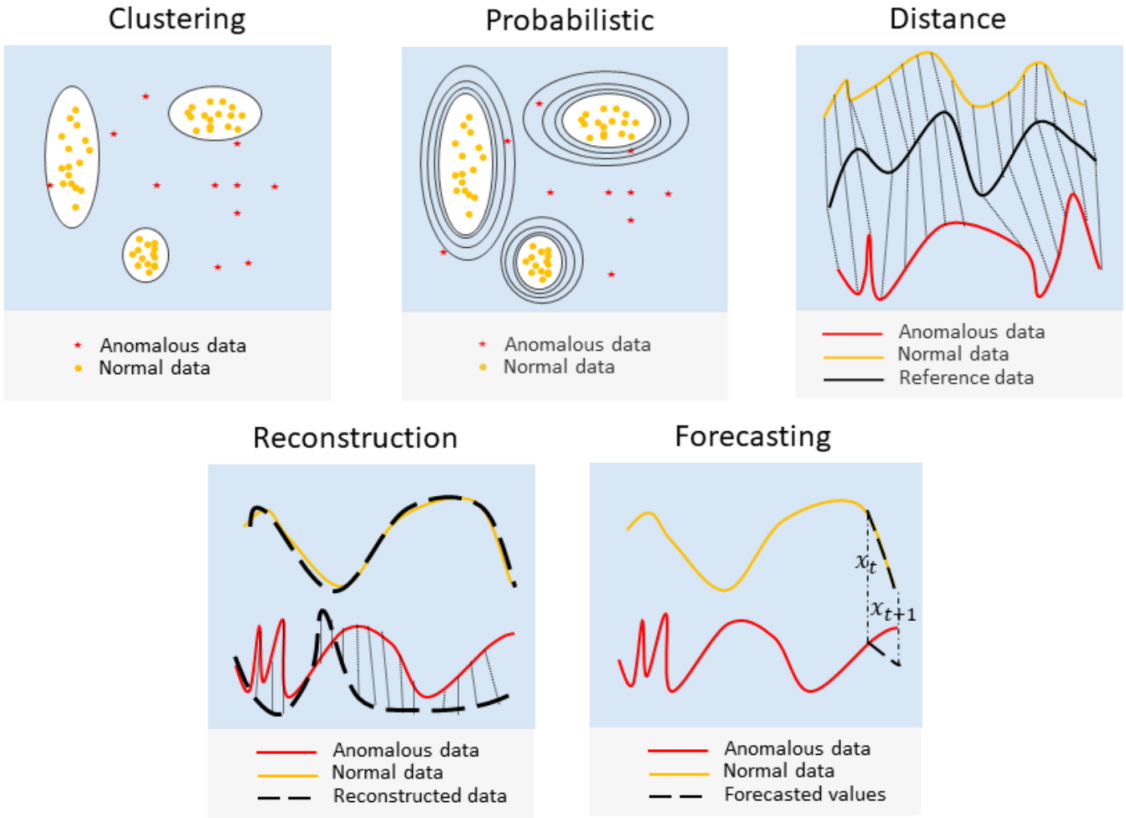


Figure 4.2: Overview of the different paradigms for anomaly detection in time-series: in contrast to clustering and probabilistic approaches, distance-based, reconstruction-based, and forecasting-based approaches take into account the temporal aspect.

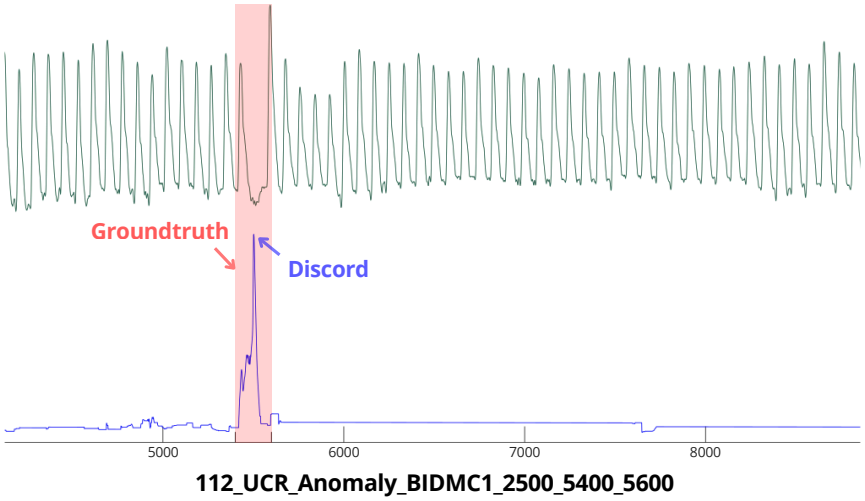


Figure 4.3: An example of time-series from the UCR dataset, where the discord was calculated using DAMP [168].

Clustering-based methods

Let $\mathcal{S}^n$ be the feature space of multivariate time-series of dimension n. Let $\mathcal{N}^n$ be the estimated sub-space of normal time-series of dimension n such that $\mathcal{N}^n \subset \mathcal{S}^n$. Let f be a feature extractor function which maps an input time-series $X \in \mathbb{R}^{n \times N}$ to $\mathcal{S}^n$. An anomaly is detected if,

$$f(X) \notin \mathcal{N}^n. \tag{4.9}$$

Note that the classification of X as an anomaly or not can also be determined with the use of a distance that is compared to a threshold. This is the case, for example, of Support Vector Data Description (SVDD) [69], which measures the distance from the centroids.

Density estimation-based methods

Density estimation-based methods mainly aim to estimate the probability density function of normal time-series denoted as $\mathbf{p}_\theta$. Given a time-series X, the likelihood function $\mathcal{L}$ of θ and a threshold τ, an anomaly is detected if,

$$\mathcal{L}(\theta|X) > \tau. \tag{4.10}$$

Distance-based methods

Distance-based methods rely on the definition of an adequate distance between two temporal sequences. This distance should measure the dissimilarity between them. Let X and R be, respectively, a given time-series and a reference normal time-series. Let us denote by D a distance for time-series. Given a predefined threshold δ, an anomaly is detected in X if,

$$D(X, R) > \delta. \tag{4.11}$$

Reconstruction-based methods

Reconstruction-based approaches aim at learning a model for the accurate and full reconstruction of a normal time-series. The assumption is that the learned model will fail when reconstructing abnormal sequences. Let X and $\hat{X}$ be respectively the original and the reconstructed time-series. Given a predefined threshold δ, an anomaly is detected in X if,

$$\|X - \hat{X}\| > \delta. \tag{4.12}$$

Forecasting-based methods

Forecasting-based approaches are based on the prediction of future states given previous observations. Similar to reconstruction-based methods, they assume that the prediction will be less accurate in the presence of an anomaly. Let $X = \{X_0, X_1, ..., X_N\}$ be a time-series where X_i refers to an observation of X at an instant i. Given a threshold δ, an anomaly is detected at an instant i if,

$$\|\hat{X}_i - X_i\| > \delta, \tag{4.13}$$

where $\hat{X}_{t_i}$ corresponds to the predicted state given the observation $X_{0:i-1} = \{X_0, X_1, ..., X_{i-1}\}$.

4.3 State-of-the-art on time-series anomaly detection

Over the last two decades, the research community has widely explored the field of anomaly detection [169], including anomaly detection in time-series. The latter can be addressed from five different perspectives. As reported in Section 4.2, we distinguish between clustering-based, density-estimation-based, distance-based, reconstruction-based, and forecasting-based techniques. Earlier techniques have investigated these five different paradigms by exploiting traditional machine learning [156, 170] and statistical tools [159]. Nevertheless, as mentioned in [145], these approaches have shown a drop in performance when dealing with high-dimensional time series. Given the recent advances in DL, DNNs have been considered as an alternative [142, 163, 162, 164], mainly taking inspiration from traditional methods. In the following, we review these five categories of approaches, starting with conventional techniques, then moving to current DL methods.

4.3.1 Clustering-based methods

Clustering-based methods are discriminative approaches aiming to estimate explicitly or implicitly decision boundaries for detecting anomalies [68, 170, 69] as depicted in Eq. 4.9. One-Class Support Vector Machine (OC-SVM) [68] is probably one of the most popular algorithms for anomaly detection. Its goal is to estimate the support of a high-dimensional distribution. This one-class classification method has been mainly used for detecting time-independent anomalies [171, 172] but has also been employed for isolating outliers in time-series [173]. Inspired by Support Vector Machines (SVM), Support Vector Data Description (SVDD) [69] is another well-known method that is often used in the context of anomaly detection [174]. Similar to SVM, kernels that map data representations to a higher dimensional space can be used. However, instead of relying on the estimation of a hyperplane, SVDD computes spherically shaped boundaries.

Shallow clustering-based approaches necessitate the hand-crafting of discriminative features and often require the selection of an appropriate kernel. Recently, with the advances in DL, there have been attempts to extend these classical approaches. Most of these meth-

ods, such as Deep SVDD [70] variants, extend traditional methods by learning a kernel that maps data to a discriminative high-dimensional feature space. This is usually carried out by optimizing a Neural Network. These approaches have shown promising results when dealing with non-sequential data. Unfortunately, the temporal modeling of time-series is often disregarded, mainly relying on a simple sliding window. As a solution, [175] suggest fusing multi-scale temporal features and employing a Recurrent Neural Networks (RNN) to model temporal dependencies.

4.3.2 Density-estimation methods

As described in Section 4.2.2, these probabilistic approaches detect anomalies by estimating the normal data density function. For example, [176] proposed a method referred to as Local Outlier Factor (LOF) to detect anomalies by computing the local density. [177] calculate the local connectivity for determining anomalies instead. In [178] and [179], a Gaussian Mixture Model (GMM) and Kernel Density Estimation (KDE) are respectively used for estimating the density of normal representations. Over the last years, efforts have been made to introduce DNN-based probabilistic methods. For instance, [180] proposed to train an auto-encoder for extracting relevant representations before fitting a GMM. Nevertheless, as for clustering-based methods, probabilistic approaches usually do not model the temporal aspect restricting their effectiveness in the context of time-series anomaly detection.

4.3.3 Distance-based methods

Distance-based methods usually define explicitly a distance between a time-series and a reference to detect anomalies [181, 182, 183, 184], as described in Section 4.2.2. Among the most used distance-based algorithms, one can refer to Dynamic Time Warping [185], which aims at finding the optimal match between two ordered sequences. Earlier distance-based methods are mostly characterized by a relatively high complexity induced by the optimal matching and the need for defining a reference time-series [181]. To address those issues, some methods such as [65] reduce the computational cost by only using a small initial snip-

pet instead of a full reference. Alternatively, the DAMP algorithm introduced by Lu et al. [168] can efficiently handle datasets with trillions of data points, by implementing strategies like iterative doubling for backward nearest neighbor search, forward processing for pruning non-discord subsequences and relying on parallel vectors to reduce the computation cost.

4.3.4 Reconstruction-based methods

Reconstruction-based methods aim at reconstructing the entire time-series, as presented in Section 4.2.2. Shallow reconstruction-based time-series anomaly detection methods [186, 187, 188] have mainly adopted Principal Component Analysis [71] (PCA) or its variants such as kernel PCA (kPCA) [189]. These approaches estimate an orthogonal projection, then compute a reconstruction error between the original and reconstructed time-series. Lately, Auto-Encoders (AE) [190] have been introduced as the deep learning-based counterpart of PCA. Unsurprisingly, the latter has been adopted in the context of anomaly detection in time-series [191]. For example, [192] introduce a Long-Short Term Memory Variation Auto-Encoder (LSTM-VAE) architecture. While the Variation Auto-Encoder architecture (VAE) is used for learning robust representations, a Long Short-Term Memory (LSTM) network allows modeling temporal dependencies. Generative Adversarial Networks (GAN) have also been proposed as a reconstruction-based method. In [162], Audibert et al. attempted to take the best of both worlds. In particular, they introduced adversarially trained autoencoders for detecting anomalies in time-series.

4.3.5 Forecasting-based methods

As discussed in Section 4.2.2, traditional forecasting-based anomaly detection methods are primarily based on auto-regression-based models such as AutoRegressive Integrated Moving Average (ARIMA) [193]. With the recent advances in deep learning, LSTM has been used to replace auto-regression models [163]. This architecture allows modeling short-term as well as long-term temporal dependencies. [160] have recently proposed a graph-based deep learning model with an attention mechanism for capturing multivariate correlations.

4.3.6 Hybrid methods

As discussed by Zhao et al. [194], reconstruction and forecasting-based approaches have shown to be, so far, the best candidates for anomaly detection in time-series. While reconstruction-based methods allow modeling inconsistencies within the global distribution of time series, forecasting-based approaches are more appropriate for capturing local anomalies. For that reason, [194] have introduced a hybrid method leveraging these two complementary paradigms. Specifically, they design a two-stream attention-based graph network that simultaneously optimizes forecasting and reconstruction losses.

4.4 Datasets and Evaluation protocol

In this section, the datasets, the evaluation criteria, the pre-processing and post-processing algorithms as well as the considered methods for the experiments are presented.

4.4.1 Datasets

A total of five datasets have been considered for evaluating recent methods for anomaly detection in time-series. Table 4.2 details the different characteristics of each dataset. The considered benchmarks are:

Secure Water Treatment (SWaT). It is a dataset collected from a testbed water treatment for 11 days proposed by Goh et al. in [195]. During the last 4 days, 36 attacks of different duration and natures have been introduced. The data collected over the seven first days have been used for training in all our experiments. During this period, the water treatment was carried out under normal conditions. In contrast, the data gathered during the last 4 days were exposed to multiple attacks. The latter were only considered for testing.

Mars Science Laboratory (MSL). It is formed by 27 telemetry signals collected from the Curiosity Rover spacecraft on Mars. Each signal consists of a multivariate time-series of dimension 55. The first dimension encloses telemetry data, while the remaining 54 correspond to a one-hot encoded command. The publicly available dataset has been released by

NASA [196]. The training and testing data are separated, and anomalies are annotated. Nevertheless, it can be noted that the experimental protocol varies from one reference to another. In particular, some studies such as [197] ignore the one-hot encoded vector considering only telemetry data. In addition, other approaches such as [160] combine the telemetric data from 27 signals assuming that it forms a unique dataset. Nevertheless, in most cases, authors do not provide sufficient information about their experimental protocol, making a direct comparison not straightforward. In this chapter, we follow the experimental protocol of [198]. Each signal is considered to be a separate and independent multivariate sub-dataset. This means that the training and testing phases are performed each time on one single sub-dataset. Finally, the average performance is reported.

Soil Moisture Active Passive dataset (SMAP). This dataset contains telemetry data and one-hot encoded vectors similar to the MSL dataset. It has also been released by NASA [196]. However, in this case, the dataset is formed by 53 signals received from the Soil Moisture Active Passive satellite. The annotated training and testing data are provided. Nevertheless, as for the MSL dataset, similar inconsistencies regarding the experimental protocol can be remarked. For that reason, we propose using the protocol of [198], where each signal is considered to be a separate and independent multivariate sub-dataset. This leads to train and test on 53 different sub-datasets and reporting the obtained average performance.

UCR time series anomaly archive (UCR). It has been recently proposed by Wu and Keogh [133]. In this work, the authors claim that most of the existing anomaly detection datasets are *trivial*. By trivial, they mean that an anomaly can be detected with a single line of MATLAB code. They also criticize the lack of realism and annotation precision in current datasets. As an alternative, they introduce the UCR dataset, which gathers 250 realistic sub-datasets. This dataset is collected from various fields, including medicine, sports, and robotics. The training and test sets are well-defined.

Automated Time-series Outlier Detection System (TODS). It is a collection of 5 synthetically generated multivariate datasets. The dataset was generated using the source code from [146], therefore producing different types of anomalies following the taxonomy of [146]. The dimension of the generated time-series is 10. Training datasets contain only normal

values, while testing datasets incorporate 5 different types of anomalies. The annotation of the outlier types is provided, therefore allowing a per-type analysis.

	SWaT [195]	MSL [198]	SMAP [196]	UCR [133]	TODS [146]
Number of datasets	1	27	55	250	5
Variables	52	55	25	1	10
Percentage of anomalies	12.14	10.48	12.82	0.38	5
Training data points	495000	58317	138004	5302449	10000
Testing data points	449919	73729	435826	12919799	10000
Type of data	Real	Real	Real	Real	Synthetic
Type of anomalies	Artificially forced	Natural	Natural	Natural/Synthetic	Synthetic

Table 4.2: Summary of the five datasets considered in the experiments. The percentage of anomalies in the testing set is reported.

Method	Type of paradigm	Nature
OC-SVM [68]	Clustering	Shallow
iForest [170]	Clustering	Shallow
ARIMA [193]	Forecasting	Shallow
DAMP [168]	Distance-based	Shallow
DA-GMM [180]	Density-estimation	Deep
THOC [175]	Clustering	Deep
USAD [162]	Reconstruction	Deep
GDN [160]	Forecasting	Deep
MTAD-GAT [194]	Hybrid (Forecasting & Reconstruction)	Deep

Table 4.3: Paradigm type and nature of evaluated methods

4.4.2 Evaluation criteria

In this section, we present the used evaluation criteria. **Precision, Recall** and **F1-scores**. The most common metrics used to evaluate the performance of time-series anomaly detection algorithms are the precision computed as follows,

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}, \tag{4.14}$$

the recall is calculated as below,

$$\text{Recall} = \frac{\text{True positives}}{\text{True Positives} + \text{False Negatives}} \tag{4.15}$$

and the F1-score corresponding to,

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{4.16}$$

It is worth noting that some methods, such as the work of Wu and Keogh [133], generally opt for other evaluation metrics. For instance, in [133] the authors argue that time-series datasets should only have a single anomalous sequence per series. Under such a setting they propose using accuracy to assess whether an anomaly has been correctly identified. Such a binary score is often simple to interpret and to use and can be used for introducing more flexibility. On the other hand, standard methods provide a more comprehensive and explainable assessment of performance across multiple instances, namely true positives, false positives, and false negatives. They can handle more than a single anomaly subsequence in a given time-series, and are therefore adapted to data with different anomaly ratios such as most of the considered benchmarks (SWaT, MSL, SMAP).

Revisited precision, recall and F1-scores for time-series. In addition to conventional performance metrics, more recent and elaborate performance metrics tailored to time-series introduced by Tatbul et al. [144] are considered. These metrics extend classical precision, recall, and F1-score from point-based to range-based anomaly detection. Figure 4.4 highlights the distinction between point-based and range-based anomalies. Contrary to the case of point-based approaches, a prediction in a time-series can be both a true positive (TP) and a false negative (FN) due to *partial overlap* with the ground truth as shown in Figure 4.4 b. Therefore, as discussed by Tatbul et al. [144], a more informative time-series evaluation process should (1) quantify the size of the partial overlap; (2) identify the overlap position, and; (3) take into account its cardinality, i.e., with how many anomalous ground truth sub-sequences it overlaps. More specifically, given a set of real anomaly sequences

$R = \{R_1, ..., R_{Nr}\}$ and a set of predicted anomaly sequences $P = \{P_1, ..., P_{N_p}\}$ the recall is expressed with respect to the number of real anomalies N_r in a dataset [144]. It seeks to reward a detector when it predicts a TP and penalizes it when the prediction is an FN as follows,

$$\text{Recall}_T(R, P) = \frac{1}{N_r} \sum_{i=1}^{N_r} \text{Recall}_T(R_i, P), \tag{4.17}$$

and,

$$\text{Recall}_T(R_i, P) = \alpha \cdot \mathbb{1}_{\sum_{j=1}^{N_p} |R_i \cap P_j| \geq 1} + \frac{1-\alpha}{\sum_{j=1}^{N_p} |R_i \cap P_j|} \cdot \mathcal{S}_c(R_i, P), \tag{4.18}$$

where $0 \leq \alpha \leq 1$ is a scaling factor that rewards the detector when it detects the existence of the anomaly R_i and $\mathbb{1}$ is an indicator function. Finally, $\mathcal{S}_c(R_i, P)$ which quantifies the overlap size is computed based on the cumulative overlap size ω as follows,

$$\mathcal{S}_c(R_i, P) = \sum_{j=1}^{N_p} \omega(R_i, R_i \cap P_j, \delta), \tag{4.19}$$

where δ returns a score depending on the overlap location between R_i and a prediction P_j (flat bias, front bias, middle bias, and back bias). Further details could be found in the original manuscript of [144]. The precision is similarly defined. It seeks to assess the quality of the predictions by rewarding a detector in the presence of a TP and penalizing it when facing an FP. It is computed as follows,

$$\text{Precision}_T = \frac{1}{N_p} \sum_{i=1}^{N_p} \text{Precision}_T(R, P_i), \tag{4.20}$$

and,

$$\text{Precision}_T(R, P_i) = \frac{1}{\sum_{j=1}^{N_r} |R_j \cap P_i|} \cdot \mathcal{S}(R, P_i), \tag{4.21}$$

where $\mathcal{S}_c(R, P_i)$ quantifies the cumulative overlap between the considered prediction P_i and all the ground truths in R as explained in Equation 4.19. It is expressed as,

$$\mathcal{S}_c(R, P_i) = \sum_{j=1}^{N_r} \omega(P_i, P_i \cap R_j, \delta). \tag{4.22}$$

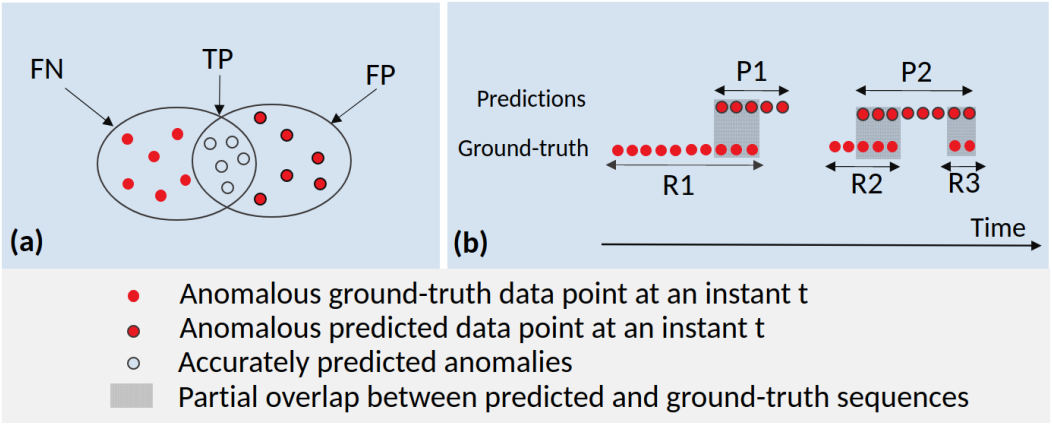


Figure 4.4: The evaluation process of (a) point-based anomalies versus (b) range-based anomalies. Range-based anomalies are characterized by partial overlap(s) with the ground truth. A more accurate evaluation for time-series should quantify the overlap in terms of **size**, **position**, and **cardinality**.

Finally, the F1-score is redefined as follows,

$$\text{F1-score}_T = \frac{2 \cdot \text{Precision}_T \cdot \text{Recall}_T}{\text{Precision}_T + \text{Recall}_T} \tag{4.23}$$

Model stability. We define model stability as the ability of a machine/deep learning algorithm to reproduce similar results when retrained under the same conditions. While OC-SVM ensures stability because of its deterministic nature, most of the considered methods rely on a random parameter initialization which may impact the final performance of the model. Ideally, the model should achieve the same results regardless of this initialization. To assess the stability, each experiment is carried out five times. Then, the mean and standard deviation of those five runs are reported. A lower standard deviation reflects higher stability. To the best of our knowledge, we are among the first to analyze this aspect experimentally in the context of anomaly detection in time-series.

Generalization to different types of anomalies. We propose reporting the performance according to the anomaly type encountered. This analysis can help identify the most suitable algorithm for a given application. To that aim, the TODS benchmark, which encloses the

annotation of 5 different types of outliers depicted in Section 4.1 is used. Although the definition of different anomaly types has been reported in several references, very few works have carried out an experimental study with respect to the anomaly type. A rare example we can mention is the work of [146]. Nevertheless, it can be noted that in this chapter, recent DL-based state-of-the-art approaches such as GDN [160], USAD [162], and MTAD-GAT [194] are not evaluated. For each anomaly type, the percentage of well-detected anomalies is reported.

Model size. In a real-world context, deploying algorithms on specific hardware with a limited memory capacity can be challenging. Therefore, being aware of the model size, which directly impacts the memory consumption, is a crucial component often neglected. For that purpose, we report the number of parameters and the size in MegaBytes (MB) of the trained deep learning models considered for this evaluation.

4.4.3 Post-processing and Pre-processing

Data Normalization. Normalizing the data is a common practice in machine learning, particularly in anomaly detection. Hence, for the sake of fairness, a data normalization pre-processing was applied in all our experiments. More specifically, the data are normalized using the maximum and minimum values in the training data as in [194].

Point Adjustment. Point adjustment initially introduced by Xu et al. [161] is a protocol that adjusts the predictions before computing performance metrics. It acts as follows: if at least one point is classified as an anomaly in an outlier segment, all the predictions in that segment are set to anomalous. The idea behind this protocol is that an algorithm triggering an alert for any point in a contiguous anomaly segment might be sufficient for a timely reaction. Figure 4.5 illustrates the point adjustment protocol by showing the ground truth, the original predictions, and the predictions after point adjustment of a given time-series. After applying the point adjustment protocol, the F1-Score goes from 0.32 to 0.85. This significant gap has therefore raised some concerns in the literature regarding the use of point adjustment. For example, [147] claim that by using this protocol, a randomly generated anomaly score might outperform several recently proposed time-series anomaly detection algorithms. In

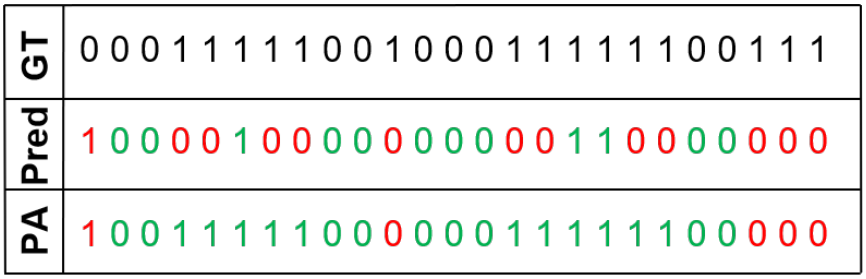


Figure 4.5: Application of the Point Adjustment (PA) on a given time-series: the Ground Truth (GT), the original prediction (Pred) and the prediction after Point Adjustment (PA) are reported. In this example, the performance of the algorithm without and with point adjustment is, respectively: Precision $= 0.75$, Recall $= 0.2$, F1-Score $= 0.32$, and Precision $= 0.92$, Recall $= 0.79$, F1-Score $= 0.85$. Best viewed in colors.

this chapter, we report the performance of existing methods with and without point adjustment.

4.4.4 Evaluated methods

We consider in total nine anomaly detection methods. Table 4.3 summarizes the characteristics of each evaluated method.

Four shallow standard methods are evaluated, namely, OC-SVM [68], iForest [170], ARIMA [193] and DAMP [168]. In addition, five recent DL-based methods have been considered: DA-GMM [180], THOC [175] USAD [162], GDN [160] and MTAD-GAT [194]. The latter has been selected according to the following criteria: (1) Relevance of the topic: all the chosen anomaly detection algorithms are unsupervised and have been specifically designed for detecting anomalies in time-series. (2) Publication date: all the DL-based algorithms are recent. In particular, they have been introduced between 2018 and early 2022. (3) Impact: the chosen algorithms have been published in top-tier conferences and are highly cited papers from the field. (4) Code availability: the official codes of the selected algorithms are publicly available. (5) Diversity: methods from different paradigms have been considered. The only paradigm that was ignored is the distance-based since we were not able to find a deep learning approach falling in this category.

Method		SWaT	MSL	SMAP	UCR	TODS	Avg. F1
USAD	P	0.28 ± 0.02	0.15 ± 0.01	0.18 ± 0.01	0.01 ± 0.00	0.05 ± 0.00	
	R	0.74 ± 0.01	0.57 ± 0.05	0.49 ± 0.01	0.48 ± 0.00	0.54 ± 0.02	
	F1	0.41± 0.02	0.21± 0.02	0.21± 0.01	0.02± 0.00	0.10± 0.00	0.19± 0.13
GDN	P	0.34± 0.03	0.31± 0.01	0.25± 0.00	0.12± 0.00	0.07± 0.01	
	R	0.72± 0.04	0.64± 0.02	0.55± 0.04	0.42± 0.00	0.59± 0.16	
	F1	0.46± 0.03	0.35± 0.01	0.33± 0.01	0.12± 0.00	0.11± 0.00	0.27± 0.14
THOC	P	0.62± 0.16	0.22± 0.01	0.16± 0.01	0.01± 0.01	0.05± 0.01	
	R	0.46± 0.13	0.46± 0.02	0.27± 0.01	0.00± 0.01	0.19± 0.03	
	F1	0.52± 0.14	0.25± 0.01	0.12± 0.01	0.00± 0.00	0.08± 0.14	0.19± 0.18
MTAD-GAT	P	0.85± 0.04	0.57± 0.04	0.58± 0.03	0.10± 0.00	0.16± 0.08	
	R	0.90± 0.03	0.79± 0.03	0.87± 0.03	0.28± 0.01	0.01± 0.02	
	F1	**0.87± 0.01**	**0.60± 0.03**	**0.65± 0.03**	**0.13± 0.01**	0.02± 0.03	0.45± 0.32
DAGMM	P	0.43± 0.00	0.12± 0.01	0.11± 0.01	0.01± 0.00	0.12± 0.00	
	R	0.71± 0.00	0.19± 0.00	0.17± 0.01	0.20± 0.00	0.49± 0.00	
	F1	0.54± 0.00	0.12± 0.00	0.10± 0.01	0.01± 0.00	**0.19± 0.00**	0.19± 0.18
OC-SVM	P	0.24± 0.00	0.15± 0.00	0.12± 0.00	0.01± 0.00	0.05± 0.00	
	R	0.85± 0.00	0.66± 0.00	0.66± 0.00	0.73± 0.00	0.85± 0.00	
	F1	0.37± 0.00	0.24± 0.00	0.20± 0.00	0.02± 0.00	0.09± 0.00	0.18± 0.12
iForest	P	0.23± 0.10	0.18± 0.04	0.10± 0.01	0.05± 0.01	0.05± 0.01	
	R	0.83± 0.10	0.16± 0.05	0.04± 0.01	0.12± 0.01	0.04± 0.01	
	F1	0.36± 0.10	0.17± 0.03	0.08± 0.00	0.07± 0.00	0.04± 0.01	0.14± 0.12
ARIMA	P	0.13± 0.00	0.28± 0.00	0.17± 0.00	0.01± 0.00	0.05± 0.00	
	R	0.99± 0.00	0.83± 0.00	0.82± 0.00	0.85± 0.00	0.69± 0.00	
	F1	0.23± 0.00	0.28± 0.00	0.19± 0.00	0.02± 0.00	0.09± 0.00	0.16± 0.09
DAMP	P	–	–	–	0.33± 0.00	–	
	R	–	–	–	0.34± 0.00	–	
	F1	–	–	–	**0.28**± 0.00	–	–

Table 4.4: Results in terms of traditional performance metrics of evaluated state-of-the-art methods (precision P, recall R, F1-score) on the 5 considered datasets **without Point Adjustment (PA)**. The experiments have been performed 5 times for each algorithm and dataset. The mean and standard deviation are reported. The bold and underlined results correspond to the first and second-best F1-Score, respectively.

Method		SWaT	MSL	SMAP	UCR	TODS	Avg. F1
USAD	P	0.32± 0.02	0.22± 0.02	0.26± 0.01	0.02± 0.00	0.07± 0.00	
	R	0.89± 0.03	0.99± 0.03	0.95± 0.01	0.95± 0.00	0.71± 0.02	
	F1	0.47± 0.02	0.33± 0.01	0.34± 0.02	0.04± 0.00	0.13± 0.00	0.26± 0.16
GDN	P	0.40± 0.05	0.39± 0.02	0.36± 0.01	0.31± 0.01	0.10± 0.02	
	R	0.72± 0.04	1.00± 0.00	1.00± 0.00	0.99± 0.00	0.75± 0.12	
	F1	0.57± 0.05	0.50± 0.02	0.46± 0.01	**0.39± 0.01**	0.16± 0.02	0.42± 0.14
THOC	P	0.77± 0.08	0.31± 0.01	0.26± 0.01	0.06± 0.02	0.09± 0.01	
	R	0.86± 0.02	0.87± 0.02	0.84± 0.02	0.06± 0.02	0.35± 0.06	
	F1	0.81± 0.05	0.41± 0.02	0.34± 0.01	0.06± 0.02	0.14± 0.02	0.35± 0.26
MTAD-GAT	P	0.86± 0.04	0.60± 0.04	0.59± 0.03	0.17± 0.00	0.16± 0.08	
	R	0.96± 0.03	0.86± 0.03	0.91± 0.03	0.57± 0.02	0.01± 0.02	
	F1	**0.90± 0.01**	**0.64± 0.04**	**0.67± 0.03**	0.25± 0.01	0.02± 0.03	0.50± 0.32
DAGMM	P	0.49± 0.00	0.20± 0.00	0.16± 0.00	0.03± 0.00	0.15± 0.00	
	R	0.90± 0.00	0.44± 0.00	0.41± 0.00	0.78± 0.00	0.63± 0.00	
	F1	0.64± 0.00	0.25±0.00	0.19±0.00	0.06±0.00	**0.24±0.00**	0.28± 0.19
OCSVM	P	0.26± 0.00	0.25± 0.00	0.15± 0.00	0.02± 0.00	0.05± 0.00	
	R	0.95± 0.00	0.95± 0.00	0.85± 0.00	0.85± 0.00	0.85± 0.00	
	F1	0.41± 0.00	0.40± 0.00	0.26± 0.00	0.04± 0.00	0.09± 0.00	0.24± 0.15
iForest	P	0.26± 0.12	0.47± 0.04	0.10± 0.01	0.16± 0.01	0.17± 0.06	
	R	0.97± 0.00	0.66± 0.06	0.04± 0.01	0.45± 0.01	0.17± 0.02	
	F1	0.40± 0.13	0.55± 0.04	0.36± 0.01	0.24± 0.01	0.17± 0.02	0.34± 0.13
ARIMA	P	0.13± 0.00	0.31± 0.00	0.18± 0.00	0.01± 0.00	0.06± 0.00	
	R	1.00± 0.00	1.00± 0.00	0.96± 0.00	0.97± 0.00	0.87± 0.00	
	F1	0.23± 0.00	0.39± 0.00	0.24± 0.00	0.02± 0.00	0.11± 0.00	0.20± 0.13
DAMP	P	–	–	–	0.35± 0.41	–	
	R	–	–	–	0.51± 0.50	–	
	F1	–	–	–	**0.40**± 0.43	–	–

Table 4.5: Results in terms of traditional performance metrics of evaluated state-of-the-art methods (precision (P), recall (R), F1-score (F1)) on the 5 considered datasets **with Point Adjustment (PA)**. The experiments have been performed 5 times for each algorithm and dataset. The mean and standard deviation are reported. The bold and underlined results correspond to the first and second-best F1-Score, respectively.

4.5 Results

4.5.1 Performance using standard metrics

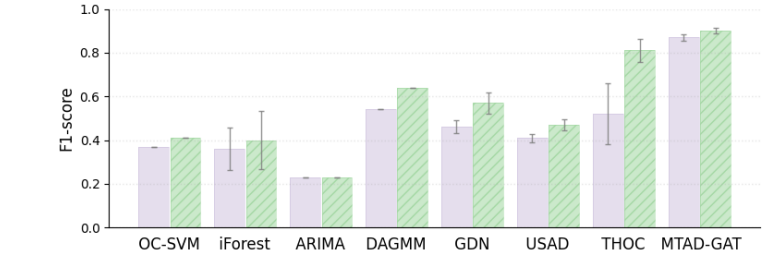
Best performing approach. Table 4.4 reports the performance of the evaluated methods using standard metrics (precision, recall, and F1-score) on the five considered datasets. The performance is first averaged over the data of every dataset, which in turn is averaged over 5 runs. For a more intuitive visualization of the results, Figure 4.6 shows the F1-score with and without Point Adjustment (PA). In general, MTAD-GAT [194] is the best-performing approach, surpassing other methods on three datasets, namely, SWaT, MSL, SMAP. This can be explained by the fact that this approach is hybrid as it is based on forecasting and reconstruction losses. Indeed, this allows the simultaneous detection of local and global anomalies. Nevertheless, it can be remarked that the results obtained on TODS contradict this statement. Indeed, MTAD-GAT registers inferior performance on this benchmark as compared to other methods, including DL and conventional methods. Two hypotheses might justify this drop: (i) the TODS dataset encloses complex anomalies that are *moderately local* and are hardly captured by a simple forecasting and/or reconstruction approach, favoring probabilistic modeling as in DAGMM [180]. However, the higher performance obtained for GDN [160] and USAD [162] partly disprove this assumption; and (ii) the synthetic data in TODS are not realistic, making them hardly predictable. Another observation that can be made is that GDN [160] presents the second-best performance on two datasets, namely MSL, SMAP. This confirms the relevance of using graph representations for modeling time-series. Surprisingly, graph-based approaches (GDN and MTAD-GAT) remain relatively effective on a univariate dataset (UCR), although modeling the connectivity between variables is unnecessary.

DL vs conventional methods. As reported in Table 4.4 and Figure 4.6, DL methods, specifically MTAD-GAT and GDN, generally outperform conventional methods. For example, the superiority of DL approaches is extremely noticeable when comparing GDN and ARIMA, which are both forecasting techniques. This increase in performance can be explained by the fact that ARIMA struggles to model the dependencies between variables. However, the

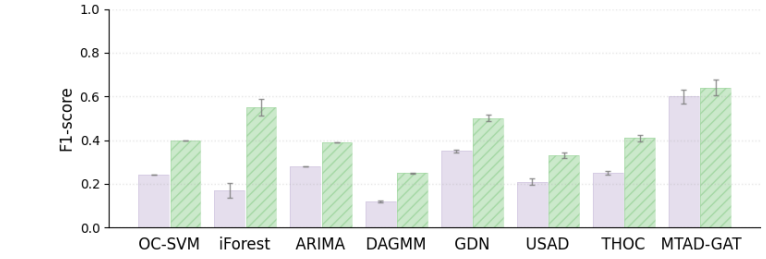
gap in performance between DL and traditional methods is less visible in some cases, supporting the assumption of [133] and contradicting [145], which argues that DL methods are more effective in the presence of high-dimensional time-series. For instance, DAMP [168]) beats all DL methods by a large margin showing an average F1-score of 0.28 against only 0.13 for the best-performing DL-techniques, OC-SVM shows comparable performance with several DL-based anomaly methods such as THOC and USAD on high-dimensional datasets. More precisely, OC-SVM achieves an F1-score of 0.24 and 0.2 on MSL and MSAP against 0.21 and 0.21 for USAD and 0.25 and 0.2 for THOC, respectively. Another observation that can be made is that conventional approaches, except iForest, seem to be suitable for applications where recall is more important than precision. An example of such an application could be the detection of debris among other objects in space [199]. On the contrary, DL approaches are overall more precise.

Impact of point adjustment. From the results of Table 4.5, and Figure 4.6, it can be noted that the Point Adjustment (PA) process significantly boosts the performance. In particular, the highest performance gain can be observed for iForest on the MSL dataset, where the F1-score increases from 17% to 55%. This can be explained by the fact that PA adjusts the predictions before computing the metrics. The adjustment is made in a way that rewards a detector when detecting at least one instance of an anomalous segment. The intuition behind that is that finding one anomaly in a segment is sufficient for a timely reaction. Such an intuition closely impacts the recall since it increases the number of False Positives (FP). However, as discussed in [147], using PA can induce a misleading ranking of the performance. This is confirmed in Table 4.4 where the results obtained for USAD, THOC and DAGMM are comparable and contradict the performance metrics reported in Table 4.5. In addition, before applying PA, all DL approaches seem to be in general, more effective than conventional approaches. However, after PA, this is no longer the case. For example, iForest achieves comparable performance with THOC. Overall, PA seems to bias the analysis as it treats range-based data as punctual, neglecting the overlap size and the location of anomalies. Consequently, this blurs the applicability of detectors in real-life setups.

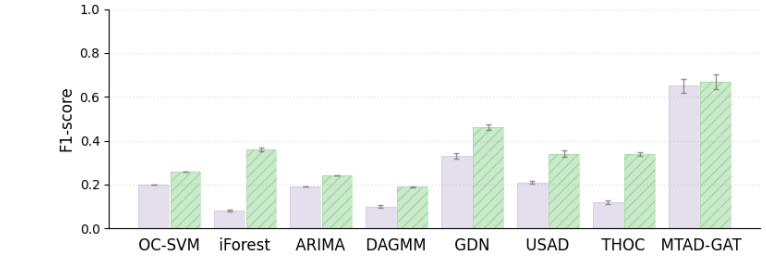
Benchmark complexity. All tested methods fail to detect effectively anomalies in UCR,



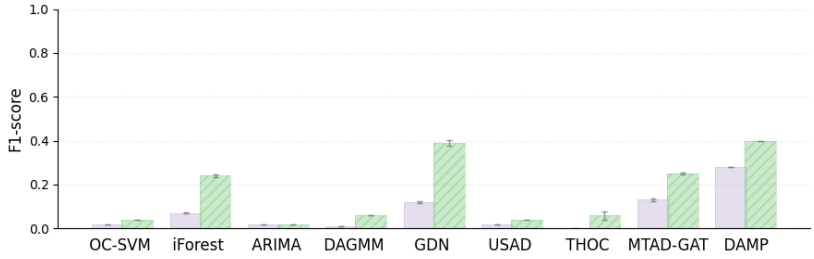
(a) SWaT



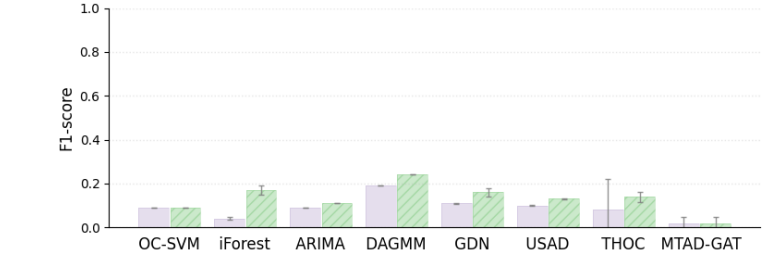
(b) MSL



(c) SMAP



(d) UCR



(e) TODS

Figure 4.6: Average F1-Score on the five datasets comparing the non-PA and PA protocols. The non-hatched and hatched bars correspond to the mean F1-Score with and without Point Adjustment (PA), respectively. The vertical black line represents the standard deviation over five runs.

although it is a univariate dataset. This might be due to its low ratio of anomalies. As discussed in Section 4.1, UCR is among the first benchmarks to mimic a more realistic configuration, highlighting the difficulty of detecting rare anomalies. The rate of anomalies might have a significant role in defining the complexity of a given dataset. For natural anomalies, two observations could be made. First, the average performance on MSL and SMAP is comparable despite having a significantly different number of variables. Second, natural but induced/forced anomalies seem easy to detect, given that all methods perform well on the SWaT dataset. Unfortunately, such a scenario is unrealistic in most real-world settings as the anomalies are generally infrequent. This point was also raised by[133], highlighting that several benchmark datasets have unrealistic anomaly densities.

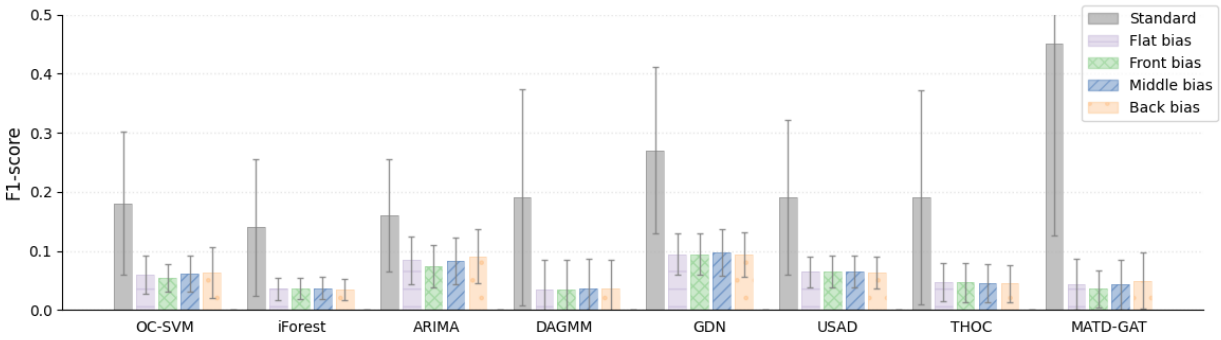


Figure 4.7: The mean performance per method on all datasets using the range-based metrics of [144], with different location biases.

4.5.2 Performance using revisited metrics for time-series

Conventional metrics vs revisited metrics. Table 4.7 shows the results of evaluation using the revisited F1-score for time-series calculated using Recall_T and Precision_T proposed by Tatbul et al. [144].

It can be remarked that there exists a significant gap in performance between the results based on conventional and revisited metrics. One main reason is that the revisited metrics consider the overlap size between the predicted sequences and the ground truth. In contrast, the traditional metrics do not take into account the sequential aspect nor quantify the overlap

between the predictions and the ground truth. Moreover, it can be noted that the results of the revisited metrics are not in full accordance with the conventional ones except for DAMP on UCR. On the one hand, GDN and USAD achieve more competitive results as compared to other approaches. On the other hand, the assumption about the superiority of the hybrid method no longer holds. Overall, both classical and DL forecasting-based techniques give the highest performance. This perspective suggests that the majority of anomalies in benchmarked datasets are local. Finally, DAGMM seems to be among the least effective methods, suggesting its inability to model the distribution of anomalies. This can be explained by the fact that anomalies do not necessarily follow a multimodal Gaussian distribution. Some observations can be made regarding the difficulty of each dataset. First, MSL seems slightly less challenging than SWaT. Second, in line with the results based on conventional metrics, the obtained performance suggests that natural anomalies are more straightforward to detect than synthetic ones. Two reasons could potentially explain that: (1) the datasets with natural anomalies have a high percentage of anomalies, and (2) synthetic datasets do not reliably reflect reality and do not include a sufficient number of anomalies.

Location bias. Herein, we analyze the results for different location biases. Table 4.7, Table 4.8, Table 4.9, and Table 4.10 show the results using flat, front, middle and back bias, respectively. As mentioned in Section 4.4.2, taking into account the size, the cardinality and the location of the overlap between a predicted sequence and its corresponding ground truth is crucial. Therefore, the location bias weights every predicted time-stamp given its location in the sequence.

Figure 4.7 depicts the overall performance for each method under different bias settings. Two observations can be made: (i) Although the idea of location bias seems theoretically interesting and flexible for different domain-specific applications, it does not practically bring more information in our experiments, as the average F1-score does not change importantly. However, the most notable results are registered for the middle and back biases as compared to the flat and front biases. This can be attributed to the uneven distribution of anomalies in datasets like SMAP and MSL. As noted by Wu and Keogh [133], most anomalies in these datasets occur towards the end of the sequences. This may explain the improved

performance of all the evaluated methods on MSL and SMAP, particularly for middle and back location biases. This also suggests that most detectors are less mature for applications necessitating an early anomaly detection such as real-time intrusion detection [200], cyberattack attempts via network activity [201] or cancer detection [137]. (ii) Additionally, all range-based metrics results are less impressive than the ones obtained using standard metrics. This drop in performance may suggest that most approaches perform poorly in identifying the overlap size and cardinality between a predicted sequence and its corresponding ground truth. In other words, a predicted sequence does not perfectly align with its corresponding ground truth sequence, as the boundaries of anomaly sequences are not well predicted.

DL vs conventional methods. Although the top-three best-performing methods are DL models (according to the conventional metrics), it can be seen that classical approaches such as DAMP can outperform DL-methods by a large margin, with advantage of a stable model. Similarly, OC-SVM can achieve comparable performance with its counterpart clustering DL approach, namely THOC. This suggests that conventional methods are not obsolete and that, depending on the application, they can be considered for anomaly detection [133].

Univariate vs multivariate. The results of Table 4.4 and Table 4.7 seem to be in accordance. Indeed, all methods except DAMP [168] seem to have poor performance on UCR, which is univariate, while on other multivariate datasets such as MSL, the performance is relatively higher.

4.5.3 Model stability

Besides reporting the precision, recall, and F1-score, it is interesting to observe the behavior of detectors when trained with different initializations. Table 4.4 and Figure 4.6 report the performance average and standard deviation for every approach after five runs. Undoubtedly the most stable methods are the deterministic ones which are ARIMA, OC-SVM, and DAMP. Among DL approaches, DAGMM seems to be the most stable. This could be explained by the fact that it is a density-based approach that relies on estimating the density of normal data. In contrast, THOC and iForest achieve less stable results, especially on SWaT.

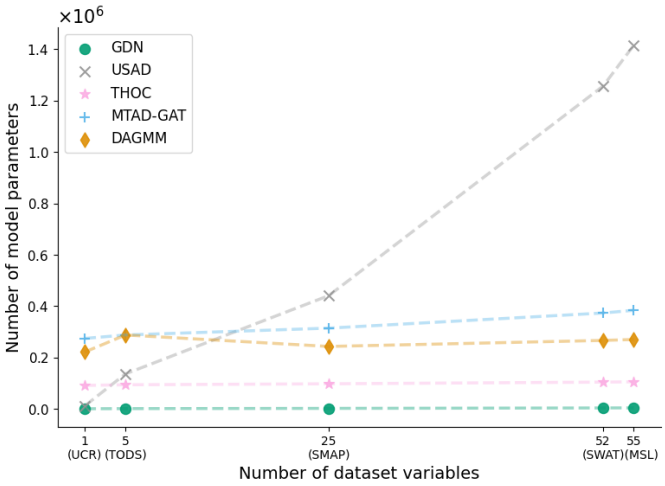


Figure 4.8: Relation between the number of the parameters of the model and the number of features in the considered dataset.

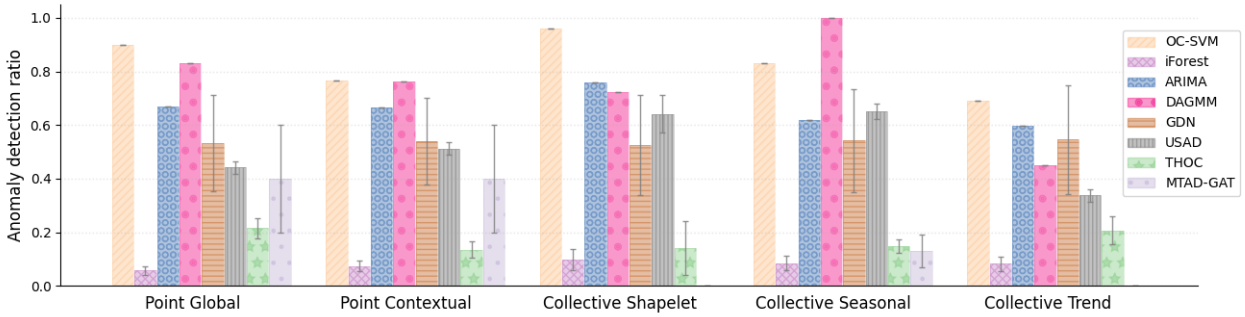


Figure 4.9: The ratio of true anomalies detected for each tested method when varying the anomaly types. All methods succeeded in partially detecting each anomaly type, except MTAD-GAT which was unable to detect any collective trend anomaly.

4.5.4 Model size and memory consumption

Tables 4.6 (a) and (b) depict the number of parameters and the model sizes in MegaBytes (MB) of the tested DL methods, respectively. GDN seems to have the lowest number of parameters when tested on all the datasets. In fact, the number of parameters in USAD is around 300 times higher than GDN. This is explained by the fact that the architecture of USAD is complex and is composed of two adversarially trained auto-encoders. In Figure 4.8, it can also be seen that contrary to other models, which vary almost linearly, the number of parameters increases at a considerably higher rate. Additionally, despite the significant difference in parameter number, GDN still achieves better results than USAD. This highlights the relevance of using graph representations not only for modeling time-series but also for building less complex model architectures.

	Number of Parameters				
Method	**SWaT**	**MSL**	**SMAP**	**UCR**	**TODS**
USAD	1.256.871	1.414.755	441.225	12.321	136.710
GDN	4.225	4.481	2.561	1.025	1.601
THOC	104.768	105.792	98.112	91.968	94.272
MTAD-GAT	373.637	384.145	314.695	274.687	288.070
DAGMM	266.930	270.542	243.452	221.780	288.070

(a) Per dataset number of model parameters

	Model Size (MB)				
Method	**SWaT**	**MSL**	**SMAP**	**UCR**	**TODS**
USAD	4,79	5,40	1,68	0,05	0,54
GDN	0,02	0,02	0,01	0,01	0,01
THOC	0,41	0,42	0,39	0,36	0,37
MTAD-GAT	1,62	1,66	1,39	1,05	1,05
DAGMM	1,50	1,50	1,40	1,30	1,40

(b) Per dataset model size (MB)

Table 4.6: Number of parameters and model size (MB) of the trained models on different datasets

4.5.5 Generalization to different types of anomalies

Figure 4.9 shows the percentage of detected anomalies per type for all the tested methods. In general, it can be noted that for the majority of tested techniques, collective trend anomalies are probably the most challenging to detect. ARIMA and GDN, which are predictive approaches, show the best generalization capacity to different types of anomalies. OC-SVM easily detects global point and collective shapelet anomalies but still presents decent results for other anomaly types. The results obtained for USAD suggest that it is more robust to collective outliers (e.g., collective shapelet and seasonality), which can be explained by

Dataset		USAD	GDN	THOC	MTAD-GAT	DAGMM	OC-SVM	iForest	ARIMA	DAMP
SWaT	P_T	5.83 ± 2.10	7.03 ± 1.73	29.05 ± 11.13	3.31 ± 0.82	10.32 ± 0.22	3.12 ± 0.00	0.38 ± 0.14	3.95 ± 0.00	–
	R_T	32.84 ± 3.96	25.05 ± 2.90	2.04 ± 1.08	39.46 ± 3.85	18.89 ± 0.22	85.66 ± 0.00	90.21 ± 0.14	80.85 ± 0.00	–
	F1$_T$	9.76 ± 2.96	10.81 ± 1.88	3.81 ± 1.96	6.07 ± 1.33	13.34 ± 0.26	6.02 ± 0.00	0.77 ± 0.29	7.53 ± 0.00	–
MSL	P_T	12.10 ± 0.79	24.94 ± 1.74	15.92 ± 1.43	7.68 ± 0.45	11.09 ± 0.25	5.28 ± 0.00	3.89 ± 1.23	8.43 ± 0.00	–
	R_T	20.14 ± 2.63	22.93 ± 2.17	22.34 ± 2.13	18.46 ± 0.45	3.79 ± 0.25	83.16 ± 0.00	58.46 ± 1.23	41.06 ± 0.00	–
	F1$_T$	6.66 ± 0.30	15.35 ± 1.33	7.70 ± 0.44	3.19 ± 0.23	1.61 ± 0.04	8.56 ± 0.00	6.25 ± 1.38	9.37 ± 0.00	–
SMAP	P_T	12.04 ± 0.95	15.00 ± 0.32	8.14 ± 0.47	14.01 ± 0.00	07.09 ± 0.94	06.56 ± 0.00	01.78 ± 0.64	14.01 ± 0.00	–
	R_T	25.01 ± 0.70	20.23 ± 1.60	8.07 ± 0.81	52.72 ± 0.00	3.79 ± 0.94	83.65 ± 0.00	37.28 ± 0.64	52.72 ± 0.00	–
	F1$_T$	6.34 ± 0.32	7.93 ± 0.33	2.59 ± 0.18	11.79 ± 0.00	0.91 ± 0.09	10.40 ± 0.00	2.85 ± 0.64	11.79 ± 0.00	–
UCR	P_T	1.06 ± 0.02	7.49 ± 0.32	1.06 ± 0.40	1.13 ± 0.17	7.09 ± 0.94	0.83 ± 0.00	2.55 ± 0.34	0.74 ± 0.00	32.49 ± 0.00
	R_T	23.20 ± 0.23	25.01 ± 1.18	00.11 ± 0.40	01.89 ± 0.17	03.79 ± 0.94	83.56 ± 0.00	40.89 ± 0.34	29.07 ± 0.00	34.20 ± 0.00
	F1$_T$	1.68 ± 0.03	4.73 ± 0.13	0.16 ± 0.05	0.54 ± 0.05	0.91 ± 0.09	1.34 ± 0.00	3.52 ± 0.31	1.19 ± 0.00	**28.47** ± 0.00
TODS	P_T	4.20 ± 0.39	5.37 ± 1.30	6.17 ± 0.47	8.06 ± 9.70	7.09 ± 0.94	1.97 ± 0.00	9.75 ± 8.62	6.85 ± 0.00	–
	R_T	47.96 ± 0.15	55.46 ± 17.13	17.66 ± 0.47	0.19 ± 9.70	03.79 ± 0.94	80.54 ± 0.00	6.45 ± 8.62	63.95 ± 0.00	–
	F1$_T$	7.67 ± 0.65	8.43 ± 1.47	8.87 ± 0.73	0.27 ± 0.35	0.91 ± 0.09	3.82 ± 0.00	4.35 ± 0.41	12.33 ± 0.00	–
Avg. F1$_T$		6.42 ± 2.66	**9.45** ± 3.53	4.63 ± 3.23	4.37 ± 4.26	3.54 ± 4.91	6.03 ± 3.24	3.55 ± 1.80	<u>8.44</u> ± 4.02	–

Table 4.7: The *flat-bias* performance (in %) of the tested methods on the five benchmarks using the metrics proposed by Tatbul et al. [144]. The average and the standard deviation of five runs are reported.

the fact that it is a reconstruction approach that can essentially capture global inconsistencies. DAGMM effectively detects collective seasonal anomalies but shows less impressive results for collective shape outliers. Again, this might return to the probabilistic nature of DAGMM, which is coupled with a sliding window. Finally, MTAD-GAT fails in detecting collective anomalies, despite being hybrid.

4.5.6 Discussion

In the following, we summarize the main findings of the present evaluation study:
(i) It is generally difficult to vote for a best-performing approach or paradigm and the performance of an approach highly depends on the considered use case and the nature of the encountered anomalies. For instance, although the hybrid approach MTAD-GAT seems to outperform most other methods, they also exhibit limitations, such as their unsuitability for detecting collective shapelet and trend anomalies. This highlights the need for researchers to explicitly discuss the specific settings or applications under which their algorithms are effective, ensuring that practitioners understand the circumstances in which these methods

Dataset		USAD	GDN	THOC	MTAD-GAT	DAGMM	OC-SVM	iForest	ARIMA	DAMP
SWaT	P_T	5.90 ± 2.13	7.05 ± 1.73	29.05 ± 11.14	3.12 ± 0.76	10.46 ± 0.23	3.31 ± 0.00	0.39 ± 0.14	4.04 ± 0.00	–
	R_T	32.01 ± 3.48	25.04 ± 2.96	2.21 ± 1.13	46.32 ± 4.80	18.16 ± 0.23	85.71 ± 0.00	90.28 ± 0.14	81.19 ± 0.00	–
	F1$_T$	9.81 ± 2.96	10.84 ± 1.89	4.09 ± 2.04	5.80 ± 1.29	13.27 ± 0.26	6.38 ± 0.00	0.79 ± 0.29	7.71 ± 0.00	–
MSL	P_T	12.10 ± 0.81	24.94 ± 1.76	16.31 ± 1.50	7.58 ± 0.46	11.10 ± 0.25	5.27 ± 0.00	4.07 ± 1.32	6.90 ± 0.00	–
	R_T	21.14 ± 2.62	22.93 ± 2.17	19.96 ± 1.92	18.71 ± 0.46	2.91 ± 0.25	81.25 ± 0.00	55.42 ± 1.32	42.30 ± 0.00	–
	F1$_T$	6.71 ± 0.23	15.35 ± 1.33	8.06 ± 0.47	3.08 ± 0.29	1.56 ± 0.05	8.13 ± 0.00	6.27 ± 1.40	7.27 ± 0.00	–
SMAP	P_T	12.02 ± 0.95	15.09 ± 0.31	8.17 ± 0.48	11.02 ± 0.00	07.08 ± 0.94	3.63 ± 0.00	1.92 ± 0.71	11.02 ± 0.00	–
	R_T	26.34 ± 0.79	19.13 ± 1.64	6.90 ± 1.23	54.10 ± 0.00	3.88 ± 0.94	82.38 ± 0.00	35.29 ± 0.71	54.10 ± 0.00	–
	F1$_T$	6.42 ± 0.25	7.61 ± 0.30	2.27 ± 0.13	8.43 ± 0.00	0.90 ± 0.11	6.48 ± 0.00	2.96 ± 0.61	8.43 ± 0.00	–
UCR	P_T	1.06 ± 0.02	7.53 ± 0.30	1.06 ± 0.40	1.13 ± 0.17	7.08 ± 0.94	0.79 ± 0.00	2.70 ± 0.35	0.76 ± 0.00	33.12 ± 0.00
	R_T	22.95 ± 0.29	25.10 ± 1.09	0.11 ± 0.40	1.90 ± 0.17	3.88 ± 0.94	82.86 ± 0.00	38.95 ± 0.35	29.11 ± 0.00	33.30 ± 0.00
	F1$_T$	1.67 ± 0.02	4.96 ± 0.10	0.15 ± 0.05	0.51 ± 0.03	0.90 ± 0.11	1.45 ± 0.00	3.62 ± 0.32	1.21 ± 0.00	**25.88** ± 0.00
TODS	P_T	4.18 ± 0.42	5.32 ± 1.22	6.20 ± 0.42	8.01 ± 9.7	7.08 ± 0.94	2.47 ± 0.00	9.79 ± 8.65	6.79 ± 0.00	–
	R_T	47.95 ± 1.15	55.49 ± 17.24	17.75 ± 0.42	0.19 ± 9.70	3.88 ± 0.94	80.28 ± 0.00	6.56 ± 8.65	63.96 ± 0.00	–
	F1$_T$	7.67 ± 0.65	8.37 ± 1.36	8.86 ± 0.75	0.23 ± 0.35	0.90 ± 0.11	4.72 ± 0.00	4.37 ± 0.35	12.22 ± 0.00	–
Avg. F1$_T$		6.46 ± 2.67	**9.43** ± 3.51	4.69 ± 3.33	3.61 ± 3.14	3.51 ± 4.89	5.43 ± 2.26	3.60 ± 1.79	<u>7.37</u> ± 3.54	–

Table 4.8: The *front-bias* performance (in %) of the tested methods on the five benchmarks using the metrics proposed by Tatbul et al. [144]. The average and the standard deviation of five runs are reported.

Dataset		USAD	GDN	THOC	MTAD-GAT	DAGMM	OC-SVM	iForest	ARIMA	DAMP
SWaT	P_T	5.92± 2.14	7.05± 1.73	29.05± 11.13	3.44± 0.85	10.40± 0.22	3.37± 0.00	0.40± 0.15	3.82± 0.00	–
	R_T	33.82± 4.32	25.23± 3.01	2.17± 1.31	43.91± 4.31	20.18± 0.22	85.71± 0.00	90.28± 0.15	80.57± 0.00	–
	F1$_T$	9.93± 3.03	10.84± 1.88	4.01± 2.36	6.34± 1.40	13.72± 0.25	6.48± 0.00	0.80± 0.30	7.31± 0.00	–
MSL	P_T	12.08± 0.81	25.18± 1.80	15.90± 1.41	7.60± 0.46	11.09± 0.25	5.67± 0.00	4.10± 1.33	8.31± 0.00	–
	R_T	22.21± 3.30	25.47± 2.22	23.79± 2.48	18.46± 0.46	4.05± 0.25	84.91± 0.00	60.78± 1.33	40.78± 0.00	–
	F1$_T$	6.66± 0.37	16.52± 1.76	7.41± 0.43	3.01± 0.23	1.52± 0.04	9.27± 0.00	6.55± 1.49	8.91± 0.00	–
SMAP	P_T	12.04± 0.96	15.06± 0.32	8.00± 0.50	13.63± 0.00	7.08± 0.94	5.82± 0.00	1.89± 0.70	13.63± 0.00	–
	R_T	26.08± 0.96	22.54± 1.79	8.13± 0.47	54.03± 0.00	4.16± 0.94	84.84± 0.00	38.90± 0.70	54.03± 0.00	–
	F1$_T$	6.56± 0.39	8.47± 0.25	02.38± 0.18	11.46± 0.00	0.89± 0.10	9.62± 0.00	3.01± 0.69	11.46± 0.00	–
UCR	P_T	1.04± 0.02	7.59± 0.29	1.06± 0.40	1.14± 0.17	7.08± 0.94	0.90± 0.00	2.69± 0.35	0.75± 0.00	33.25± 0.00
	R_T	23.75± 0.24	25.24± 1.23	0.15± 0.40	2.01± 0.17	4.16± 0.94	84.21± 0.00	42.58± 0.35	29.20± 0.00	36.07± 0.00
	F1$_T$	1.65± 0.03	4.39± 0.19	0.19± 0.08	0.54± 0.06	0.89± 0.10	1.51± 0.00	3.71± 0.32	1.21± 0.00	**29.40**± 0.00
TODS	P_T	4.26± 0.40	5.41± 1.31	6.22± 0.46	8.02± 9.70	7.08± 0.94	2.04± 0.00	9.72± 8.62	6.82± 0.00	–
	R_T	47.91± 1.16	55.57± 17.02	17.72± 0.46	0.18± 9.70	4.16± 0.94	80.65± 0.00	6.45± 8.62	64.05± 0.00	–
	F1$_T$	7.77± 0.66	8.47± 1.44	8.94± 0.73	0.23± 0.35	0.89± 0.10	3.97± 0.00	4.34± 0.39	12.27± 0.00	–
Avg. F1$_T$		6.51± 2.72	**9.76**± 3.97	4.59± 3.21	4.32± 4.19	3.58± 5.07	6.17± 3.10	3.68± 1.87	<u>8.23</u>± 3.93	–

Table 4.9: The *middle-bias* performance (in %) of the tested methods on the five benchmarks using the metrics proposed by Tatbul et al. [144]. The average and the standard deviation of five runs are reported.

Dataset		USAD	GDN	THOC	MTAD-GAT	DAGMM	OC-SVM	iForest	ARIMA	DAMP
SWaT	P_T	5.76 ± 2.08	7.01 ± 1.74	29.05 ± 11.13	3.51 ± 0.87	10.18 ± 0.22	2.92 ± 0.00	0.37 ± 0.14	3.85 ± 0.00	–
	R_T	33.67 ± 4.49	25.05 ± 2.86	1.88 ± 1.03	32.61 ± 2.91	19.61 ± 0.22	85.61 ± 0.00	90.14 ± 0.14	80.50 ± 0.00	–
	F1$_T$	9.69 ± 2.96	10.78 ± 1.87	3.52 ± 1.89	6.28 ± 1.35	13.40 ± 0.26	5.65 ± 0.00	0.75 ± 0.29	7.34 ± 0.00	–
MSL	P_T	12.31 ± 0.73	24.94 ± 1.72	15.53 ± 1.37	7.78 ± 0.44	11.07 ± 0.25	5.28 ± 0.00	3.72 ± 1.14	9.96 ± 0.00	–
	R_T	19.14 ± 2.64	24.05 ± 2.01	24.71 ± 2.45	18.20 ± 0.44	4.67 ± 0.25	85.07 ± 0.00	61.51 ± 1.14	39.82 ± 0.00	–
	F1$_T$	6.48 ± 0.37	15.58 ± 1.30	7.06 ± 0.37	3.25 ± 0.16	1.64 ± 1.64	8.87 ± 0.00	6.13 ± 1.37	10.76 ± 0.00	–
SMAP	P_T	12.06 ± 0.25	14.91 ± 0.33	8.11 ± 0.47	16.99 ± 0.00	7.11 ± 0.94	9.49 ± 0.00	1.64 ± 0.57	16.99 ± 0.00	–
	R_T	23.67 ± 0.70	21.33 ± 1.57	9.23 ± 0.42	51.33 ± 0.00	3.69 ± 0.94	84.91 ± 0.00	39.27 ± 0.57	51.33 ± 0.00	–
	F1$_T$	6.12 ± 0.25	8.07 ± 0.33	2.75 ± 0.22	13.75 ± 0.00	0.92 ± 0.08	13.30 ± 0.00	2.70 ± 0.65	13.75 ± 0.00	–
UCR	P_T	1.06 ± 0.02	7.44 ± 0.35	1.06 ± 0.40	1.14 ± 0.17	7.11 ± 0.94	0.87 ± 0.00	2.39 ± 0.32	0.73 ± 0.00	31.86 ± 0.0
	R_T	23.45 ± 0.20	24.92 ± 1.26	0.11 ± 0.40	1.88 ± 0.17	3.69 ± 0.94	84.26 ± 0.00	42.83 ± 0.32	29.03 ± 0.00	35.09 ± 0.0
	F1$_T$	1.67 ± 0.03	4.11 ± 0.18	0.17 ± 0.06	0.54 ± 0.06	0.92 ± 0.08	1.21 ± 0.00	3.37 ± 0.31	1.16 ± 0.00	**26.68** ± 0.0
TODS	P_T	4.21 ± 0.39	5.42 ± 1.40	6.14 ± 0.63	8.12 ± 9.70	7.11 ± 0.94	1.47 ± 0.00	9.70 ± 8.59	6.92 ± 0.00	–
	R_T	47.96 ± 1.16	55.43 ± 17.02	17.57 ± 0.63	0.18 ± 9.70	3.69 ± 0.94	80.80 ± 0.00	6.34 ± 8.59	63.93 ± 0.00	–
	F1$_T$	7.69 ± 0.65	8.48 ± 1.58	8.82 ± 0.84	0.29 ± 0.37	0.92 ± 0.08	2.89 ± 0.00	4.32 ± 0.46	12.43 ± 0.00	–
Avg. F1$_T$		6.33 ± 2.64	**9.40** ± 3.76	4.46 ± 3.10	4.82 ± 4.96	3.56 ± 4.93	6.38 ± 4.33	3.45 ± 1.78	<u>9.09</u> ± 4.51	–

Table 4.10: The *back-bias* performance (in) %) of the tested methods on the five benchmarks using the metrics proposed by Tatbul et al. [144]. The average and the standard deviation of five runs are reported.

should be considered [133].

(ii) The considered forecasting approaches tend to have the most consistent range-based performance with respect to standard metrics.

(iii) Traditional approaches are not necessarily obsolete; in some cases, they can achieve performances that are comparable to DL methods. This supports the claim of [133] to question the assumption that deep learning is the definitive solution for time-series anomaly detection. Since they are usually recall-oriented, they usually detect most types of anomalies but at the cost of a higher false positive rate.

(iv) The Point Adjustment (PA) protocol is unreliable as it overestimates the detector performance, and in the case of traditional approaches that are already recall-oriented, this triggers an even higher false positive rate.

(v) Multivariate time-series are challenging due to the high dimensionality of data. On the other hand, univariate time-series can be challenging when the anomaly ratio is very low.

(vi) Most models achieve low performance using range-based metrics, highlighting the difficulty of detecting the anomaly boundaries.

(vii) Model stability and memory consumption can vary importantly from one method to another. Hence, depending on the end-goal application, these metrics can be essential for selecting the most suitable model in accordance with the hardware specifications.

4.6 Conclusion

This chapter proposes an extensive evaluation study of recent time-series anomaly detection methods. To the best of our knowledge, we are the first to analyze these algorithms based on a more elaborate experimentation protocol. In contrast to previous evaluation studies, which only consider the standard performance metrics, we take into account revisited performance criteria specifically designed for time series in our analysis. In addition, the model stability, the model size as well as the robustness to different types of anomalies are also investigated. All these additional elements give a more complete picture of the current state-of-the-art. Moreover, the proposed protocol is timely and could be beneficial for future investigations, providing more insights regarding their applicability in a real-world context. In particular, the insights in this study are leveraged in the next chapter, which presents an unsupervised deepfake detector that extends the UAD formulation to the video level detection with the aim of improving generalization.

Chapter 5

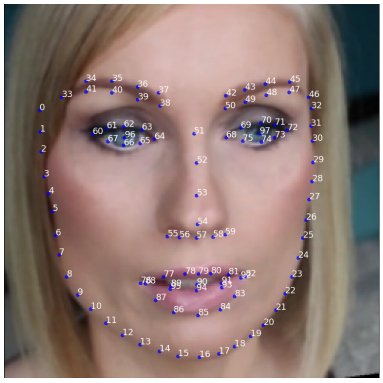
Facial Region-Based Ensembling for Unsupervised Temporal Deepfake Localization

This chapter addresses the challenge of temporal deepfake localization. Instead of classifying entire videos as real or fake, the goal is isolating forged frames in untrimmed videos that might be partially manipulated. Recently, few deepfake localization methods have emerged. They are mostly supervised, therefore relying on costly annotations and suffering from a lack of generalization to unseen manipulations. Similar to image-based deepfake detection, we propose reformulating deepfake localization as an unsupervised time-series anomaly detection problem. Hence, to investigate the relevance of the proposed formulation, recent state-of-the-art techniques in anomaly detection for time-series are evaluated in the context of deepfake localization.

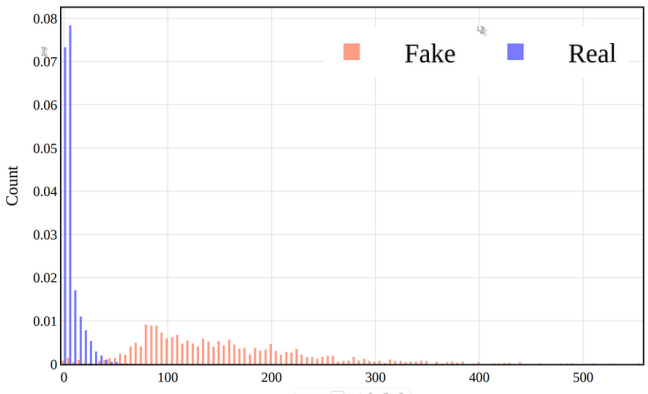
5.1 Introduction

The rise of deepfake technology, involving the creation of realistic facial media using Deep Neural Networks (DNN), calls into question the credibility of digital content [12, 13]. One

major risk is the misuse of these manipulated data for spreading misinformation. Consequently, the development of effective deepfake detection methods has become crucial. Current deepfake detection strategies [23, 28] generally rely on binary classification, focusing on the prediction of one label for an entire video. Hence, these approaches simplify the problem by assuming that forged videos are temporally segmented. This, however, hinders their application in a real-world scenario, especially if real-time performances are required. A more plausible approach would be to localize deepfakes in an untrimmed video stream that can be locally forged. Recently, a few methods have been proposed for temporal localization [202, 26, 203, 204]. The latter are trained in a supervised manner, thereby inheriting two major shortcomings. First, a large set of annotated data is needed, which can be costly and hard to obtain. Second, as discussed in [23, 35], overfitting issues can occur, causing a poor generalization to unseen manipulations.



(a) Example of extracted facial landmarks and their identifiers (IDs).



(b) Histogram of the standard deviation of landmark displacements for real and fake images.

Figure 5.1: Comparison of landmark displacement statistics extracted from fake and real videos in ForgeryNet [26].

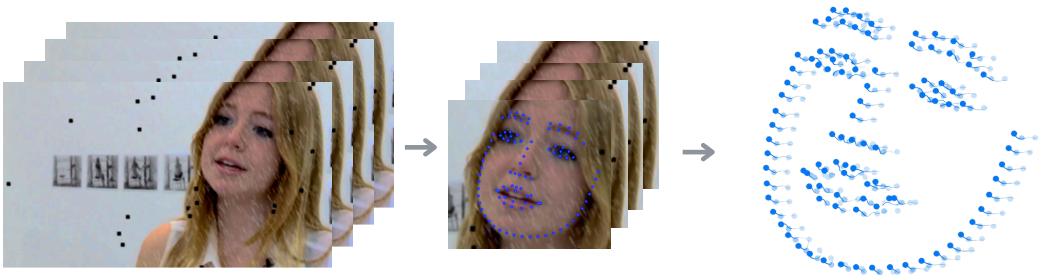
Building on our pioneering work on unsupervised deepfake detection [35], we propose to reformulate the problem of deepfake localization as an unsupervised anomaly detection problem in multivariate time-series. In other words, we suggest learning a time-series model using only real videos and considering out-of-distribution frames as deepfakes at inference. Specifically, we represent each video by the position trajectories of facial landmarks. These

trajectories can be seen as a multivariate time-series, which can be prone to temporal inconsistencies in the case of forged videos. As experimentally demonstrated in Figure 5.1, a noticeable discrepancy exists between the standard deviation of landmark displacements of real and fake videos. Furthermore, such a geometric representation has the advantage of being low-dimensional, resulting in more compact models. It is also universal across all datasets as it is robust to illumination changes and image content, thereby reducing overfitting risks. Hence, we propose to study the suitability of recent time-series anomaly detection for the concrete use case of landmark-based deepfake localization. Furthermore, we propose a simple, yet effective, region-based ensembling strategy for deepfake localization relying on autoencoder (AE) architectures. Extensive experiments and analysis demonstrate the relevance of the proposed formulation as well as the introduced ensembling methods, suggesting a promising direction for future research in deepfake temporal localization.
In short, our contributions can be summarized as follows: (1) The formulation of temporal deepfake localization as an unsupervised anomaly detection problem in time-series. (2) An ensemble of lightweight autoencoders focusing on facial regions, trained only on real videos. (3) A comprehensive analysis and comparison of recent anomaly detection techniques on time-series in the context of deepfake localization.
This chapter is structured as follows: Section 5.2 formulates temporal deepfake localization as a time-series anomaly detection problem. Section 5.3 details the proposed region-based ensembling approach. Section 5.4 describes the experiments and analyzes the results. Finally, Section 5.5 concludes this work.

5.2 Unsupervised Anomaly Detection in Time Series for Deepfake Localization using Geometric Representations

An untrimmed video $\mathbf{V}$ can be defined as a temporally-ordered sequence of T images denoted as $\mathbf{V} = \{\mathbf{I}_t\}_{1 \leq t \leq T}$ with $\mathbf{I}_t \in \mathbb{R}^{h \times w \times c}$ and h, w and c being the height, width and the number of channels of $\mathbf{I}_t$, respectively. We assume that $\mathbf{l} = \{l_t\}_{1 \leq t \leq T}$ corresponds to the ground-truth label vector of $\mathbf{V}$, with $l_t \in \{0, 1\}$ representing the label of $\mathbf{V}$ at an instant t.

(a) **Input videos pre-processing into landmark sequences**



(b) **Ensembling of per-facial regions encoders**

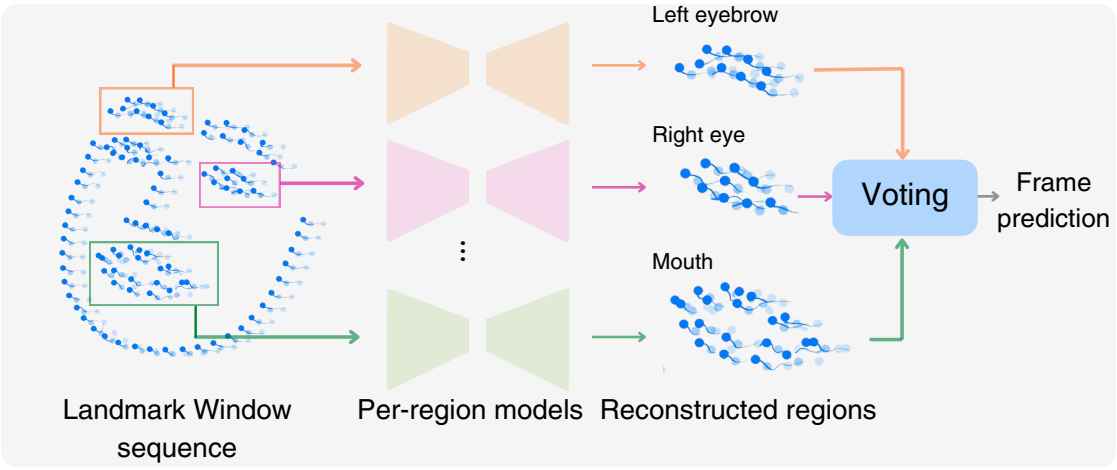


Figure 5.2: Overview of the proposed facial region-focused ensembling. (a) Extraction of facial landmark sequences to be used for training several individual Autoencoders (AEs), each trained on a specific facial region trajectories (i.e., nose and mouth). (b) Inference from the per-facial regions Autoencoders (AEs) and aggregation of the individual results via a voting strategy to produce the finale frame prediction.

Note that $l_t = 1$ in the presence of a forgery and $l_t = 0$ otherwise. We denote by $\mathbf{V}_t$ the subsequence of $\mathbf{V}$ formed by $(\mathbf{I}_{t-\tau_1}, \mathbf{I}_{t-\tau_1+1}, ..., \mathbf{I}_t, ..., \mathbf{I}_{t+\tau_2-1}, \mathbf{I}_{t+\tau_2})$ with $\tau_1$and τ_2 two integers defining the position and the size of a sliding window and T_s being its length. The goal of deepfake localization is estimating a function $f : \mathbb{R}^{h\times w\times c\times T_s} :\rightarrow \{0, 1\}$ for all t,

$$f(\mathbf{V}_t) = l_t. \tag{5.1}$$

Existing localization methods mostly learn f in a supervised manner using deep learning architectures [26, 203, 204]; thereby relying on costly annotations. Moreover, supervision leads to a lack of generalization to unseen manipulations, as this was demonstrated in the context of deepfake detection [23, 35]. To address this issue, we propose reformulating the problem of deepfake localization as an unsupervised anomaly detection task. Thus f can be viewed as a composition of two functions $f = \Phi \circ \Psi$ where $\Psi : \mathbb{R}^{h\times w\times c\times T_s} \rightarrow \mathcal{X}$ models normal time-series and is learned using only real data and $\Phi : \mathcal{X} \rightarrow \{0, 1\}$ is a thresholding function only used at inference.

Another aspect that should be considered is the model size. In fact, existing multivariate time-series anomaly detection architectures have been initially designed for relatively low dimensional data [205]. Hence, directly modelling videos as time-series might results in cumbersome models. As a solution, we propose the use of geometric representations, e.g., 2D facial landmarks. As shown in [206], in the context of deepfake detection, they can be used for obtaining more compact models, while demonstrating more robustness to illumination changes and noise. In other words, Ψ can be defined as $\Psi = \Psi_2 \circ \Psi_1$ such that $\Psi_1 : \mathbb{R}^{h\times w\times c\times T_s} \rightarrow \mathbb{R}^{2\times n\times T_s}$ maps a video subsequence to its corresponding 2D facial landmark subsequence and $\Psi_2 : \mathbb{R}^{2\times n\times T_s} \rightarrow \mathcal{X}$.

5.3 Facial region-based ensembling for unsupervised deepfake localization

As unsupervised anomaly detection for time-series approaches were not originally proposed for deepfake temporal localization, they do not explicitly focus on artifact-prone facial regions. Those regions, however, have been proven to be extremely effective in capturing deepfake artifacts [207, 35] from different deepfake generation methods. Hence, as illustrated in Figure 5.2, we propose an ensembling strategy of different models that are focused on localized regions. For that purpose, we train an ensemble of K autoencoders, each one trained on a subset of landmarks belonging to a manually-selected facial region such as the mouth or the nose. Then, a voting strategy is applied for building the final predictions.

More specifically, given an input video $\mathbf{V}_t \in \mathbb{R}^{h \times w \times c \times T_s}$ processed as a 2D landmark sequence denoted by $\mathbf{X}_t = \Psi_1(\mathbf{V}_t) \in \mathbb{R}^{2 \times n \times T_s}$, we select K specific regions. We denote the position of the set of the n_k landmarks belonging to the region of index $k \in \{1, ..., K\}$ as $\mathbf{X}_t^k =\in \mathbb{R}^{2 \times n_k \times T_s}$. For each $k \in \{1, ..., K\}$, an autoencoder that aims at learning the distribution of authentic region-specific landmark trajectories is considered. To this end, given an encoder $\text{Enc}_k(.)$ and a decoder $\text{Dec}_k(.)$, for all windowed sequences $\mathbf{X}_t^k$, our model is trained as,

$$\begin{cases} \mathbf{z} = \text{Enc}_k(\mathbf{X}_t^k), \\ \hat{\mathbf{X}}_t^k = \text{Dec}_k(\mathbf{z}), \end{cases} \tag{5.2}$$

with $\mathbf{z} \in \mathbb{R}^{T_s \times d}$ being the d-dimensional latent representation. The learning is optimized using the mean squared distance formulated as,

$$\mathcal{L}_r = \frac{1}{N_b} \sum_{t=0}^{N_b} ||\hat{\mathbf{X}}_t^k - \mathbf{X}_t^k||_2^2, \tag{5.3}$$

with N_b being the total batch samples and $||\cdot||_2$ denoting the L2-norm. Similarly to [208, 209], a statistical model termed Peak Over Threshold (PoT) [210] is used to automatically select an adequate threshold λ based on the training sequences. Such an approach identifies a suitable value at risk by fitting the distribution of the training data with a Generalized Pareto

Distribution. Hence, given λ and a window $\mathbf{X}_t^k$, the prediction l_t^k associated with the frame t is expressed as,

$$l_t^k = \mathbb{1}(\frac{1}{2 \times n_k \times T_s} \sum_{i}^{2n_k} \sum_{j}^{T_s} ||\hat{\mathbf{X}}_{t,i,j}^k - \mathbf{X}_{t,i,j}^k||_2^2 > \lambda). \tag{5.4}$$

with $\mathbb{1}(.)$ being an indicator function. Finally, given the K predictions l_t^k from different autoencoders, the final prediction is built via the soft majority voting rule.

5.4 Experiments

5.4.1 Experimental settings

Dataset

We used the ForgeryNet dataset [26], a comprehensive benchmark for temporal forgery localization. It is formed by 2,896,062 images and 221,247 partially manipulated videos. Nevertheless, we select only the data that contain one person per video. In total, we consider 9866 real videos from the official training set and 1,516 videos from the validation set for testing (as annotations for the test set are not yet available). Note that our test set comprises six different forgery methods.

Baselines

In addition to the proposed ensembling approach, we evaluate seven recent anomaly detection methods for time-series. As discussed in [43], these approaches adopt different learning paradigms. First, four reconstruction-based methods are considered, namely **TranAD** [209], **USAD** [162], **OmniAnomaly** [208] and **MAD-GAN**. TranAD and USAD are respectively based on transformer and AE architectures that are trained adversarially. OmniAnomaly uses a stochastic Recurrent Neural Network (RNN) and a planar normalizing flow to generate reconstruction probabilities. Finally, MAD-GAN is a GAN-inspired approach using an RNN as a base model for modeling spatio-temporal dependencies. Second, forecasting approaches are also tested for the use case of deepfake localization, including **CAE-M** [205],

DAGMM [180] and **GDN** [160]. CAE-M feeds a reconstruction error and the learned feature representations to an auto-regressive network that predicts future feature values. DAGMM constrains the feature space to follow Gaussian mixture model distribution. Then, an RNN is employed for predicting a future data point. Last but not least, GDN models the relationships between data features as a graph coupled with an attention mechanism. For comparing with supervised deepfake localization methods [26, 204, 203, 202], only **MDS** [202], a multimodal technique with decoupled audio-video networks, is compatible with our setting. It maximizes the similarity of real audio and real visual features and minimizes it otherwise. The other baselines either require audiovisual input data or are not accessible.

Evaluation metrics

For evaluating the proposed ensembling strategy as well as state-of-the-art methods, we report the following metrics: the standard Precision, Recall, and F1-score metrics. Note that the results are reported with and without the Point Adjustment protocol, referred to as (**PA**) and (**non-PA**), respectively. The PA protocol proposed in [161] is commonly used for evaluating unsupervised anomaly detection in time-series [209, 162, 208]. It assumes that if a single point within an anomalous segment is detected, then the entire segment is correctly predicted as anomalous. Furthermore, we compute the t-Precision, the t-Recall, and the t-F1-scores [144], which are metrics tailored for time-series by taking into account factors like the location of detected anomalies and the cumulative overlap between predicted and ground-truth segments. Finally, similar to deepfake detection methods, we also report the Area Under the Curve (AUC) metric. In all our experiments, **Bold** and underline report the best and second best results, respectively.

Implementation details

For each video, we detect and crop the faces. Then, we extract from each frame a total of 98 landmarks using SPIGA [211]. The landmark values are normalized between 0 and 1. The average lengths of training and testing sequences are respectively equal to 160 and 119

	Method	Paradigm	AUC	Precision	Recall	F1-score
Sup.	MDS [202]	–	0.4943	0.4704	0.6931	0.5604
Unsup.	TranAD [209]	Reconstruction	0.7177	0.8018	0.4878	0.6066
	USAD [162]	Reconstruction	0.7779	0.8330	0.5000	0.6046
	DAGMM [180]	Probabilistic	0.7573	0.8314	0.5644	0.6724
	GDN [160]	Forecasting	0.7837	**0.8397**	0.6187	0.7125
	MAD-GAN [164]	Reconstruction	0.8497	0.7980	0.7859	0.7919
	OmniAnomaly [208]	Probabilistic	0.7068	0.7998	0.4642	0.5874
	CAE-M [205]	Reconstruction	<u>0.9182</u>	<u>0.8385</u>	<u>0.9130</u>	<u>0.8742</u>
	Ours	Reconstruction	**0.9302**	0.8090	**0.9538**	**0.8754**

Table 5.1: Results in terms of standard performance metrics on ForgeryNet under the PA protocol.

frames. We use the same autoencoder architecture as proposed in CAE-M [205] from this repository[1]. The models are trained 5 epochs, one sequence at a time on an NVIDIA RTX A4000 GPU. We use the AdamW [212] optimizer with a learning rate of 10^{-3} and weight decay of 10^{-5}.

5.4.2 Results

Performance using standard metrics

Table 5.1 and Table 5.2 report the obtained results in terms of Precision, Recall, and F1-score with and without the PA protocol, respectively. In the former, it can be noted that the proposed ensemble generally outperforms other approaches including the supervised baseline. This confirms the adequacy of following a region-based strategy for spatially modelling deepfake artifacts. It can also be seen that except CAE and MAD-GAN, most approaches unsupervised achieve comparable results. Hence, it remains unclear whether reconstruction-based on forecasting methods are more suitable for the complex scenario of deepfake localization. This might also suggest that both reconstruction and forecasting approaches are able to capture discrepancies. In the latter case, when the predictions are not adjusted, it can be observed that all approaches suffer from an expected significant

[1] https://github.com/imperial-qore/TranAD/

	Method	Paradigm	AUC	Precision	Recall	F1-score
Sup.	**MDS** [202]	–	0.4663	**0.5104**	**0.3325**	**0.4027**
Unsup.	**TranAD** [209]	Reconstruction	0.4967	0.2757	0.0459	0.0787
	USAD [162]	Reconstruction	0.5080	0.3657	0.0647	0.1100
	DAGMM [180]	Probabilistic	0.5015	0.3155	0.0527	0.0904
	GDN [160]	Forecasting	0.5153	0.4097	0.0820	0.1367
	MAD-GAN [164]	Reconstruction	<u>0.5425</u>	0.4629	0.1715	0.2503
	OmniAnomaly [208]	Probabilistic	0.4933	0.2417	0.0370	0.0641
	CAE-M [205]	Reconstruction	0.5314	<u>0.4720</u>	0.1222	0.1941
	Ours	Reconstruction	**0.5491**	0.4597	<u>0.1916</u>	<u>0.2704</u>

Table 5.2: Results in terms of standard performance metrics on ForgeryNet under the non-PA protocol.

performance drop. Nevertheless, our ensemble still surpasses unsupervised state-of-the-art methods, including CAE-M with an increase of 1.77% and 7.63% in terms of AUC and F1-score, against 1.2% and 0.12% under the PA protocol, respectively. In comparison with MDS, although they reach better precision, recall and F1-score under the non-PA protocol, we achieve a higher AUC of 93.02% and 54.91% with the PA and non-PA protocols respectively. This suggests that with our approach the forged and real frames are more separable than with the supervised baseline. Additionally, contrary to MDS, our method is trained using a single modality and does not require annotated deepfake data. Notably, MDS presents overfitting signs since it achieves significantly higher AUC under the in-dataset setting reported in [202], with more than 90% against 46.63% under the cross-dataset setting (see Table 5.2).

Performance using range-based metrics

Table 5.4 and Table 5.3 report the obtained results on ForgeryNet in terms of range-based metrics including the t-Precision, the t-Recall, and the t-F1-score proposed in [144], under the PA and the non-PA protocols, respectively. It can be observed from Table 5.3 that the obtained results with range-based metrics are consistent with the standard metrics results shown in Table 5.2 and Table 5.1. In fact, the proposed ensemble achieves the highest

	Method	Paradigm	t-Precision	t-Recall	t-F1-score
Sup.	**MDS** [202]	–	0.3039	**0.3348**	<u>0.2376</u>
Unsup.	**TranAD** [209]	Reconstruction	0.3424	0.0552	0.0815
	USAD [162]	Reconstruction	0.4101	0.0763	0.1110
	DAGMM [180]	Probabilistic	0.3730	0.0639	0.0937
	GDN [160]	Forecasting	0.4226	0.0933	0.1229
	MAD-GAN [164]	Reconstruction	**0.5066**	0.2068	0.2239
	OmniAnomaly [208]	Probabilistic	0.3130	0.0434	0.0657
	CAE-M [205]	Reconstruction	<u>0.4683</u>	0.1546	0.2041
	Ours	Reconstruction	0.4354	<u>0.2362</u>	**0.2706**

Table 5.3: Results in terms of range-based metrics (t-Precision, t-Recall and t-F1-score) proposed in [144] on ForgeryNet under the non-PA protocol.

t-F1-score, followed by MAD-GAN and CAE-M. This demonstrates the robustness of our strategy as compared to unsupervised state-of-the-art techniques, suggesting that it can detect consecutive anomalies rather than random point anomalies. However, in Table 5.4, we observe that our method is no longer the best performing. This can be explained by the fact that the adjustment harms our performance, by boosting the t-Recall at the expense of the t-Precision. Notably, PA does not always yield a reliable comparison. As shown in [147], it can boost the performance a random detector making it comparable to a well-trained one. Furthermore, the compatibility of this protocol with range-based metrics is debatable. In fact, by treating an anomalous segment and a single point equally, the temporal information that range-based metrics aim to capture, based on the predicted anomaly location and cumulative overlap, is dissipated. Hence, we report only non-PA results in the following experiments.

Selection of facial regions

Since we propose a facial region-focused ensemble, we report in Table 5.5 the performance of our AE trained on different facial regions. The best performance is achieved using the jawline and eyebrows models. This can be explained by the fact that during the blending stage, deepfake generation methods fail to perfectly align the foreground and background

	Method	Paradigm	t-Precision	t-Recall	t-F1-score
	MDS [202]	–	0.2321	0.6521	0.3034
Unsup.	**TranAD** [209]	Reconstruction	0.3357	0.4537	0.3653
	USAD [162]	Reconstruction	0.3926	0.5582	0.4281
	DAGMM [180]	Probabilistic	0.3564	0.5303	0.3959
	GDN [160]	Forecasting	0.3847	0.5852	0.4226
	MAD-GAN [164]	Reconstruction	**0.4214**	0.7591	**0.4842**
	OmniAnomaly [208]	Probabilistic	0.3039	0.4361	0.3364
	CAE-M [205]	Reconstruction	0.3465	0.8635	0.4635
	Ours	Reconstruction	0.2487	**0.9369**	0.3695

Table 5.4: Results using range-based metrics proposed in [144] on ForgeryNet under the PA protocol.

Facial regions	#Landmarks	AUC	Precision	Recall	F1-score
Pupils (P)	2	0.5186	0.4109	0.0916	0.1498
Left Eye (LE)	8	0.5182	0.4064	0.0923	0.1505
Right Eye (RE)	8	0.5060	0.3409	0.0602	0.1023
Eyes (E)	16	0.5216	0.4265	0.0976	0.1588
Left Brow (LB)	9	0.5257	0.4451	0.1065	0.1718
Right Brow (RB)	9	0.5115	0.3733	0.0756	0.1258
Brows (B)	18	**0.5346**	**0.4796**	**0.1257**	**0.1992**
Nose (N)	9	0.5104	0.3603	0.0786	0.1290
Mouth (M)	20	0.5092	0.3587	0.0710	0.1185
Jawline (J)	33	0.5296	0.4672	0.1121	0.1808

Table 5.5: Results using individual facial regions on ForgeryNet in terms of standard performance metrics under the non-PA protocol.

faces, resulting in noisy landmarks within those facial areas. It is also interesting to observe the mismatch between the left and right facial areas. Specifically, a difference in terms F1-score of 4.70% and 4.77% can be observed between the right and the left eyebrows, and the left and right eye respectively.

Role of the ensembling

Table 5.6 gives the obtained results by considering different combinations of the three most relevant regions. Mainly, we compare the simple concatenation of the region-based geomet-

Brows	Eyes	Jawline	Ensembled	AUC	F1-score
✗	✓	✓	✓	**0.5388**	**0.2446**
✗	✓	✓	✗	0.5309	0.1869
✓	✗	✓	✓	**0.5491**	**0.2704**
✓	✗	✓	✗	0.5173	0.1497
✓	✓	✗	✓	**0.5392**	**0.2368**
✓	✓	✗	✗	0.5301	0.1854
✓	✓	✓	✓	**0.5513**	**0.2913**
✓	✓	✓	✗	0.5374	0.2088

Table 5.6: Feature combination versus ensembling strategy of the three most relevant facial regions under the non-PA protocol. Experiments are performed on ForgeryNet.

ric features against the proposed ensemble strategy. It can be noted that the ensembling consistently enhances the performance as compared to the direct concatenation of region-based features within a single model. This might be explained by the fact that implicitly learning region-based features with a single model is challenging. As discussed in [100], capturing local artifacts using a CNNs is not straightforward as successive convolution layers tend to eliminate low-level features.

Model size

Finally, Table 5.7 reports the number of parameters of each method. It can be seen that our method as well as CAE-M have a significantly lower number of parameters in comparison to state-of-the-art techniques, including the supervised baseline MDS (with 7.229 against 122.777.092 parameters).

5.4.3 Discussion

Figure 5.3 illustrates a challenging case that may impact the performance of our method, which originates from the use of geometric representations, specifically, the facial landmarks under unconstrained conditions. In fact, faces captured in the wild may be subject to motion blur, extreme poses, or occlusions, which leads to unreliable face and landmark detection,

	Method	Paradigm	#Parameters
Sup.	**MDS** [202]	–	122.777.092
Unsup.	**TranAD** [209]	Reconstruction	3197004
	USAD [162]	Reconstruction	50.109
	DAGMM [180]	Probabilistic	50.016
	GDN [160]	Forecasting	20.074
	MAD-GAN [164]	Reconstruction	48.613
	OmniAnomaly [208]	Probabilistic	38.872
	CAE-M [205]	Reconstruction	7.229
	Ours	Reconstruction	7.229

Table 5.7: Number of model parameters

as seen in the blurry image of Figure 5.3. This case highlights the importance of robust face and landmark detection in ensuring consistent deepfake detection performance.

Figure 5.3: Landmark detection fails on blurry images, despite the subject being real.

5.5 Conclusion

In this chapter, temporal deepfake localization has been formulated as an unsupervised time-series anomaly detection problem. To assess the suitability of the proposed formulation, state-of-the-art methods in the general field of time-series anomaly detection have been benchmarked under the complex scenario of deepfake localization. Instead of using raw videos, a geometric representation is used, namely, the trajectories of facial landmarks, enabling the use of relatively lightweight architectures. Furthermore, to better model localized artifacts, a facial region-based ensembling strategy has been introduced. The obtained

results have not only demonstrated the relevance of the proposed formulation but have also shown the superiority of the introduced ensembling method as compared to state-of-the-art techniques. However, our approach might be sensitive to compression and noisy landmark extraction.

Chapter 6

When Unsupervised Domain Adaptation meets One-class Anomaly Detection: Addressing the Two-fold Unsupervised Curse by Leveraging Anomaly Scarcity

The previous chapters introduced frameworks for type-agnostic image and video deepfake detection, formulated as unsupervised anomaly detection tasks. Despite their relevance, these methods rely on deep learning models, which makes them sensitive to variation in domain-specific factors unrelated to the forgery, such as subject identity or acquisition conditions. This issue is often mitigated using unsupervised domain adaptation (UDA), which motivates the investigations presented in this chapter, which in turn introduces the first fully unsupervised domain adaptation framework for unsupervised anomaly detection (UAD) in generic image classification.

Although UDA has been effective in binary and multi-class classification, extending it to UAD remains a difficult task due to the lack of supervision in both the source and target

domains. We formally define this challenge as the two-fold unsupervised curse. To address it, we propose a novel approach that assumes anomalies are rare in the target domain. It leverages clustering to identify the dominant cluster in the target feature space, assumed to represent normal data, and aligns it with the source normal features. More specifically, it fits a hypersphere around the source features while jointly aligning them with the dominant target-domain feature cluster. Extensive experiments on standard UAD adaptation benchmarks demonstrate the effectiveness of the proposed framework and validate the relevance of this new paradigm. These findings lay the foundation for further investigation into domain-invariant unsupervised anomaly detection techniques for improving the generalization of deepfake detectors.

6.1 Introduction

Anomaly Detection (AD) can be seen as the identification of outliers deviating from a usual pattern. The growing interest in AD in both academia and industry is mainly due to its relevance in numerous practical scenarios, such as early disease detection in medical imaging [213, 214] and industrial inspection [215, 43, 216, 217, 218]. By definition, anomalies rarely occur. Annotating anomalous data is, therefore, often difficult and costly [219, 215, 220], hindering the collection of large-scale datasets. As a result, state-of-the-art methods mostly tackle AD as an unsupervised problem [169, 63], where the objective is to learn only from the normal class.

Despite achieving promising results, recent approaches in AD [70, 216, 221, 222, 219] typically assume that training and inference data are drawn from the same distribution. This assumption does not always hold in unconstrained scenarios, where a *domain shift* [34] between training and testing data can naturally arise due to varying setups, such as different lighting conditions and variations in object pose [220]. As a result, a model trained on a dataset sampled from a given domain, usually called *source* dataset, will show degraded performance when tested on a dataset from a different domain, generally termed *target* dataset. For instance, an AD model for medical imaging trained on images acquired using a

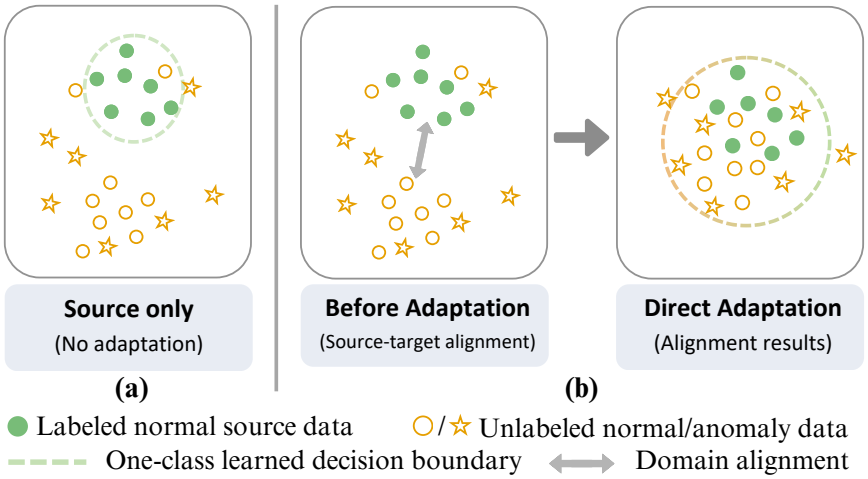


Figure 6.1: **Illustration of the two-fold unsupervised curse**: **(a)** The decision boundary learned from the source set without any adaptation does not allow generalization to the target domain. **(b)** Direct alignment of the unlabeled target with the one-class source features leads to the confusion of normal and abnormal samples.

given Magnetic Resonance Imaging (MRI) device can fail to generalize to samples captured with a different MRI system.

To reduce such a domain gap while avoiding costly annotation efforts, Unsupervised Domain Adaptation (UDA) [223, 224] has proven to be an effective solution in binary and multi-class classification tasks [225, 224]. UDA aims at learning domain-invariant features by relying on labeled source and unlabeled target data at the same time. However, the task of **unsupervised** domain adaptation for **unsupervised** anomaly detection (UAD) is ill-posed as the goal is to: *align the source and the target feature distributions using only normal source data and unlabeled target data formed by both normal and anomalous samples* (see Figure 6.2 (c)). Hence, a direct extension of standard UDA techniques developed for binary/multi-class classification [224, 223] would not be applicable as these methods usually aim at minimizing the distance between the estimated distributions from the entire source and target training sets. Indeed, this would lead to the erroneous alignment of both normal and anomalous target samples with normal source samples, as illustrated in Figure 6.1 (b). Given the learned decision boundary, this would lead to the confusion of normal and abnormal samples from the target set. As it requires addressing two unsupervised tasks

simultaneously, we refer to this described problem as the *two-fold unsupervised curse*.

To the best of our knowledge, no prior work has tried to address this two-fold unsupervised challenge, i.e., , unsupervised domain adaptation for one-class image anomaly detection described in Figure 6.2 (c). Indeed, related works have mainly simplified the problem by either (1) assuming the availability of labeled abnormal and normal source data, resulting in UDA for a binary classification setting [226] (see Figure 6.2 (a)), or (2) maintaining the source one-class setup while accessing only few normal target data referred to as few-shot supervised adaptation for unsupervised anomaly detection [226, 227, 228, 229, 230] (see Figure 6.2 (b)). Nevertheless, annotating even a few samples might still be constraining, particularly in the field of anomaly detection, where expert knowledge is often needed, such as for tumor annotation in medical images [213, 214] or for industrial inspection [216, 221, 218]. Moreover, few-shot adaptation approaches are known to be prone to overfitting issues since few shots cannot fully represent the normal target distribution [231]. This calls for a fully unsupervised domain adaptation approach that leverages the diversity of the available large, unlabeled target datasets.

In this chapter, we investigate whether the rare occurrence of anomalies could be exploited to address the two-fold unsupervised curse. We herein propose the first unsupervised domain adaptation framework for unsupervised image anomaly detection. Our solution starts by identifying a dominant cluster assumed to be formed by normal target data and then aligning it with normal source samples. Specifically, our method utilizes a trainable ResNet-based [123] feature extractor to process both the source and target features. A frozen CLIP visual encoder [232] is also used to generate corresponding target features, which are then clustered using K-means to identify the samples of the dominant cluster. These samples are mapped into the ResNet-based [123] feature space and aligned with the source features. For the domain adaptation task, a contrastive strategy [232, 233] ensures the similarity between the dominant target cluster and normal source samples, while for the anomaly detection task, a Deep Support Vector Data Description (DSVDD) [70] objective enforces feature compactness on the normal source data. Our framework is modular, allowing for flexible component changes, and supports various adaptation strategies, including

statistical and adversarial alignment. Experiments on standard UDA benchmarks [234, 235, 236, 237] for *semantic anomaly detection* [222] demonstrate its effectiveness. Our method achieves state-of-the-art (SoA) performance, even against few-shot adaptation methods.

Contributions. The main contributions of this work can be summarized as follows: (1) The two-fold unsupervised curse of UDA for one-class anomaly detection is formalized, and the induced challenges are outlined. (2) A solution to the two-fold unsupervised problem is proposed by leveraging an intrinsic property of anomalies, i.e., , their scarcity. (3) A UDA method for one-class semantic anomaly detection is introduced, leveraging a Vision Language Model, namely CLIP [232], for dominant cluster identification and alignment using a contrastive strategy. (4) Extensive experiments and analysis are conducted on several benchmarks [234, 235, 236, 237], demonstrating the relevance of the proposed framework under both fully unsupervised and few-shot adaptation settings.

chapter organization. Section 6.2 reviews UAD works under domain shift. Section 6.3 defines the two-fold unsupervised curse, while Section 6.4 and Section 6.5 detail one possible solution for solving it. Section 6.6 and Section 6.7 cover the experiments and limitations of this method. Section 6.8 concludes and outlines future work.

6.2 Related Works: Anomaly detection under domain shift

Unsupervised image anomaly detection is a well-established research area [169, 63, 219, 70, 216, 221, 222] where the aim is to learn a function ζ using a single class corresponding to normal data from the normal-only dataset $\mathcal{D}^n = \{(\mathbf{X}_i, y_i); y_i = 0\}_{i=1}^N$, to classify whether an input image $\mathbf{X}$ is normal ($y = 0$) or not ($y = 1$). This is achieved by optimizing the objective,

$$\min_{\zeta} \mathbb{E}_{(\mathbf{X}_i, y_i)\sim\mathcal{D}^n} \left[\mathcal{L}\left(\zeta(\mathbf{X}_i), y_i = 0\right)\right], \tag{6.1}$$

where $\mathcal{L}$ is a loss enforcing feature compactness as in DSVDD [70] or a reconstruction loss typically used in autoencoders-based methods [216, 221]. Although achieving impressive performance on standard benchmarks, the majority of AD methods [169, 63, 219, 70, 216, 221, 222] overlook the domain gap problem where training and testing data denoted as

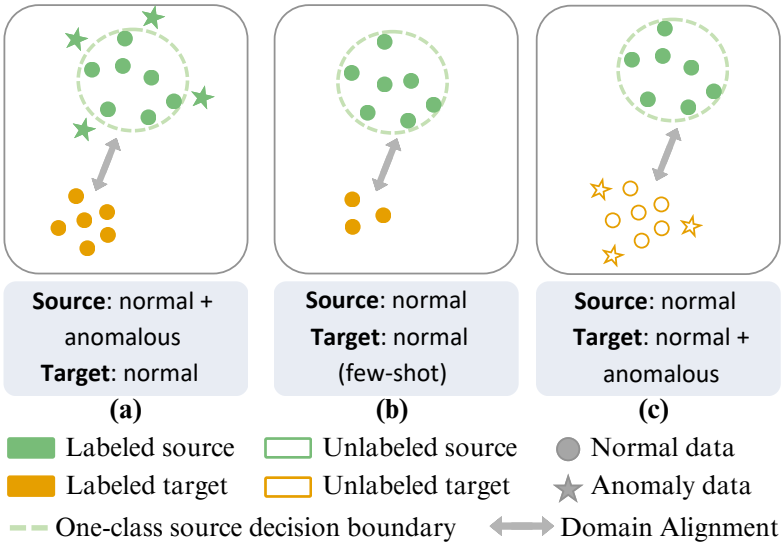


Figure 6.2: **Comparison of our setting with previous works**: **(a)** supervised source anomaly detection with supervised domain adaptation [226], **(b)** unsupervised one-class source anomaly detection with few-shot domain adaptation [228, 230, 229], **(c) our considered setting**: unsupervised one-class source anomaly detection with unsupervised domain adaptation.

$\mathcal{D}^s$ and $\mathcal{D}^t$, respectively, follow different distributions due to uncontrolled variations in the acquisition setting [220, 238]. This domain shift induces, therefore, a significant drop in performance. To solve this issue, a handful of *Domain Generalization* (DG) methods for UAD have been proposed recently [239, 238, 220]. Cohen, Kahana, and Hoshen [239] propose a domain-disentanglement approach that removes predefined nuisance attributes (e.g., pose, lighting) from the source features using contrastive loss, preventing these factors from interfering with the anomaly task, improving the performance on unseen domains. However, without an actual target set, this method requires defining and labeling nuisance factors within the source dataset, which is challenging, as mentioned in their chapter. In [238], multiple source domains are considered for learning domain-invariant features, thereby assuming the availability of diverse large-scale datasets, which is not always guaranteed. To avoid relying on multiple domains during training, a self-supervised strategy is adopted in [220]. Nevertheless, the success of this approach heavily depends on the similarity between the augmented data and target samples. As a result, it necessitates tailoring augmentation techniques to

unseen target datasets, if at all possible. Given its effectiveness, *Domain Adaptation* has also been explored to address the domain shift problem in AD [227, 240, 226, 228, 229]. Those techniques usually adopt a few-shot adaptation paradigm by having access to a limited number of annotated target samples. While these methods offer innovative solutions for aligning source and target normal data, they still rely on costly annotations [219] and are exposed to overfitting risks [231]. This emphasizes the need for a fully unsupervised domain adaptation for UAD. However, addressing this problem remains difficult due to the unsupervised nature of both anomaly detection and domain adaptation, as detailed in the next section.

6.3 The Two-fold Unsupervised Curse

Let us denote as $\mathcal{D}^s = \{(\mathbf{X}_i^s, y_i^s)\}_{i=1}^{N_s}$ a labeled dataset from a given domain called *source* formed by N_s samples, where a sample $\mathbf{X}_i^s \in \mathbb{R}^{h \times w \times c}$ and its associated label $y_i^s \in \{0, 1\}$, $\forall i = \{1, ..., N_s\}$. Let $\mathcal{D}^t$ be a second unlabeled dataset from a different domain, i.e., , *target*, denoted as $\mathcal{D}^t = \{\mathbf{X}_i^t\}_{i=1}^{N_t}$ and formed by N_t samples where $\mathbf{X}_i^t \in \mathbb{R}^{h \times w \times c}$, $\forall i = \{1, ..., N_t\}$. In the following, we assume that $\mathcal{D}^t$ shares the same label space as $\mathcal{D}^s$ and that there exists a domain gap between $\mathcal{D}^s$ and $\mathcal{D}^t$. The goal of Unsupervised Domain Adaptation (UDA) for anomaly detection (whether formulated as a binary or one-class classification problem), is to learn a model $\zeta : \mathbb{R}^{h \times w \times c} \rightarrow \{0, 1\}$ using both $\mathcal{D}^s$ and $\mathcal{D}^t$ that generalizes to the target domain. In other words, it aims at learning a domain invariant feature extractor $f : \mathbb{R}^{h \times w \times c} \mapsto \mathcal{X}$ such that $\zeta = g \circ f$ with $g : \mathcal{X} \mapsto \{0, 1\}$ being the classifier and $\mathcal{X}$ the feature space given by f. This objective is achieved by minimizing the following adaptation upper bound [241],

$$\epsilon^t \leq \epsilon^s + d(f(\mathcal{D}^s), f(\mathcal{D}^t)) + \lambda \,, \tag{6.2}$$

where ϵ^t and ϵ^s are the expected classification errors on the target and source domains, respectively; $d(f(\mathcal{D}^s), f(\mathcal{D}^t))$ estimates the discrepancy between the feature distributions from the two domains, and λ accounts for the joint error on source and target of an ideal detector.

While strategies for minimizing this upper bound are feasible in the context of binary or even multi-class classification [224, 225, 223], the non-availability of anomalous data during training makes it difficult in the context of one-class classification, where $d(f(\mathcal{D}^s), f(\mathcal{D}^t))$ cannot be estimated. In fact, we can only use a subset $\mathcal{D}^{s,n} \subset \mathcal{D}^s$ formed by normal data for training. For that reason, existing works on domain adaptation for one-class anomaly detection [227, 228, 230] revisit the formulation given in Eq (6.2) by slightly simplifying the problem. They pose it as a few-shot domain adaptation setting (instead of a fully unsupervised scenario). This means that they assume having access to a small labeled subset $\mathcal{D}^{t,n} \subset \mathcal{D}^t$ composed of normal samples only. As a result, they reformulate Eq (6.2) as,

$$\epsilon^{t,n} \leq \epsilon^{s,n} + d(f(\mathcal{D}^{s,n}), f(\mathcal{D}^{t,n})) + \lambda \ , \tag{6.3}$$

where $\epsilon^{s,n}$ and $\epsilon^{t,n}$ represent the source and target expected classification errors related to the normal class, respectively, since ϵ^s is not measurable in this context.

Nevertheless, in a fully unsupervised setup, we have access to $\mathcal{D}^t = \mathcal{D}^{t,a} \cup \mathcal{D}^{t,n}$ where $\mathcal{D}^{t,a}$ represents the subset of $\mathcal{D}^t$ formed by anomalies, without any prior information regarding the labels. Hence, directly aligning the feature distributions estimated from the source and target data by approximating $d(f(\mathcal{D}^{s,n}), f(\mathcal{D}^t))$ would lead to obtaining a classification boundary that is completely obsolete for target data, as shown in Figure 6.1 (b). We call this problem the *two-fold unsupervised curse* as it is a consequence of a lack of supervision: (1) in the task of anomaly detection, as it is formulated as a one-class problem where only normal source data are used; and (2) in the task of domain adaptation which is fully unsupervised where only an unlabeled target set is available. Given that the problem is ill-posed, it remains a significant challenge that has not been addressed in the existing UAD literature.

6.4 Rare Anomalies to the Rescue

To tackle the two-fold unsupervised curse described in Section 6.3, we introduce a key assumption and the main hypothesis it entails for enabling UDA for one-class anomaly de-

tection.

Assumption (anomaly scarcity). *For an unlabeled target dataset $\mathcal{D}^t = \mathcal{D}^{t,n} \cup \mathcal{D}^{t,a}$, we assume that the number of anomalous samples is significantly smaller than the number of normal samples, i.e., $|\mathcal{D}^a| \ll |\mathcal{D}^n|$, with $|\cdot|$ refers to the cardinality.*

Hypothesis (dominant cluster existence). *Considering a target unlabeled anomaly detection dataset $\mathcal{D}^t = \mathcal{D}^{t,n} \cup \mathcal{D}^{t,a}$* ***under the anomaly scarcity assumption****, where $\mathcal{D}^{t,n}$ and $\mathcal{D}^{t,a}$ are respectively the normal and abnormal subsets, we hypothesize that there exists a feature extractor $\psi : \mathbb{R}^{h \times w \times c} \to \mathcal{X}$ that generates from $\mathcal{D}^t$ a compact dominant cluster $\mathcal{C} \in \mathcal{X}$ predominated by normal samples.*

The anomaly scarcity assumption often holds as it reflects most real-world scenarios where anomalies are rare compared to normal instances. Our main objective is therefore to find a feature extractor that verifies *the dominant cluster existence hypothesis*. We emphasize that this hypothesis is not granted and remains challenging. Nevertheless, it is a core component of the proposed method discussed in Section 6.5, as it enables the introduction of a novel paradigm to approach UDA for one-class UAD. The paradigm consists of the following steps: (1) finding a feature exactor ψ that can generate a compact dominant cluster of features $\mathcal{C}$ corresponding to normal samples within an unlabeled target dataset $\mathcal{D}^{\text{t}}$, (2) identifying the subset of samples $\tilde{\mathcal{D}}^{\text{t,n}}$ corresponding to this cluster in the feature space of ψ, and (3) aligning the identified subset $\tilde{\mathcal{D}}^{\text{t,n}}$ with the source normal samples $\mathcal{D}^{s,n}$ in the feature space of the source feature extractor f. Formally, we revisit Eq (6.3) as follows,

$$\epsilon^{t,n} \leq \epsilon^{s,n} + d(f(\mathcal{D}^{s,n}), f(\tilde{\mathcal{D}}^{t,n})) + \lambda \,, \tag{6.4}$$

where $\tilde{\mathcal{D}}^{t,n} = \{\mathbf{X}_i^t \mid \psi(\mathbf{X}_i^t) \in \mathcal{C}\}$. Note that ψ can be obtained by focusing on learning compact cross-domain features from which $\mathcal{C}$ can be identified through feature grouping and selection techniques such as clustering or filtering. As such, the proposed paradigm for UDA in one-class UAD lays the foundation for future research, where various technical choices can be explored at each stage.

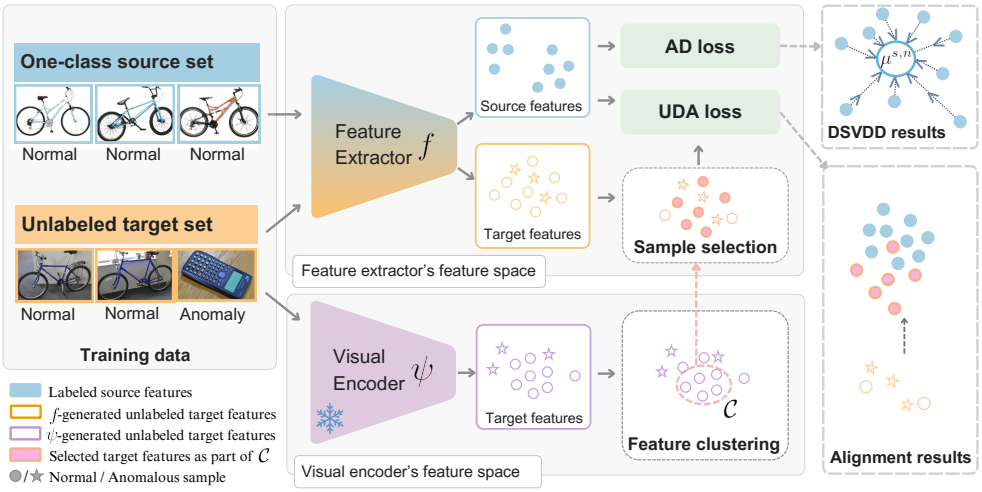


Figure 6.3: **Our Solution**: The top branch uses a trainable feature extractor with a DSVDD objective for one-class source data. The bottom branch clusters the features using a frozen CLIP visual encoder to identify the dominant feature cluster and align it with normal source features. ● are normals and ★ are anomalies .

6.5 Methodology

Building on the assumption and hypothesis formulated in Section 6.4, we present our methodology for introducing UDA to unsupervised visual *semantic* one-class anomaly detection, as one possible solution for tackling the two-fold unsupervised curse under this setting. Note that this choice was motivated by the available baselines [228, 230, 240, 226, 227].

Our approach has two branches. The upper branch depicts a trainable backbone f that learns from both source and target domain data. The source features are optimized using a Deep Support Vector Data Description (DSVDD) objective [70]. The lower branch focuses on visual feature extraction from the unlabeled target domain, through a frozen CLIP visual encoder [232], defined as the ψ feature extractor. Clustering is applied to these visual features to estimate the dominant cluster $\mathcal{C}$. Samples identified within $\mathcal{C}$ in the ψ visual encoder's representation space are then selected within the space of the feature extractor f and then aligned with the normal source features.

Training. Specifically, given source and target image datasets $\mathcal{D}^{s,n}$ and $\mathcal{D}^{t}$, we apply

DSVDD on the source data, enforcing feature compactness by minimizing the radius of a hypersphere to encapsulate the normal source representations. This is done by solving the following optimization problem,

$$\min_{\theta_f} \mathcal{L}_{AD} = \min_{\theta_f} \frac{1}{N_s} \sum_{i=1}^{N_s} \| f(\mathbf{X}_i^s) - \boldsymbol{\mu}^{s,n} \|_2^2, \forall \mathbf{X}_i^s \in \mathcal{D}^{s,n}, \tag{6.5}$$

where $\boldsymbol{\mu}^{s,n}$ is the mean of the source features. For clustering, we use a K-means algorithm. Note that ψ can be f itself in a self-training fashion or any frozen visual encoder such as CLIP [232] or DINO-v2 [242]. The dominant cluster is identified as,

$$\mathcal{C} = \arg\max_{\mathcal{C}_k} |\mathcal{C}_k| \text{ for } k \in \{1, ..., K\}, \tag{6.6}$$

where $|\mathcal{C}_k|$ is the size of the k-th cluster $\mathcal{C}_k$, and K is a hyperparameter defining the number of expected components in the space of $\psi(\mathcal{D}^t)$. When clustering is applied to $f(\mathcal{D}^t)$, the selected features for alignment are $\tilde{\mathcal{D}}^{t,n} = \mathcal{C}$. When clustering is applied to $\psi(\mathcal{D}^t)$, the selected samples are:

$$\tilde{\mathcal{D}}^{t,n} = \{ f(\mathbf{X}_i^t) \mid \psi(\mathbf{X}_i^t) \in \mathcal{C} \} \; \forall \mathbf{X}_i^t \in \mathcal{D}^t \tag{6.7}$$

Alignment between source and target features is achieved using a contrastive strategy, where UDA loss is computed as:

$$\mathcal{L}_{UDA} = \frac{1}{N_s \times |\tilde{\mathcal{D}}^{t,n}|} \sum_{i=1}^{N_s} \sum_{j=1}^{|\tilde{\mathcal{D}}^{t,n}|} \ell_{i,j}, \text{ with } \ell_{i,j} = -\log \frac{\exp\left(\frac{1}{\tau} \cdot \mathsf{sim}\left(f(\mathbf{X}_i^s), f(\mathbf{X}_j^t)\right)\right)}{\sum_{p=1}^{N_t} \mathbb{1}_{[\mathbf{X}_p^t \notin \tilde{\mathcal{D}}^{t,n}]} \exp\left(\frac{1}{\tau} \cdot \mathsf{sim}\left(f(\mathbf{X}_i^s), f(\mathbf{X}_p^t)\right)\right)} \tag{6.8}$$

where $\mathsf{sim}(\cdot, \cdot)$ denotes the cosine similarity, and τ is the temperature.

Finally, the overall loss is:

$$\mathcal{L} = \lambda_1 \, \mathcal{L}_{AD} + \lambda_2 \, \mathcal{L}_{UDA}, \tag{6.9}$$

where λ_1 and λ_2 are hyperparameters for $\mathcal{L}_{AD}$ and $\mathcal{L}_{UDA}$.

Inference. Note that the visual encoder ψ is discarded at inference and only the feature extractor f is used to determine whether the input data is anomalous by calculating whether

it falls inside or outside the hypersphere estimated by the DSVDD model. The pseudo-code of our methodology is given in Algorithm 2.

Algorithm 2: Training Procedure of UDA for UAD

Input : Source dataset $\mathcal{D}^{s,n}$, target dataset $\mathcal{D}^{t}$, cluster count K, epochs N_{epochs}, iterations N_{iter}, encoder f, frozen model ψ

Output: Updated encoder θ_f

$s\_feats\_center \leftarrow \texttt{mean}(f(\mathcal{D}^{s,n}))$; `// Initialize DSVDD center`

for *epoch* $= 1$ **to** N_{epochs} **do** `// Epoch loop`

  for *iter* $= 1$ **to** N_{iter} **do** `// Iteration loop`

    $s\_feats, t\_feats \leftarrow f(s\_batch, t\_batch)$;

    $t\_clip\_feats \leftarrow \psi(t\_batch)$;

    $t\_clusters \leftarrow \texttt{KMeans}(t\_clip\_feats, K)$;

    indices $\leftarrow \texttt{dom\_cluster\_indices}(t\_clusters)$; `// Eq (6,7)`

    $t\_dominant\_feats \leftarrow t\_feats$[indices];

    $t\_non\_dominant\_feats \leftarrow t\_feats$[$\neg$indices];

    pos_pairs $\leftarrow \texttt{pair}(s\_feats, t\_dominant\_feats)$;

    neg_pairs $\leftarrow \texttt{pair}(s\_feats, t\_non\_dominant\_feats)$;

    $\mathcal{L}_{\text{UDA}} \leftarrow \texttt{contrast}$(pos_pairs, neg_pairs) ; `// Eq (8)`

    $\mathcal{L}_{\text{UAD}} \leftarrow \texttt{DSVDD}(s\_feats, s\_feats\_center)$; `// Eq (5)`

    $\mathcal{L} \leftarrow \lambda_1 \mathcal{L}_{\text{UAD}} + \lambda_2 \mathcal{L}_{\text{UDA}}$; `// Eq (9)`

    $\mathcal{L}$.backward();

    $\theta_f \leftarrow \texttt{update}(\theta_f)$;

6.6 Experimental Results

6.6.1 Experimental Setting

This section describes the datasets, the baselines used and the implementation details of our experiments. We report the performance using Area Under the ROC Curve (AUC) using **bold** and underline for the best and second performances, respectively.

Datasets. We evaluate our approach on four standard UDA benchmark datasets, **Office-Home** [234], **Office31** [234],**VisDA** [236], and **PACS** [237]. For the AD task, we adopt a standard one-vs-all protocol, since we focus specifically on the one-class setting, where a

(a) **OfficeHome [235]**

Normal class	**Source only** DSVDD	**Few-shot adaptation** BiOST	TSA	ILDR	IRAD	MsRA	**Ours**	**Unsup. adapt.** **Ours**
Clip Art → Product (C → P)								
Bike	97.48	43.00	69.10	89.90	90.30	94.30	98.34	85.71
Calculat.	83.47	69.00	72.20	84.90	82.20	98.70	97.76	97.70
Drill	81.57	66.40	66.20	75.30	73.00	84.50	74.19	96.64
Hammer	83.32	50.10	77.40	74.70	84.50	80.10	89.55	82.63
Kettle	87.74	63.00	63.10	77.50	75.80	85.50	94.08	89.16
Knives	78.09	48.80	51.90	55.20	63.90	64.40	79.25	76.63
Pan	74.00	57.70	63.70	72.20	76.00	80.50	93.08	91.07
Paperclip	53.04	27.40	74.70	78.70	67.40	79.70	71.18	67.98
Scissors	86.45	56.40	64.70	79.50	68.90	85.50	87.71	88.43
Soda	51.21	50.20	57.40	70.30	53.30	72.40	61.16	92.37
Avg.	77.64	53.20	66.04	75.82	73.53	82.56	<u>84.63</u>	**86.83**
±std	±14.04	±11.65	±7.36	±8.81	±10.24	±9.33	±11.93	±8.66
Product → Clip Art (P → C)								
Bike	82.55	52.70	65.80	83.10	85.70	86.60	82.06	92.99
Calculat.	62.82	65.20	63.40	87.20	79.20	91.90	91.59	89.88
Drill	71.81	47.00	57.10	63.90	71.20	73.50	70.58	77.54
Hammer	68.02	43.70	68.60	60.20	77.00	73.00	84.33	65.42
Kettle	71.85	47.70	61.50	68.80	70.00	73.40	75.38	78.19
Knives	57.22	63.10	57.50	65.30	70.30	73.10	77.74	71.99
Pan	71.44	49.30	63.50	69.30	72.80	80.00	83.72	82.46
Paperclip	26.19	45.10	49.90	69.70	61.80	69.00	67.05	55.93
Scissors	63.42	38.60	70.10	66.20	70.00	72.30	86.35	77.63
Soda	66.82	56.90	55.80	60.20	63.29	59.40	69.08	62.63
Avg.	64.21	50.93	61.32	69.39	72.13	75.22	**78.79**	<u>75.47</u>
±std	±14.22	±8.11	±5.94	±8.55	±6.76	±8.62	±7.74	±11.13

(b) **Office31 [234]**

Normal class	**Source only** DSVDD	**Few-shot adaptation** BiOST	TSA	ILDR	IRAD	MsRA	**Ours**	**Unsup. adapt.** **Ours**
Webcam → Amazon (W → A)								
Backpack	86.48	59.90	76.30	91.90	90.20	95.20	95.40	97.62
Bookcase	35.77	56.60	59.60	78.40	82.20	84.50	76.25	91.16
Bottle	70.00	60.80	66.80	74.50	72.10	74.00	72.48	77.32
Chair	56.92	57.60	63.40	85.30	80.90	87.20	85.50	92.06
Lamp	82.26	50.50	60.90	72.60	67.50	70.00	82.38	81.50
Headpho.	88.91	57.60	75.90	88.90	81.60	92.20	92.53	95.06
Keyboard	79.83	58.20	69.90	88.30	93.20	95.40	95.40	93.36
Laptop	51.79	59.10	63.00	86.20	98.10	99.00	95.63	79.97
Mouse	83.95	65.80	53.40	84.90	79.60	89.90	96.65	92.97
Pen	48.54	68.50	69.10	75.50	71.40	73.90	72.72	71.20
Avg.	68.45	59.46	65.83	82.65	81.68	86.13	<u>86.49</u>	**87.22**
±std	±17.84	±4.70	±6.86	±6.46	±9.37	±9.72	±9.44	±8.48
Amazon → Webcam (A → W)								
Backpack	79.42	47.90	59.00	81.60	91.20	97.50	99.28	97.59
Bookcase	60.68	49.90	72.30	88.90	89.40	93.10	85.23	94.29
Bottle	40.94	66.00	69.80	86.90	95.30	96.20	93.65	94.95
Chair	71.66	67.00	66.20	76.10	90.30	90.10	93.67	99.08
Lamp	94.63	55.50	68.60	73.10	81.30	83.90	94.57	97.61
Headpho.	70.99	68.30	72.40	93.70	91.60	96.00	96.54	96.04
Keyboard	77.90	66.00	76.90	91.10	95.70	98.10	90.62	76.59
Laptop	91.61	62.10	72.20	85.70	97.10	98.20	94.32	97.67
Mouse	72.17	69.10	69.40	82.20	85.40	86.50	96.35	81.41
Pen	44.26	79.10	86.10	97.60	98.90	99.60	97.09	99.99
Avg.	70.43	63.09	71.29	85.69	91.62	<u>93.92</u>	**94.13**	93.52
±std	±16.81	±9.03	±6.66	±7.26	±5.15	±5.11	±3.72	±7.52

Table 6.1: Ten-run average and standard deviation of AUC (%) on the Office datasets [235, 234].

single class is available as normal and the remaining are anomalies. We adopt the experimental protocol of previous DA works [228, 230, 227] to allow for a fair comparison –that is, we show results on ten classes from the ClipArt and Product domains for Office-Home, ten classes from Webcam and Amazon for Office31, and twelve classes from the domains of Computer Aided Designs (CAD) (synthetic objects) and real object photos of VisDA. On PACS, like [220], we consider the Photo domain as source and the remaining three domains as targets.

Baselines. As no other works on UDA for visual semantic UAD were previously introduced, we compare our method with several few-shot adaptation SoA approaches. Specifically, we consider **BiOST** [227] which is a one-shot approach, **TSA** [240], **ILDR** [226], **IRAD** [230], and **MsRA** [228] that are few-shot adaptation methods. Furthermore, we introduce our few-shot adaptation variant (**Ours-Few-shot**), which augments the target domain with normal and pseudo-anomalous samples similar to [243]. This augmentation yields semantically positive and negative pairs [243], useful for the contrastive alignment strategy described in Section 6.5.

Implementation details. In all experiments, the source set has only one-class normal data, while the unlabeled target set includes mostly normals with 10% randomly sampled anomalies. Training uses SGD with a cosine-annealing scheduler, learning rate of 10^{-3}, weight decay of 5×10^{-7}, batch size 256 and λ_1 and λ_2 are set to 1. CLIP-ViT-B32 is the frozen visual encoder ψ for feature clustering. Contrastive loss temperature τ is 0.07. To align with the setting of the baselines [228, 230, 227, 240], ResNet50 is the trainable backbone f, initialized on ImageNet [244]. K-means clustering [245] uses 2, 10, and 5 components for Office, VisDA, and PACS, respectively. Like the baselines, the few-shot adaptation experiments use 10 (Office, PACS) and 100 shots (VisDA) **labeled as normal**, respectively.

Normal class	w/o adaptation			w/ adaptation				
	Zero-shot		Source finetuned DSVDD	Few-shot			Unsup. **Ours**	Super-vised
	R50	CLIP		BiOST	MsRA	Ours		
CAD → Real								
Aero.	41.05	74.97	67.71	36.80	81.56	81.55	84.86	90.91
Bicycle	67.35	90.28	65.12	59.20	68.45	74.58	81.45	81.73
Bus	28.58	42.27	66.01	47.90	68.12	72.26	82.17	72.16
Car	32.48	64.16	78.65	53.80	69.44	82.78	62.76	68.42
Horse	68.81	75.48	67.24	58.00	68.77	80.17	83.52	88.70
Knife	67.78	95.28	62.43	54.10	70.39	71.52	68.82	78.90
Motor.	60.07	82.25	69.45	58.10	65.64	80.16	91.15	83.46
Person	71.69	56.26	42.11	58.70	59.18	51.24	69.68	85.19
Plant	62.47	89.65	57.77	42.10	65.81	71.46	70.58	82.63
Skate.	85.00	91.52	60.70	41.60	61.30	63.17	83.71	83.73
Train	30.13	57.74	54.75	52.40	69.73	60.62	69.98	85.11
Truck	26.05	45.08	62.08	43.10	59.05	73.67	57.84	78.91
Avg.	53.45	72.08	62.84	50.48	67.28	71.93	<u>75.54</u>	**81.65**
±std	±19.57	±17.83	±8.55	±7.55	±5.79	±9.08	±9.80	±6.11

Table 6.2: AUC (%) on the target domain of our UDA anomaly detector on VisDA [236] compared **with various adaptation paradigms** (from zero-shot, i.e., pretrained Visual encoders, few-shot, to supervised, i.e., Oracle).

6.6.2 Comparison against State-of-the-art.

Our method outperforms previous SoA on all benchmarks of our evaluation, as shown in Table 6.1 and Table 6.2. More specifically, our fully unsupervised variant importantly improves upon previous few-shot adaptation SoA on C $\rightarrow$ P and W $\rightarrow$ A of the Office-Home [235] and Office31 [234] datasets. In addition, we observe an improvement of over 10% in the VisDA dataset [236] with the fully unsupervised methodology over previous few-shot adaptation approaches, despite being challenged by the two-fold unsupervised curse. These results highlight the relevance of the proposed method, even in the presence of a large domain gap, as in the case of synthetic CAD images and real-world photos.

In the P $\rightarrow$ C and A $\rightarrow$ W adaptation of the Office datasets, our few-shot adaptation variant also registers SoA performance, closely followed by our model trained under the fully unsupervised setting. These results highlight the flexibility of our framework, which can leverage minimal labeled target data when available but remains highly effective in a fully unsupervised setup.

Furthermore, we compare the performance of our model to two pretrained visual en-

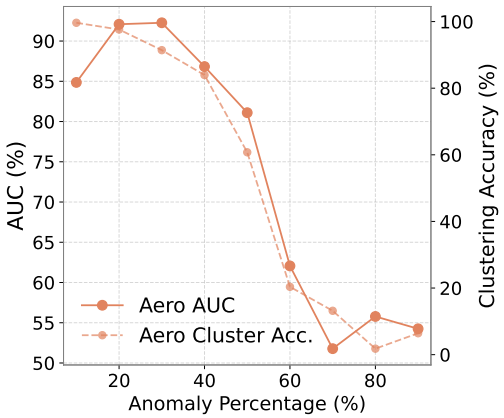


Figure 6.4: Assessing the validity of anomaly scarcity assumption.

coders [232], namely ResNet50 and CLIP-ViT-B32 in Table 6.2. While the CLIP-ViT-B32 architecture achieves an average AUC of 72.08%, our unsupervised method (75.54%) still outperforms it on the VisDA dataset [236]. In contrast, the ResNet50 shows a significantly lower performance, with an average AUC of only 53.45%. These results demonstrate that despite their strong performance, pretrained visual encoders are not specifically tailored for the domain adaptation task; thus, they remain vulnerable to domain shift. Therefore, training domain adaptation-specific models is still necessary to effectively bridge the gap between two given domains.

6.6.3 Additional Experiments

Unless stated otherwise, all the following experiments are performed on VisDA [236].

Anomaly scarcity assumption. To evaluate the impact of anomaly scarcity, we vary the anomaly ratio in the unlabeled target set from 10% to 90% and report our method's performance alongside clustering accuracy in Figure 6.4 for the Aeroplane class from VisDA. The results indicate a strong correlation between AUC performance and clustering accuracy. As the anomaly proportion increases, the AUC gradually degrades, with a drastic drop beyond 50%, where the dominant cluster assumption no longer holds. This is further evidenced by a significant decrease in the clustering accuracy.

Few-shot versus unsupervised adaptation paradigms. The results presented in Table 6.2 compare pretrained visual encoders and source-only detectors with different adaptation paradigms, i.e., few-shot, unsupervised, and supervised (oracle). The source-only finetuned model improves slightly over the pretrained ResNet50 visual encoder [123] but still has lower performance than the adaptation approaches, achieving an average AUC of 62.84%. Among the few-shot methods, our few-shot adaptation variant outperforms BiOST [227] and MsRA [228], achieving the highest AUC of 71.93%, which is comparable to the performance of a pretrained CLIP-ViT-B32 visual encoder. However, our unsupervised adaptation method surpasses all these models, with an average AUC of 75.54% indicating its ability to effectively mitigate domain gaps without relying on labeled target data. This can be explained by the fact that after clustering, our model has access to more representative normal target data than few-shot models, hence better generalizing to the target normal class. On the other hand, the Oracle, which has access to the target labels, achieves the highest performance (81.65%). The small gap between our unsupervised method and the oracle demonstrates the effectiveness of our approach even without supervision.

Ablation on the framework components. Table 6.3 provides the results obtained when each component, namely the use of a adaptation loss, the dominant cluster identification through clustering, the use of an auxiliary visual encoder $\psi(\mathcal{D}^t)$ or the trainable features $f(\mathcal{D}^t)$. The results show that without adaptation, a model trained only on source data generalizes poorly to the target domain with only 62.84%. Direct adaptation of the source and the unlabeled target without clustering leads to inconsistent results, indicating low generalization capabilities to the target domain. Introducing clustering results in a significant performance boost. This can be seen when clustering is applied to the original representations of the feature extractor, as the performance improves by +5.63%, highlighting the importance of identifying the dominant cluster prior to alignment. Note that our method still outperforms

Table 6.3: Ablation on the components of the proposed method.

w/ Adaptation	w/ Clustering	w/ CLIP ψ	**AUC** (%)
✗	✗	✗	62.84±8.55
✓	✗	✗	64.33±6.42
✓	✓	✗	68.47±8.30
✓	✓	✓	**75.54**±9.80

the best few-shot adaptation baseline MsRA [228] (68.47% vs 67.28%) with just clustering and alignment (i.e., **w/o CLIP** where ϕ is self-trained) across all the VisDA classes. Finally, the best results are achieved when all components are combined. This setup boosts the average AUC to 75.54% on all VisDA classes. The substantial performance gains can be attributed to CLIP's rich visual features, which, together with clustering and alignment, help achieve a more robust anomaly detector capable of better handling domain shift. This remains valid when ϕ and ψ are both CLIP (See Table 6.8 in the Appendix).

Clustering methods. We compare different clustering techniques on three UDA benchmarks in Table 6.4. The first observation we make is that any type of clustering improves the performance. K-means and GMM have comparable results, without one clearly and consistently outperforming the other across datasets and adaptation directions. Meanshift clustering offers a performance increase compared to source-only models. However, its performance remains lower than that of the other clustering methods. In our experiments, we chose K-means clustering as it achieves comparable performance to GMM while requiring fewer parameters and simpler optimization. We further investigate the optimal number of K-Means components, as shown in Figure 6.5. The figure indicates that using 8 to 10 components yields the highest performance, with an AUC of approximately 75-76%. Decreasing the number of components would gradually degrade the performance. This suggests that a lower number of clusters may not capture the characteristics of the majority class, leading to inaccurate clustering and thus negatively impacting the generalization of the anomaly detection model across domains.

Similar to [246, 239], it uses a kNN density estimator to detect anomalies. Our results suggest that both methods benefit from the adaptation, as a consistent average improvement of +12.7% and +8.23% is seen across all the twelve classes of VisDA.

Beyond DSVDD by using other AD objectives. To assess whether our alignment approach applies to other unsupervised AD methods, we replace DSVDD with Mean-shifted Contrastive loss (MSC) [247] in Table 6.7. MSC adapts contrastive loss to the one-class

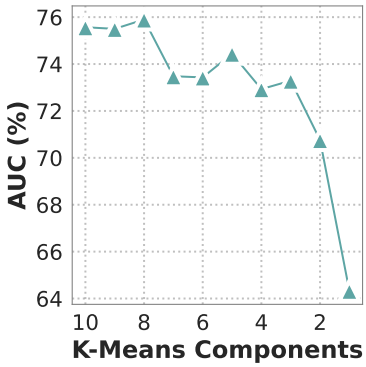


Figure 6.5: K-Means[245] components variation on VisDA.

setting by shifting augmented representations of the normal samples toward the mean of pretrained normal features, preserving their compactness. It can be seen that our unsupervised adaptation improves the performance on average by +8.23 on the VisDA dataset.

Alignment strategies. By comparing several alignment strategies in Table 6.5, we observe that any alignment strategy, in general, improves the performance consistently for all adaptation benchmarks. Contrastive alignment consistently outperforms other adaptation losses, including statistical (MMD) [77] and adversarial (GRL) [248] strategies.

Dataset	**w/o Adapt.** (Src Only)	**w/ Adaptation**			
		KMeans [245]	**GMM** [249]	**MeanShift** [250]	**kNN** [249]
VisDA	62.84±08.55	**75.54**±10.23	72.24±08.81	74.14±07.44	71.65±06.68
A→W	72.57±18.69	94.82±07.52	**96.45**±05.43	87.32±12.53	82.73±10.96
W→A	67.70±18.32	**87.72**±12.63	87.00±13.60	86.68±08.53	83.63±06.04
C→P	77.31±15.13	90.54±14.06	**90.85**±11.18	85.95±10.94	78.67±15.19
P→C	63.88±15.60	70.92±11.36	**76.15**±13.32	73.50±13.49	71.38±11.71

Table 6.4: Clustering ablation. GMM and K-means use 10 components for VisDA and 2 for other datasets. $k = 2$ for VisDA and $k = 1$ for the remaining datasets.

Comparison against domain generalization methods. Table 6.6 compares the results of GNL [220] with DA methods on the PACS dataset [237], with Photo as the source and Art, Cartoon, and Sketch as the target domains. It can be seen that UDA consistently

Dataset	**w/o Adapt.** (Src Only)	**w/ Adaptation**		
		GRL [248]	**MMD** [77]	**Contrastive** [233]
VisDA	62.84±08.55	71.84±10.89	73.12±10.36	**75.54**±9.80
A→W	72.57±18.69	90.43±10.68	86.86±12.97	**94.82**±07.52
W→A	67.70±18.32	83.49±11.04	83.92±09.90	**87.72**±12.63
C→P	77.31±15.13	83.40±12.35	82.10±12.16	**90.54**±14.06
P→C	63.88±15.60	66.78±15.02	67.00±14.41	**70.92**±11.35

Table 6.5: Performance in terms of AUC (%) using different domain adaptation losses on VisDA, Office31 and OfficeHome.

outperforms GNL [220], particularly on Cartoon and Sketch domains. This suggests that, unlike DG methods, which aim to generalize to any unseen domain solely by training on the source domain, UDA can be more effective for semantic UAD since it exposes the model to the target domain during training, even if it is unlabeled.

Adapt. type	**Method**	**Source domain: Photo →**			**Avg. ± std**
		Art	**Cartoon**	**Sketch**	
None	Source only	64.06	64.08	57.35	61.83±3.17
DG	GNL [220]	65.62	67.96	<u>62.39</u>	65.32±2.28
DA	MsRA [228] (Few-shot)	**71.43**	<u>69.89</u>	61.87	67.73±4.19
	Ours (Unsup.)	<u>67.20</u>	**75.35**	**74.04**	**72.20**±3.57

Table 6.6: AUC (%) of Domain Generalization (DG) for anomaly detection, trained ***ONLY*** on the source domain Photo (Ph.) and tested on unseen domains. DA means Domain Adaptation.

f: ResNet50 + ψ: CLIP-ViT-B32					
DSVDD [70]			**MSC [247]**		
ZS	**Src**	**UDA**	**ZS**	**Src**	**UDA**
53.45	62.84	75.54 (**+12.7**↑)	74.39	72.87	81.10 (**+8.23**↑)

Table 6.7: Our UDA approach on two anomaly detection methods [70, 247]. ZS and Src mean Zero-shot and Source only.

Feature extractor. Table 6.8 compares several feature extractor backbones, showcasing the effectiveness of the proposed UDA method against source-only and the robustness of the methodology to the backbone used. Specifically, MobileNet-V2 [251], as the smallest architecture, shows the weakest improvement, while ResNet18 [123] and ResNet50 [123] show better performance. Transformer-based models like CLIP-ViT-B32 [232] show an even higher performance. These results highlight the superiority of transformer models over CNN-based architectures for domain adaptation tasks on challenging datasets like VisDA [236], which comes at the cost of having a larger architecture.

	Our framework backbone composition: ψ: CLIP-ViT-B32 (fixed)			
Adapation setting	ϕ: MobileNet-V2 [251]	ϕ: ResNet18 [123]	ϕ: Resnet50 [123]	ϕ: CLIP-ViT-B32 [232]
Source only (DSVDD)	55.63±06.06	56.80±10.90	62.83±8.60	81.89±17.38
Ours (UDA)	**70.36**±10.00	**72.47**±11.04	**75.54**±11.57	**85.92**±13.12

Table 6.8: AUC performance on VisDA [236] of different trainable feature extractors using our method, in comparison against the source-only-trained model.

Pretrained visual encoders. As we are considering an artificial anomaly detection setting, i.e., in the training datasets, only one object class is considered as "normal" while all other classes are treated as "anomalies", our approach requires visual encoders that can effectively capture global object-level features. Hence, in Figure 6.6b, we compare our unsupervised model using various visual encoders against our source-only model, few-shot baselines (BiOST, MsRA), and an oracle (supervised) model. Among the visual encoders, CLIP [232] and CLIPSeg [252] exhibit the highest consistency and overall performance, with medians ranging between 75% and 80% AUC. SigLIP [253] achieves comparable performance, though with slightly more variability. In contrast, Dino-v2 [242] shows noticeably lower performance, suggesting that its representations may be less effective at capturing global object-level features. As expected, the source-only model performs the worst, while the Oracle model reaches the highest and most stable performance. Compared to the baselines, our model with different visual encoders significantly outperforms the few-shot baselines, which have much lower performance.

AUC	Src Only	Few-shot		Unsupervised (**Ours**) ϕ : R50 (fixed)				
		BiOST	MsRA	**w/o** CLIP	w/ CLIP	w/ SigLIP	w/ DINO-v2	w/ CLIPSeg
Avg	62.84	50.48	67.28	68.47	<u>75.54</u>	72.64	65.71	**76.85**
±std	±8.55	±7.55	±5.79	±8.30	±9.80	±10.05	±13.71	±8.92

Table 6.9: VisDA performance with different visual encoders. **w/o CLIP** means the ϕ is self-trained.

Performance using additional metrics. Table 6.10 presents the anomaly detection performance on the VisDA [236] dataset using additional metrics, including Accuracy (Acc.), Balanced Accuracy (B.acc.), Precision (P.), and Recall (R.). The results shown for our few-shot and unsupervised variants align with those presented in Table 6.2, particularly in terms of B.acc. and R.. Our few-shot variant outperforms MsRA [228], demonstrating superior anomaly detection performance. This improvement may be attributed to using a contrastive alignment strategy, which explicitly maximizes the similarity between normal data in both source and target domains, as opposed to the implicit adversarial-based alignment used in MsRA [228]. The complexity of the VisDA [236] dataset, with its diverse backgrounds and resolutions in the target domain (Real), especially in comparison with other benchmarks, may also be an impacting factor in this performance drop. In contrast, our unsupervised approach benefits from the highest performance, underscoring the benefit of exposing the model to larger amounts of unlabeled data, which may result in better adaptation in complex datasets.

Sensitivity analysis. In Figure 6.6a, we investigate the impact of the two hyper-parameters λ_1 and λ_2 controlling the one-class optimization and domain alignment, respectively. Specifically, we set λ_1 to 1 and vary λ_2 to assess the impact of the adaptation loss. We find that the best performance is reached when both λ_1 and λ_2 are set to 1, indicating that the domain alignment objective is as important as the one-class classification during model optimization.

Qualitative Results: Histograms of anomaly scores. The anomaly score distributions of the four methods (Source-Only, MsRA, Ours-Few-shot, and Ours) tested on all VisDA [236]

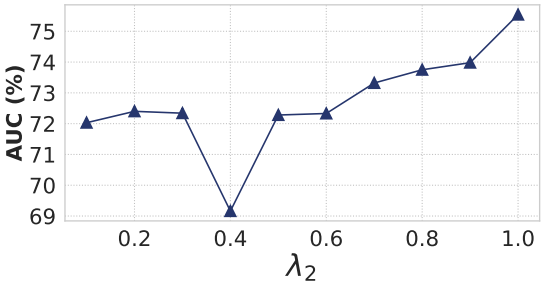
Table 6.10: Anomaly detection performance on the VisDA [236] dataset for the setting f:R50 + ψ:CLIP-ViT-B32 using additional metrics, such as Accuracy (Acc.), Balanced Accuracy (B.acc.), Precision (P), and Recall (R).

Normal class	Source only (DSVDD)				Few-shot (MsRA)				Few-shot (**Ours**)				Unsup. (**Ours**)			
	Acc.	B.acc.	P	R	Acc.	B.acc.	P	R	Acc.	B.acc.	P	R	Acc.	B.acc.	P	R
CAD → Real																
Aeroplane	65.54	63.62	95.66	65.86	62.17	66.09	96.44	61.51	77.28	74.46	97.21	77.75	83.42	77.35	97.35	84.44
Bicycle	56.40	61.36	96.42	55.74	45.20	55.61	95.54	43.81	66.33	67.72	97.17	66.15	72.49	73.61	97.87	72.34
Bus	53.91	62.83	94.90	51.78	56.11	59.71	93.54	55.26	59.82	67.26	95.86	58.04	75.59	75.30	96.59	75.66
Car	69.95	71.61	95.94	69.52	33.57	51.14	90.65	29.14	79.50	75.35	96.02	80.55	73.59	60.21	92.41	76.96
Horse	62.21	63.07	94.74	62.03	34.32	53.02	92.87	30.36	73.62	73.14	96.55	73.73	74.17	75.30	97.06	73.93
Knife	54.61	59.84	95.06	53.68	53.42	64.22	96.46	51.49	70.06	65.96	95.69	70.79	56.87	63.97	96.08	55.60
Motorcycle	63.33	64.39	93.61	63.02	47.28	56.13	91.64	44.73	69.50	72.70	95.92	68.58	79.56	77.61	96.24	80.13
Person	13.57	50.33	91.14	03.68	53.94	57.62	92.21	52.94	36.46	51.49	90.28	32.42	61.90	64.56	94.15	61.19
Plant	55.61	56.08	95.31	55.54	66.65	52.51	94.49	68.55	69.84	66.08	96.70	70.34	58.65	65.43	97.15	57.73
Skateboard	60.32	59.01	97.29	60.43	27.72	54.41	97.47	25.52	64.21	59.97	97.33	64.56	73.23	76.46	98.92	72.97
Train	47.36	53.90	91.53	45.71	48.84	53.48	91.30	47.67	59.23	57.60	92.33	59.64	67.33	64.70	94.04	67.99
Truck	59.72	58.87	93.23	59.92	71.18	52.56	91.23	75.46	66.94	67.28	95.25	66.86	74.84	58.05	91.82	79.07
Avg.	55.28	60.41	94.57	53.91	50.03	56.38	93.65	48.87	66.07	66.58	95.53	65.78	**70.97**	**69.38**	**95.81**	**71.50**
±std	14.41	05.50	01.87	17.10	13.42	04.53	04.76	15.49	11.17	07.23	02.12	12.40	08.18	07.21	02.21	9.13

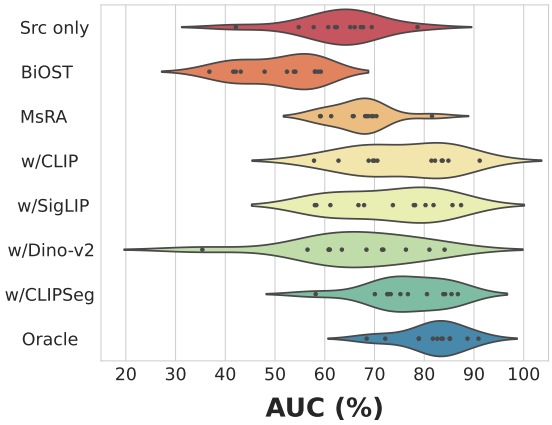
classes are given in Figure 6.7. Overall, our few-shot and unsupervised methods better discriminate normal and anomalous target-domain samples compared to the Source-Only model. The few-shot variant shows a flatter anomaly distribution, likely due to the use of jigsaw-generated pseudo-anomalies, which closely resemble the original target normals. This may have led the model to focus on local changes in actual anomalies, resulting in a broader range of anomaly scores and less emphasis on global features. In contrast, our unsupervised method exhibits a more peaked anomaly distribution. However, the incomplete separation of normal and anomalous scores suggests clustering limitations and highlights the need for filtering or noise removal mechanisms to better identify normal target samples.

6.7 Limitations and Future Work

Our method, presented in Section 6.5, is one possible solution for addressing the problem of UDA for UAD. However, it is worth noting that it was tested in the context of *semantic anomaly detection* [222], adopting a one-vs-all protocol, to facilitate the comparison with the closest baselines, namely [228, 230]. These methods typically require the use of global



(a) Sensitivity analysis on VisDA when $\lambda_1 = 1$ (i.e., , AD objective) and λ_2 varies (i.e., , the UDA objective).



(b) Distribution of the AUC performance across the classes of VisDA of our unsupervised approach when using ϕ as ResNet50 coupled with various visual encoders ψ vs. BiOST [227], MsRA [228], and the Oracle.

Figure 6.6: (a) Sensitivity analysis. (b) AUC comparison for different visual encoders.

features in contrast to standard anomaly detection, where fine-grained representations are usually targeted. For that reason, our method focuses mostly on global representations, while local features would be conceptually more suitable for fine-grained anomaly detection. In future works, we aim to extend our study to fine-grained anomaly detection by exploiting more relevant local representations, such as industrial and medical UAD.

6.8 Conclusion

This work is the first to address unsupervised domain adaptation (UDA) for one-class-based unsupervised anomaly detection (UAD), subject to what we refer to as the two-fold unsupervised curse. To address this ill-posed problem, an inherent property of anomalies, namely, their scarcity, is leveraged. This characteristic allows utilizing clustering, –as one possible solution– for identifying a dominant cluster within the unlabeled target set. Assuming this cluster to be predominantly composed of normal data, a contrastive alignment strategy is then used to align its features with the normal source representations. Extensive experiments on standard UDA benchmarks demonstrate that the proposed method effectively

mitigates the domain gap and enhances anomaly detection performance across different domains, outperforming other supervised adaptation approaches without requiring target annotations. Finding the optimal feature extractor remains an open research question. In future work, we intend to further explore compact representations across domains to improve the proposed domain adaptation framework.

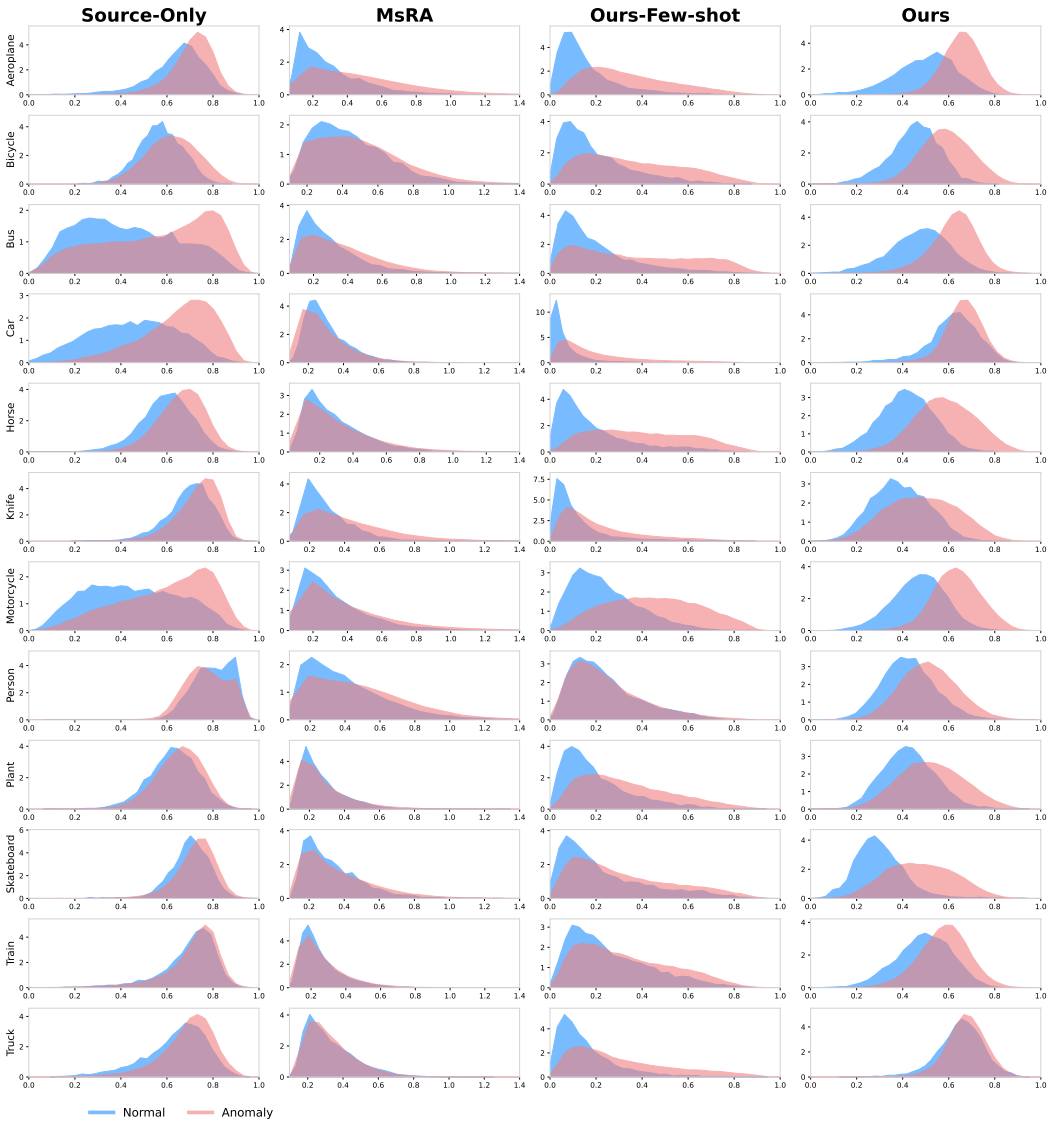


Figure 6.7: Histogram of anomaly scores for all classes of VisDA [236] (x-axis: anomaly score, and y-axis: count).

Chapter 7

Conclusion

This final chapter provides a summary of the main findings from our research contributions in the field of deepfake detection using unsupervised anomaly detection and discusses the future directions of this thesis work.

7.1 Summary

In this thesis, we posited that the lack of generalization observed in current deepfake detectors can be attributed to both forgery-related and forgery-unrelated factors. This guided our investigation towards addressing generalization at two distinct levels: generalization across different deepfake types, as well as generalization across domains.

In our first contribution, UNTAG, presented in Chapter 3, we addressed the problem of forgery-related generalization for image-level deepfake detection. This was done by reformulating the task as an unsupervised anomaly detection (UAD) problem. This formulation enabled training only on genuine faces, which helped treat any deviations from their learned distribution as a deepfake, regardless of their type. This resulted in a type-agnostic deepfake detector using a single, unified solution. However, although this setting eliminated the need for annotated fake training data, it remained challenging due to the absence of anomaly labels and the lack of prior knowledge of the artifact types. This motivated the introduction of a self-supervision mechanism that enhances the detector's sensitivity to artifact-prone re-

gions. Experimental results demonstrated that our UAD-based formulation is capable of detecting various deepfake types, including unseen and stacked forgeries, while being trained only on real faces.

While UNTAG successfully addressed generalization across different deepfake types, it only considered image-level detection, which makes it inapplicable to deepfake videos. Indeed, in real-world scenarios, deepfakes can occur as videos that are fully or partially manipulated in specific frames. This motivated the extension of the UAD formulation to temporal data for ensuring type-agnostic detection for videos. This led to a new task in deepfake detection, referred to as unsupervised temporal deepfake localization. This task naturally aligns with the goals of unsupervised time-series anomaly detection (TSAD), particularly since videos can be interpreted as pixel trajectories evolving over time. Nevertheless, these techniques have never been applied to deepfake detection, which motivated our second contribution, presented in Chapter 4, which consists of an evaluation of existing state-of-the-art multivariate unsupervised TSAD approaches. The aim of this study is to understand the underlying paradigms, strengths, and limitations of these techniques. This assessment helped determine their maturity for real-world applications such as deepfake temporal localization.

Our third contribution, presented in Chapter 5, built on the evaluation study of Chapter 4. It introduced the first fully unsupervised method for localizing manipulated frames, thereby accommodating more realistic scenarios where deepfakes occur as videos that can be partially manipulated or fully manipulated. This approach is also based on the unsupervised anomaly detection (UAD) formulation, where geometric representations (i.e., facial landmarks) are used as time-series input. Since this unsupervised setting is challenging due to the lack of supervision and prior knowledge on artifact types across deepfakes, we introduced a facial-region-focused ensembling strategy that helped focusing on these regions and enhanced detection.

Despite improving the forgery-related generalization capabilities of deepfake detectors for both the image and video data through the UDA formulation, the learned distribution of genuine faces remains sensitive to domain shift. This issue can be attributed to the one-class model's reliance on deep learning architectures. For instance, when this detector is

evaluated on new genuine data from another domain where environmental conditions differ from those seen during training, it may incorrectly classify them as deepfakes. To address this challenge without relying on annotation efforts, we considered integrating unsupervised domain adaptation (UDA) into a generic framework of unsupervised image-based anomaly detection. Despite UDA being well-established in binary and multi-class classification, its integration into unsupervised anomaly detectors is challenging, resulting in what we refer to as the "two-fold unsupervised curse"; a byproduct of the absence of labels in both the source and target domain data. This two-fold unsupervised challenge motivated our fourth contribution presented in Chapter 6.

Finally, our fifth contribution is under preparation and extends our work on UDA for UAD to fine-grained unsupervised image-based anomaly detection. In our earlier work, the UDA benchmarks were object detection datasets. To apply a one-class protocol in this context, we defined one object class as normal and treated all other object classes as anomalies. Despite this protocol being common in semantic anomaly detection, it remains unrealistic, as deepfakes often depict faces with fine-grained and localized inconsistencies. Therefore, in this work, we considered a more challenging setup, aiming to detect instances where anomalies occur locally in images depicting the same object as the one seen during training. Experiments on fine-grained anomaly detection datasets, such as industrial benchmarks [64, 254], have demonstrated promising results, which opens the door for further investigations in the context of unsupervised type-agnostic facial forgery detection.

7.2 Future work

While this thesis has addressed key challenges in deepfake detection, namely their generalization issues, several important research directions remain open for further exploration. These include domain-adaptive unsupervised deepfake detection, explainability, and content-agnostic generalizable detection methods. Investigating these directions would, therefore, improve the robustness, interpretability, and generalization of current deepfake detectors, thereby moving closer to achieving detection systems appropriate for real-world settings.

We outline these prospective avenues below:

7.2.1 Domain-adaptive unsupervised deepfake detection

Although our investigation of UDA for UAD showed promising results on common UDA benchmarks, its direct application to facial deepfakes remains an open challenge. Several factors contribute to this challenge:

First, the lack of fine-grained UAD with domain shift datasets led us to generate synthetic shifts for fine-grained AD industrial benchmarks. Although we achieved competitive results on these industrial datasets, the simulated domain shifts may not accurately reflect real-world shifts, making it difficult to draw conclusions about the transferability of our proposed setting to real-world tasks, such as deepfake detection. In fact, deepfakes are highly diverse and complex, with subtle artifacts, which makes the domain adaptation less straightforward. This highlights the need for collecting both generic and facial images UAD benchmarks that incorporate real-world domain shifts.

Second, despite the UAD formulation enabling type-agnostic deepfake detection, it remains challenging to learn discriminative representations from genuine data without access to labeled anomalies. For instance, our investigations from Chapters 3 and 5 revealed that one-class classification is insufficient in the absence of adequate mechanisms to guide the one-class detector towards artifact-prone regions. This emphasizes the need for more comprehensive representation learning from the normal data, which can be achieved by explicitly modeling multiple sources of prior knowledge to express the notion of normality across various levels of abstraction. More specifically, hierarchical representations can be constructed by integrating low-level cues, such as texture consistency, frequency-domain residuals, and local noise patterns around facial regions, with mid- to high-level semantic information, including facial geometry, temporal dynamics, or identity features. This multi-level feature encoding would express a more robust notion of normality, where deviations introduced by deepfakes could be identified as outliers, at any level of abstraction, without requiring explicit anomaly labels.

7.2.2 Vision-language models for explainable deepfake detection

Most deepfake detectors rely on binary classification and often lack interpretability regarding how their predictions are made. Recent advances in Vision-Language Models (VLMs) have enabled the development of detectors capable of generating human-understandable textual explanations that highlight inconsistencies associated with deepfakes. However, most current research [255, 256, 257] remains in the early stages, primarily exploring the zero-shot capabilities of VLMs for both detection and explanation. This opens up significant opportunities for fine-tuning and for investigating which artifacts are truly detectable by these models, as well as how VLMs can model and explain the concept of normality.

7.2.3 Extension to content-agnostic forgery detection

A natural and plausible extension to the works of this thesis involves generalization beyond facial content. This is particularly important, as digital forgeries are not limited to faces. Generic forgeries often include (1) synthetic images targeting various fields such as science, medicine, geography, or history (e.g., fake satellite [258] or medical [259] images), and (2) fully synthetic videos, such as body reenactments [260] where individuals appear to perform actions they never did. Similar to facial deepfakes, these forgeries pose serious risks of misinformation and misuse, such as manipulating public opinion or distorting historical facts. Detecting such content is therefore essential not only to mitigate these threats but also to enable applications such as copyright protection, where verifying content authenticity is required.

Nevertheless, adapting the methods proposed in this thesis to this broader setting introduces new challenges, primarily due to the lack of consistent spatial or semantic structures in general forged content. Unlike faces, generic manipulated scenes do not have predefined regions that can serve as priors to guide detection. Artifacts may appear anywhere in the image and affect any object, increasing the complexity of the task. One way to address this is by learning hierarchical representations that capture a robust notion of normality at multiple levels of abstraction as described earlier.

However, since normal data in this context does not belong to a single class, as is the case with faces, this would require extending anomaly detection beyond the traditional one-class setting to an unsupervised multiclass anomaly detection formulation, as explored in [261, 262].

References

[1] Hany Farid. *Photo forensics*. MIT press, 2016.

[2] Eric Kee, James F O'brien, and Hany Farid. "Exposing Photo Manipulation from Shading and Shadows." In: *ACM Trans. Graph.* 33.5 (2014), pp. 165–1.

[3] Eric Kee and Hany Farid. "Exposing digital forgeries from 3-D lighting environments". In: *2010 IEEE International Workshop on Information Forensics and Security*. IEEE. 2010, pp. 1–6.

[4] Eric Kee, Micah K Johnson, and Hany Farid. "Digital image authentication from JPEG headers". In: *IEEE transactions on information forensics and security* 6.3 (2011), pp. 1066–1075.

[5] Xunyu Pan, Xing Zhang, and Siwei Lyu. "Exposing image splicing with inconsistent local noise variances". In: *2012 IEEE International conference on computational photography (ICCP)*. IEEE. 2012, pp. 1–10.

[6] I Perov et al. *DeepFaceLab: A simple, flexible and extensible face swapping framework*. 2018.

[7] Deepfakes. *Faceswap*. Accessed: 07-Mar-2025. 2017. URL: https://faceswap.dev/.

[8] Víctor Murillo-Ligorred et al. "Knowledge, integration and scope of deepfakes in arts education: The development of critical thinking in postgraduate students in primary education and master's degree in secondary education". In: *Education Sciences* 13.11 (2023), p. 1073.

[9] Kristin Houser. *Kendrick Lamar deepfakes made by South Park creators' studio*. Accessed: 07-Mar-2025. 2022. URL: https://www.freethink.com/culture/kendrick-lamar-deepfakes.

[10] *Free AI video generator - create AI videos in 140 languages*. Accessed: 07-Mar-2025. 2025. URL: https://www.synthesia.io/.

[11] Zack Sharf. *Lucasfilm hired the YouTuber who used deepfakes to tweak Luke Skywalker 'Mandalorian' VFX*. Accessed: 07-Mar-2025. 2021. URL: https://www.indiewire.com/features/general/lucasfilm-hires-deepfake-youtuber-mandalorian-skywalker-vfx-1234653720/.

[12] Jane Wakefield. *Deepfake presidents used in Russia-Ukraine war*. https://www.bbc.com/news/technology-60780142. Accessed: 13-07-2022.

[13] Sarah Cahlan. *How misinformation helped spark an attempted coup in Gabon*. https://www.washingtonpost.com/politics/2020/02/13/how-sick-president-suspect-video-helped-sparked-an-attempted-coup-gabon/. Accessed: 2022-02-09.

[14] Dan Milmo. "Company worker in Hong Kong pays out £20m in deepfake video call scam". In: *The Guardian* (2024). Accessed: 07-Mar-2025.

[15] Olina Banerji. "Why schools need to wake up to the threat of AI 'deepfakes' and bullying". In: *Education Week* (2024), pp. 14–17.

[16] Catherine Han et al. "Characterizing the MrDeepFakes Sexual Deepfake Marketplace". In: *Proceedings of the USENIX Security Symposium*. To appear. USENIX, 2025.

[17] Chris Rohlfs. "Generalization in neural networks: A broad survey". In: *Neurocomputing* 611 (2025), p. 128701. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2024.128701. URL: https://www.sciencedirect.com/science/article/pii/S0925231224014723.

[18] Dat Nguyen et al. “LAA-Net: Localized Artifact Attention Network for Quality-Agnostic and Generalizable Deepfake Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 17395–17405.

[19] Kaede Shiohara and Toshihiko Yamasaki. “Detecting Deepfakes with Self-Blended Images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18720–18729.

[20] Lingzhi Li et al. “Face x-ray for more general face forgery detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 5001–5010.

[21] François Chollet. “Xception: Deep learning with depthwise separable convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258.

[22] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[23] Andreas Rossler et al. “Faceforensics++: Learning to detect manipulated facial images”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1–11.

[24] Yuezun Li et al. “Celeb-df: A large-scale challenging dataset for deepfake forensics”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3207–3216.

[25] Brian Dolhansky et al. *The DeepFake Detection Challenge Dataset*. 2020. arXiv: 2006.07397 [cs.CV].

[26] Yinan He et al. “Forgerynet: A versatile benchmark for comprehensive forgery analysis”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4360–4369.

[27] Tianchen Zhao et al. “Learning self-consistency for deepfake detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15023–15033.

[28] Liang Chen et al. “Self-supervised Learning of Adversarial Example: Towards Good Generalizations for Deepfake Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18710–18719.

[29] Yuyang Sun et al. “Generalized Deepfakes Detection with Reconstructed-Blended Images and Multi-scale Feature Reconstruction Network”. In: *2024 IEEE International Joint Conference on Biometrics (IJCB)*. 2024, pp. 1–11. DOI: 10.1109/IJCB62174.2024.10744491.

[30] Jiaran Zhou et al. “Freqblender: Enhancing deepfake detection by blending frequency knowledge”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 44965–44988.

[31] Ruben Tolosana et al. “Deepfakes and beyond: A survey of face manipulation and fake detection”. In: *Information Fusion* 64 (2020), pp. 131–148.

[32] Nesryne Mejri, Konstantinos Papadopoulos, and Djamila Aouada. “Leveraging High-Frequency Components for Deepfake Detection”. In: *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*. 2021, pp. 1–6. DOI: 10.1109/MMSP53017.2021.9733606.

[33] Sowmen Das et al. “Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3776–3785.

[34] Joquin Quinonero-Candela et al. *Dataset Shift in Machine Learning*. en. London, England: MIT Press, 2022. ISBN: 9780262545877.

[35] Nesryne Mejri, Enjie Ghorbel, and Djamila Aouada. “Untag: Learning generic features for unsupervised type-agnostic deepfake detection”. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, pp. 1–5.

[36] Daichi Zhang et al. “Detecting Deepfake Videos with Temporal Dropout 3DCNN.” In: *IJCAI*. 2021, pp. 1288–1294.

[37] Zhixi Cai et al. “Marlin: Masked autoencoder for facial video representation learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 1493–1504.

[38] Yuting Xu et al. “Tall: Thumbnail layout for deepfake video detection”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 22658–22668.

[39] Daichi Zhang et al. “Learning natural consistency representation for face forgery video detection”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 407–424.

[40] Zhiyuan Yan et al. “Generalizing deepfake video detection with plug-and-play: Video-level blending and spatiotemporal adapter tuning”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 12615–12625.

[41] Du Tran et al. “Learning Spatiotemporal Features with 3D Convolutional Networks”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 4489–4497. DOI: 10.1109/ICCV.2015.510.

[42] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[43] Nesryne Mejri et al. “Unsupervised anomaly detection in time-series: An extensive evaluation and analysis of state-of-the-art methods”. In: *Expert Systems with Applications* 256 (2024), p. 124922. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2024.124922. URL: https://www.sciencedirect.com/science/article/pii/S0957417424017895.

[44] Nesryne Mejri et al. “Facial Region-Based Ensembling for Unsupervised Temporal Deepfake Localization”. In: *2024 IEEE International Conference on Multimedia and Expo (ICME)*. 2024, pp. 1–6. DOI: 10.1109/ICME57554.2024.10688329.

[45] Virginia de Sa. “Learning Classification with Unlabeled Data”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Cowan, G. Tesauro, and J. Alspector. Vol. 6. Morgan-Kaufmann, 1993. URL: https://proceedings.neurips.cc/paper_files/paper/1993/file/e0ec453e28e061cc58ac43f91dc2f3f0-Paper.pdf.

[46] Jie Gui et al. “A survey on self-supervised learning: Algorithms, applications, and future trends”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[47] Longlong Jing and Yingli Tian. “Self-supervised visual feature learning with deep neural networks: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.11 (2020), pp. 4037–4058.

[48] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. “Unsupervised representation learning by predicting image rotations”. In: *arXiv preprint arXiv:1803.07728* (2018).

[49] Richard Zhang, Phillip Isola, and Alexei A Efros. “Colorful image colorization”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer. 2016, pp. 649–666.

[50] Mehdi Noroozi and Paolo Favaro. “Unsupervised learning of visual representations by solving jigsaw puzzles”. In: *European conference on computer vision*. Springer. 2016, pp. 69–84.

[51] Hanwen Liang et al. “Self-supervised spatiotemporal representation learning by exploiting video continuity”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 36. 2. 2022, pp. 1564–1573.

[52] Simon Jenni, Givi Meishvili, and Paolo Favaro. “Video representation learning by recognizing temporal transformations”. In: *European conference on computer vision*. Springer. 2020, pp. 425–442.

[53] Di Yang et al. “Self-supervised video representation learning via latent time navigation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 3. 2023, pp. 3118–3126.

[54] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[55] Ishan Misra and Laurens van der Maaten. “Self-supervised learning of pretext-invariant representations”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 6707–6717.

[56] Kaiming He et al. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[57] Xinlei Chen et al. “Improved Baselines with Momentum Contrastive Learning”. In: *arXiv preprint arXiv:2003.04297* (2020).

[58] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. “Colorization as a proxy task for visual understanding”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6874–6883.

[59] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[60] Alexandros Haliassos et al. “Leveraging Real Talking Faces via Self-Supervision for Robust Forgery Detection”. In: *arXiv preprint arXiv:2201.07131* (2022).

[61] Alexandros Haliassos et al. “Lips don’t lie: A generalisable and robust approach to face forgery detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5039–5049.

[62] Sheldon Fung et al. “DeepfakeUCL: Deepfake Detection via Unsupervised Contrastive Learning”. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2021, pp. 1–8.

[63] Lukas Ruff et al. “A Unifying Review of Deep and Shallow Anomaly Detection”. In: *Proceedings of the IEEE* 109.5 (2021), pp. 756–795. DOI: 10.1109/JPROC.2021.3052449.

[64] Paul Bergmann et al. “MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9592–9600.

[65] Kyle Hundman et al. “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding”. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 387–395.

[66] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[67] Ingo Steinwart, Don Hush, and Clint Scovel. “A Classification Framework for Anomaly Detection.” In: *Journal of Machine Learning Research* 6.2 (2005).

[68] Bernhard Schölkopf et al. “Support vector method for novelty detection”. In: *Advances in neural information processing systems* 12 (1999).

[69] David MJ Tax and Robert PW Duin. “Support vector data description”. In: *Machine learning* 54 (2004), pp. 45–66.

[70] Lukas Ruff et al. “Deep one-class classification”. In: *International conference on machine learning*. PMLR. 2018, pp. 4393–4402.

[71] Michael Kirby and Lawrence Sirovich. “Application of the Karhunen-Loeve procedure for the characterization of human faces”. In: *IEEE Transactions on Pattern analysis and Machine intelligence* 12.1 (1990), pp. 103–108.

[72] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.

[73] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).

[74] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. “Maximum Likelihood from Incomplete Data via the EM Algorithm”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22. URL: https://www.jstor.org/stable/2984875.

[75] Emanuel Parzen. “On Estimation of a Probability Density Function and Mode”. In: *The Annals of Mathematical Statistics* 33.3 (1962), pp. 1065–1076. DOI: 10.1214/aoms/1177704472. URL: https://doi.org/10.1214/aoms/1177704472.

[76] Youshan Zhang. “A survey of unsupervised domain adaptation for visual recognition”. In: *arXiv preprint arXiv:2112.06745* (2021).

[77] Arthur Gretton et al. “A kernel method for the two-sample-problem”. In: *Advances in neural information processing systems* 19 (2006).

[78] Mingsheng Long et al. “Learning transferable features with deep adaptation networks”. In: *International conference on machine learning*. PMLR. 2015, pp. 97–105.

[79] Mingsheng Long et al. “Deep transfer learning with joint adaptation networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 2208–2217.

[80] Baochen Sun, Jiashi Feng, and Kate Saenko. “Return of frustratingly easy domain adaptation”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016.

[81] Yaroslav Ganin et al. “Domain-adversarial training of neural networks”. In: *Journal of machine learning research* 17.59 (2016), pp. 1–35.

[82] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *arXiv preprint arXiv:1511.06434* (2015).

[83] Scientific Foresight Unit. *Tackling deepfakes in European policy*. Tech. rep. Accessed: 2022-02-09. Tech. rep., European Parliamentary Research Service. https://www. europarl . . ., 2021.

[84] Darius Afchar et al. “Mesonet: a compact facial video forgery detection network”. In: *2018 IEEE international workshop on information forensics and security (WIFS)*. IEEE. 2018, pp. 1–7.

[85] Nicolo Bonettini et al. “Video face manipulation detection through ensemble of cnns”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 5012–5019.

[86] Ruben Tolosana et al. “Deepfakes evolution: Analysis of facial regions and fake detection performance”. In: *International Conference on Pattern Recognition*. Springer. 2021, pp. 442–456.

[87] Hao Lin et al. “Improved Xception with Dual Attention Mechanism and Feature Fusion for Face Forgery Detection”. In: *arXiv preprint arXiv:2109.14136* (2021).

[88] Beijing Chen et al. “Locally GAN-generated face detection based on an improved Xception”. In: *Information Sciences* 572 (2021), pp. 16–28. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2021.05.006.

[89] Ning Yu, Larry S Davis, and Mario Fritz. “Attributing fake images to gans: Learning and analyzing gan fingerprints”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 7556–7566.

[90] Francesco Marra et al. “Do gans leave artificial fingerprints?” In: *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE. 2019, pp. 506–511.

[91] Lucy Chai et al. “What makes fake images detectable? understanding properties that generalize”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 103–120.

[92] Davide Cozzolino et al. “Forensictransfer: Weakly-supervised domain adaptation for forgery detection”. In: *arXiv preprint arXiv:1812.02510* (2018).

[93] Sheng-Yu Wang et al. “Cnn-generated images are surprisingly easy to spot... for now”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8695–8704.

[94] Nesryne Mejri, Konstantinos Papadopoulos, and Djamila Aouada. “Leveraging High-Frequency Components for Deepfake Detection”. In: *IEEE Workshop on Multimedia Signal Processing*. 2021.

[95] Sitong Liu et al. “Block shuffling learning for Deepfake Detection”. In: *arXiv preprint arXiv:2202.02819* (2022).

[96] Hasam Khalid and Simon S Woo. “OC-FakeDect: Classifying Deepfakes Using One-class Variational Autoencoder. In 2020 IEEE”. In: *CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 2794–2803.

[97] Yunjey Choi et al. “Stargan v2: Diverse image synthesis for multiple domains”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8188–8197.

[98] Tero Karras et al. “Analyzing and improving the image quality of stylegan”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8110–8119.

[99] Tero Karras et al. “Alias-free generative adversarial networks”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[100] Hanqing Zhao et al. “Multi-attentional deepfake detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2185–2194.

[101] Yonghyun Jeong et al. “FrePGAN: Robust Deepfake Detection Using Frequency-level Perturbations”. In: *arXiv preprint arXiv:2202.03347* (2022).

[102] Davide Cozzolino Giovanni Poggi Luisa Verdoliva. “Extracting camera-based fingerprints for video forensics”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2019.

[103] Yuezun Li and Siwei Lyu. “Exposing deepfake videos by detecting face warping artifacts”. In: *arXiv preprint arXiv:1811.00656* (2018).

[104] Shuang Ma et al. “Contrastive Learning of Global and Local Video Representations”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[105] Jiameng Pu et al. “Deepfake videos in the wild: Analysis and detection”. In: *Proceedings of the Web Conference 2021*. 2021, pp. 981–992.

[106] Lukas Ruff. *Deep one-class learning: a deep learning approach to anomaly detection*. Technische Universitaet Berlin (Germany), 2021.

[107] Izhak Golan and Ran El-Yaniv. “Deep anomaly detection using geometric transformations”. In: *Advances in neural information processing systems* 31 (2018).

[108] Chun-Liang Li et al. “Cutpaste: Self-supervised learning for anomaly detection and localization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9664–9674.

[109] Kihyuk Sohn et al. “Learning and evaluating representations for deep one-class classification”. In: *arXiv preprint arXiv:2011.02578* (2020).

[110] Todd K Moon. “The expectation-maximization algorithm”. In: *IEEE Signal processing magazine* 13.6 (1996), pp. 47–60.

[111] Yuchen Luo et al. “Generalizing face forgery detection with high-frequency features”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16317–16326.

[112] Lingzhi Li et al. “Faceshifter: Towards high fidelity and occlusion aware face swapping”. In: *arXiv preprint arXiv:1912.13457* (2019).

[113] Justus Thies, Michael Zollhöfer, and Matthias Nießner. “Deferred neural rendering: Image synthesis using neural textures”. In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–12.

[114] Justus Thies et al. “Face2face: Real-time face capture and reenactment of rgb videos”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2387–2395.

[115] Lele Chen et al. “Hierarchical cross-modal talking face generation with dynamic pixel-wise loss”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7832–7841.

[116] Aliaksandr Siarohin et al. “First order motion model for image animation”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[117] Ohad Fried et al. “Text-based editing of talking-head video”. In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–14.

[118] Youngjoo Jo and Jongyoul Park. “Sc-fegan: Face editing generative adversarial network with user’s sketch and color”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1745–1753.

[119] Cheng-Han Lee et al. “Maskgan: Towards diverse and interactive facial image manipulation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5549–5558.

[120] Yuval Nirkin, Yosi Keller, and Tal Hassner. “Fsgan: Subject agnostic face swapping and reenactment”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 7184–7193.

[121] Yu Deng et al. “Disentangled and controllable face image generation via 3d imitative-contrastive learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5154–5163.

[122] Shichao Dong et al. “Implicit Identity Leakage: The Stumbling Block to Improving Deepfake Detection Generalization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[123] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[124] Camillo Lugaresi et al. “Mediapipe: A framework for building perception pipelines”. In: *arXiv preprint arXiv:1906.08172* (2019).

[125] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019).

[126] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[127] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

[128] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[129] Tero Karras et al. “Training Generative Adversarial Networks with Limited Data”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[130] Bendong Zhao et al. “Convolutional neural networks for time series classification”. In: *Journal of Systems Engineering and Electronics* 28.1 (2017), pp. 162–169.

[131] Ganapathy Mahalakshmi, S Sridevi, and Shyamsundar Rajaram. “A survey on forecasting of time series data”. In: *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE’16)*. IEEE. 2016, pp. 1–8.

[132] Ane Blázquez-García et al. “A review on outlier/anomaly detection in time series data”. In: *ACM Computing Surveys (CSUR)* 54.3 (2021), pp. 1–33.

[133] Renjie Wu and Eamonn Keogh. “Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress”. In: *IEEE Transactions on Knowledge and Data Engineering* (2021).

[134] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. “Anomaly detection for IoT time-series data: A survey”. In: *IEEE Internet of Things Journal* 7.7 (2019), pp. 6481–6494.

[135] R Devaki, V Kathiresan, and S Gunasekaran. “Credit card fraud detection using time series analysis”. In: *International Journal of Computer Applications* 3 (2014), pp. 8–10.

[136] Eamonn Keogh et al. “Finding unusual medical time-series subsequences: Algorithms and applications”. In: *IEEE Transactions on Information Technology in Biomedicine* 10.3 (2006), pp. 429–439.

[137] Konstantina Kourou et al. “Machine learning applications in cancer prognosis and prediction”. In: *Computational and Structural Biotechnology Journal* 13 (2015), pp. 8–17. ISSN: 2001-0370. DOI: https://doi.org/10.1016/j.csbj.2014.11.005. URL: https://www.sciencedirect.com/science/article/pii/S2001037014000464.

[138] Koosha Golmohammadi and Osmar R Zaiane. “Sentiment analysis on Twitter to improve time series contextual anomaly detection for detecting stock market manipulation”. In: *Big Data Analytics and Knowledge Discovery: 19th International Conference, DaWaK 2017, Lyon, France, August 28–31, 2017, Proceedings 19*. Springer. 2017, pp. 327–342.

[139] Jehn-Ruey Jiang, Jian-Bin Kao, and Yu-Lin Li. “Semi-supervised time series anomaly detection based on statistics and deep learning”. In: *Applied Sciences* 11.15 (2021), p. 6698.

[140] Avital Oliver et al. “Realistic evaluation of deep semi-supervised learning algorithms”. In: *Advances in neural information processing systems* 31 (2018).

[141] Yingjie Zhou et al. “Feature Encoding With Autoencoders for Weakly Supervised Anomaly Detection”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.6 (2022), pp. 2454–2465. DOI: 10.1109/TNNLS.2021.3086137.

[142] Ya Su et al. “Robust anomaly detection for multivariate time series through stochastic recurrent neural network”. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2828–2837.

[143] Vikas Garg and Adam T Kalai. “Supervising Unsupervised Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_

files/paper/2018/file/72e6d3238361fe70f22fb0ac624a7072-Paper.pdf.

[144] Nesime Tatbul et al. “Precision and recall for time series”. In: *Advances in neural information processing systems* 31 (2018).

[145] Kukjin Choi et al. “Deep learning for anomaly detection in time-series data: review, analysis, and guidelines”. In: *IEEE Access* 9 (2021), pp. 120043–120065.

[146] Kwei-Herng Lai et al. “Revisiting time series outlier detection: Definitions and benchmarks”. In: *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 1)*. 2021.

[147] Siwon Kim et al. “Towards a Rigorous Evaluation of Time-Series Anomaly Detection”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36 (2022), pp. 7194–7201. DOI: 10.1609/aaai.v36i7.20680.

[148] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. “Anomaly Detection in Time Series: A Comprehensive Evaluation”. In: *Proceedings of the VLDB Endowment (PVLDB)* 15.9 (2022), pp. 1779–1797. DOI: 10.14778/3538598.3538602.

[149] Julien Audibert et al. “Do deep neural networks contribute to multivariate time series anomaly detection?” In: *Pattern Recognition* 132 (2022), p. 108945.

[150] Zahra Zamanzadeh Darban et al. “Deep learning for time series anomaly detection: A survey”. In: *arXiv preprint arXiv:2211.05244* (2022).

[151] Fanxing Liu et al. “Fedtadbench: Federated time-series anomaly detection benchmark”. In: *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE. 2022, pp. 303–310.

[152] John Paparrizos et al. “TSB-UAD: an end-to-end benchmark suite for univariate time-series anomaly detection”. In: *Proceedings of the VLDB Endowment* 15.8 (2022), pp. 1697–1711.

[153] Ya Liu et al. “Unsupervised Deep Learning for IoT Time Series”. In: *IEEE Internet of Things Journal* 10.16 (2023), pp. 14285–14306. DOI: 10.1109/JIOT.2023.3243391.

[154] Zhenyu Zhong et al. “A Survey of Time Series Anomaly Detection Methods in the AIOps Domain”. In: *arXiv preprint arXiv:2308.00393* (2023).

[155] Mohammed Ayalew Belay et al. “Unsupervised anomaly detection for iot-based multivariate time series: Existing solutions, performance analysis and future directions”. In: *Sensors* 23.5 (2023), p. 2844.

[156] Baihong Jin et al. “A one-class support vector machine calibration method for time series change point detection”. In: *2019 IEEE International conference on prognostics and health management (ICPHM)*. IEEE. 2019, pp. 1–5.

[157] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest”. In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[158] Xi Chen et al. “Autoregressive-model-based methods for online time series prediction with missing values: an experimental evaluation”. In: *arXiv preprint arXiv:1908.06729* (2019).

[159] Asrul H Yaacob et al. “Arima based network anomaly detection”. In: *2010 Second International Conference on Communication Software and Networks*. IEEE. 2010, pp. 205–209.

[160] Ailin Deng and Bryan Hooi. “Graph neural network-based anomaly detection in multivariate time series”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 5. 2021, pp. 4027–4035.

[161] Haowen Xu et al. “Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications”. In: *Proceedings of the 2018 world wide web conference*. 2018, pp. 187–196.

[162] Julien Audibert et al. “Usad: Unsupervised anomaly detection on multivariate time series”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 3395–3404.

[163] Pankaj Malhotra et al. “Long Short Term Memory Networks for Anomaly Detection in Time Series.” In: *ESANN*. Vol. 2015. 2015, p. 89.

[164] Dan Li et al. “MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks”. In: *Artificial Neural Networks and Machine Learning–ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part IV*. Springer. 2019, pp. 703–716.

[165] Yufeng Yu et al. “Time series outlier detection based on sliding window prediction”. In: *Mathematical problems in Engineering* 2014 (2014).

[166] Koosha Golmohammadi and Osmar R Zaiane. “Time series contextual anomaly detection for detecting market manipulation in stock market”. In: *2015 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE. 2015, pp. 1–10.

[167] Clive WJ Granger and Mark W Watson. “Time series and spectral methods in econometrics”. In: *Handbook of econometrics* 2 (1984), pp. 979–1022.

[168] Yue Lu et al. “Matrix profile XXIV: scaling time series anomaly detection to trillions of datapoints and ultra-fast arriving data streams”. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 1173–1182.

[169] Songqiao Han et al. “Adbench: Anomaly detection benchmark”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 32142–32159.

[170] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest”. In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[171] Kun-Lun Li et al. “Improving one-class SVM for anomaly detection”. In: *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*. Vol. 5. IEEE. 2003, pp. 3077–3081.

[172] Yanxin Wang, Johnny Wong, and Andrew Miner. “Anomaly intrusion detection using one class SVM”. In: *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.* IEEE. 2004, pp. 358–364.

[173] Junshui Ma and Simon Perkins. “Time-series novelty detection using one-class support vector machines”. In: *Proceedings of the International Joint Conference on Neural Networks, 2003.* Vol. 3. IEEE. 2003, pp. 1741–1745.

[174] Chengqiang Huang et al. “Time series anomaly detection for trustworthy services in cloud computing systems”. In: *IEEE Transactions on Big Data* 8.1 (2017), pp. 60–72.

[175] Lifeng Shen, Zhuocong Li, and James Kwok. “Timeseries anomaly detection using temporal hierarchical one-class network”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 13016–13026.

[176] Markus M Breunig et al. “LOF: identifying density-based local outliers”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.

[177] Jian Tang et al. “Enhancing effectiveness of outlier detections for low density patterns”. In: *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings 6*. Springer. 2002, pp. 535–548.

[178] Takehisa Yairi et al. “A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction”. In: *IEEE Transactions on Aerospace and Electronic Systems* 53.3 (2017), pp. 1384–1401.

[179] Michael R Lindstrom, Hyuntae Jung, and Denis Larocque. “Functional Kernel Density Estimation: Point and Fourier Approaches to Time Series Anomaly Detection”. In: *Entropy* 22.12 (2020), p. 1363.

[180] Bo Zong et al. “Deep autoencoding gaussian mixture model for unsupervised anomaly detection”. In: *International conference on learning representations*. 2018.

[181] Renato Baptista et al. “Deformation-based abnormal motion detection using 3d skeletons”. In: *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE. 2018, pp. 1–6.

[182] Diab M Diab et al. “Anomaly detection using dynamic time warping”. In: *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. IEEE. 2019, pp. 193–198.

[183] Seif-Eddine Benkabou, Khalid Benabdeslem, and Bruno Canitia. “Unsupervised outlier detection for time series by entropy and dynamic time warping”. In: *Knowledge and Information Systems* 54.2 (2018), pp. 463–486.

[184] Renato Baptista et al. “Home self-training: Visual feedback for assisting physical activity for stroke survivors”. In: *Computer methods and programs in biomedicine* 176 (2019), pp. 111–120.

[185] Donald J Berndt and James Clifford. “Using dynamic time warping to find patterns in time series.” In: *KDD workshop*. Vol. 10. 16. Seattle, WA, USA: 1994, pp. 359–370.

[186] Rob J Hyndman, Earo Wang, and Nikolay Laptev. “Large-scale unusual time series detection”. In: *2015 IEEE international conference on data mining workshop (ICDMW)*. IEEE. 2015, pp. 1616–1619.

[187] Yongjun Jin et al. “Anomaly detection in time series via robust PCA”. In: *2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE. 2017, pp. 352–355.

[188] Xudong Wang, Luis Miranda-Moreno, and Lijun Sun. “Hankel-structured tensor robust PCA for multivariate traffic time series anomaly detection”. In: *arXiv preprint arXiv:2110.04352* (2021).

[189] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Nonlinear component analysis as a kernel eigenvalue problem”. In: *Neural computation* 10.5 (1998), pp. 1299–1319.

[190] Erkki Oja. “Simplified neuron model as a principal component analyzer”. In: *Journal of mathematical biology* 15 (1982), pp. 267–273.

[191] Kishore K Reddy et al. “Anomaly detection and fault disambiguation in large flight data: A multi-modal deep auto-encoder approach”. In: *Annual Conference of the PHM Society*. Vol. 8. 1. 2016.

[192] Shuyu Lin et al. “Anomaly detection for time series using vae-lstm hybrid model”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Ieee. 2020, pp. 4322–4326.

[193] H Zare Moayedi and MA Masnadi-Shirazi. “Arima model for network traffic prediction and anomaly detection”. In: *2008 international symposium on information technology*. Vol. 4. IEEE. 2008, pp. 1–6.

[194] Hang Zhao et al. “Multivariate time-series anomaly detection via graph attention network”. In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2020, pp. 841–850.

[195] Jonathan Goh et al. “A dataset to support research in the design of secure water treatment systems”. In: *Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11*. Springer. 2017, pp. 88–99.

[196] Peggy O’Neill et al. “The NASA soil moisture active passive (SMAP) mission: Overview”. In: *2010 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2010, pp. 3236–3239.

[197] Ihssan Tinawi. “Machine learning for time series anomaly detection”. PhD thesis. Massachusetts Institute of Technology, 2019.

[198] Kyle Hundman et al. “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding”. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 387–395.

[199] Mohamed Adel Musallam et al. “Spacecraft recognition leveraging knowledge of space environment: simulator, dataset, competition design and analysis”. In: *2021 IEEE International Conference on Image Processing Challenges (ICIPC)*. IEEE. 2021, pp. 11–15.

[200] Lu Zhang et al. “A real-time intrusion detection system based on oc-svm for containerized applications”. In: *2021 IEEE 24th International Conference on Computational Science and Engineering (CSE)*. IEEE. 2021, pp. 138–145.

[201] Md Amran Siddiqui et al. “Detecting cyber attacks using anomaly detection with explanations and expert feedback”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 2872–2876.

[202] Komal Chugh et al. “Not made for each other-audio-visual dissonance-based deepfake detection and localization”. In: *Proc. of ACMMM*. 2020, pp. 439–447.

[203] Zhixi Cai et al. “Do You Really Mean That? Content Driven Audio-Visual Deepfake Dataset and Multimodal Method for Temporal Forgery Localization”. In: *DICTA*. 2022, pp. 1–10. DOI: 10.1109/DICTA56598.2022.10034605.

[204] Rui Zhang et al. “UMMAFormer: A Universal Multimodal-adaptive Transformer Framework for Temporal Forgery Localization”. In: *Proc. of ACMMM*. 2023, pp. 8749–8759.

[205] Yuxin Zhang et al. “Unsupervised deep anomaly detection for multi-sensor time-series signals”. In: *IEEE TKDE* (2021).

[206] Zekun Sun et al. “Improving the efficiency and robustness of deepfakes detection through precise geometric features”. In: *Proc. of CVPR*. 2021, pp. 3609–3618.

[207] Zihan Liu, Hanyi Wang, and Shilin Wang. “Cross-Domain Local Characteristic Enhanced Deepfake Video Detection”. In: *Proc. of ACCV*. 2022, pp. 3412–3429.

[208] Ya Su et al. “Robust anomaly detection for multivariate time series through stochastic recurrent neural network”. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2828–2837.

[209] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. “TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data”. In: *Proceedings of VLDB* 15.6 (2022), pp. 1201–1214.

[210] Alban Siffer et al. “Anomaly detection in streams with extreme value theory”. In: *Proc. of ACM SIGKDD*. 2017, pp. 1067–1075.

[211] Andrés Prados-Torreblanca, José M Buenaposada, and Luis Baumela. “Shape Preserving Facial Landmarks with Graph Attention Networks”. In: *BMVC*. BMVA Press, 2022.

[212] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).

[213] Chaoqin Huang et al. “Adapting visual-language models for generalizable anomaly detection in medical images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 11375–11385.

[214] Jinan Bao et al. “BMAD: Benchmarks for Medical Anomaly Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2024, pp. 4042–4053.

[215] Han Sun et al. “Continuous Test-time Domain Adaptation for Efficient Fault Detection under Evolving Operating Conditions”. In: *arXiv preprint arXiv:2406.06607* (2024).

[216] Hanqiu Deng and Xingyu Li. “Anomaly detection via reverse distillation from one-class embedding”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 9737–9746.

[217] Marius Beul et al. “Fast Autonomous Flight in Warehouses for Inventory Applications”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3121–3128. DOI: 10.1109/LRA.2018.2849833.

[218] Jiaqi Liu et al. “Deep industrial image anomaly detection: A survey”. In: *Machine Intelligence Research* 21.1 (2024), pp. 104–135.

[219] Romain Hermary et al. “Removing Geometric Bias in One-Class Anomaly Detection with Adaptive Feature Perturbation”. In: *IEEE Winter Conference on Applications of Computer Vision, WACV 2025, Tucson, AZ, USA, February 28-March 4, 2025*. IEEE, 2025.

[220] Tri Cao, Jiawen Zhu, and Guansong Pang. “Anomaly detection under distribution shift”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 6511–6523.

[221] Tran Dinh Tien et al. “Revisiting Reverse Distillation for Anomaly Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 24511–24520.

[222] Luc PJ Sträter et al. “Generalad: Anomaly detection across domains by attending to distorted features”. In: *arXiv preprint arXiv:2407.12427* (2024).

[223] Garrett Wilson and Diane J Cook. “A survey of unsupervised deep domain adaptation”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.5 (2020), pp. 1–46.

[224] Tarun Kalluri, Sreyas Ravichandran, and Manmohan Chandraker. “UDA-Bench: Revisiting Common Assumptions in Unsupervised Domain Adaptation Using a Standardized Framework”. In: *ECCV* (2024).

[225] Inder Pal Singh et al. “Discriminator-free unsupervised domain adaptation for multi-label image classification”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 3936–3945.

[226] Atsutoshi Kumagai, Tomoharu Iwata, and Yasuhiro Fujiwara. “Transfer Anomaly Detection by Inferring Latent Domain Representations”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/7895fc13088ee37f511913bac71fa66f-Paper.pdf.

[227] Tomer Cohen and Lior Wolf. “Bidirectional one-shot unsupervised domain mapping”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1784–1792.

[228] Shuang Li et al. “End-to-End Transferable Anomaly Detection via Multi-Spectral Cross-Domain Representation Alignment”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.12 (2023), pp. 12194–12207. DOI: 10.1109/TKDE.2021.3118111.

[229] Yachun Li et al. “Few-shot one-class domain adaptation based on frequency for iris presentation attack detection”. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 2480–2484.

[230] Ziyi Yang, Iman Soltani, and Eric Darve. “Anomaly detection with domain adaptation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2958–2967.

[231] Yisheng Song et al. “A Comprehensive Survey of Few-shot Learning: Evolution, Applications, Challenges, and Opportunities”. In: *ACM Comput. Surv.* 55.13s (2023). ISSN: 0360-0300. DOI: 10.1145/3582688. URL: https://doi.org/10.1145/3582688.

[232] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[233] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).

[234] Kate Saenko et al. “Adapting visual category models to new domains”. In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer. 2010, pp. 213–226.

[235] Hemanth Venkateswara et al. “Deep hashing network for unsupervised domain adaptation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5018–5027.

[236] Xingchao Peng et al. “Visda: The visual domain adaptation challenge”. In: *arXiv preprint arXiv:1710.06924* (2017).

[237] Da Li et al. “Deeper, broader and artier domain generalization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5542–5550.

[238] João Carvalho et al. “Invariant anomaly detection under distribution shifts: a causal perspective”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[239] Niv Cohen, Jonathan Kahana, and Yedid Hoshen. “Red PANDA: Disambiguating image anomaly detection by removing nuisance factors”. In: *The Eleventh International Conference on Learning Representations*. 2023.

[240] Shuang Li et al. “Transferable semantic augmentation for domain adaptation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 11516–11525.

[241] Shai Ben-David et al. “Analysis of Representations for Domain Adaptation”. In: *Advances in Neural Information Processing Systems*. Ed. by B. Schölkopf, J. Platt, and T. Hoffman. Vol. 19. MIT Press, 2006. URL: https://proceedings.neurips.cc/paper_files/paper/2006/file/b1b0432ceafb0ce714426e9114852ac7-Paper.pdf.

[242] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2023.

[243] Songwei Ge et al. “Robust contrastive learning using negative samples with diminished semantics”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 27356–27368.

[244] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[245] John A Hartigan and Manchek A Wong. “Algorithm AS 136: A k-means clustering algorithm”. In: *Journal of the royal statistical society. series c (applied statistics)* 28.1 (1979), pp. 100–108.

[246] Tal Reiss et al. “PANDA: Adapting Pretrained Features for Anomaly Detection and Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 2806–2814.

[247] Tal Reiss and Yedid Hoshen. “Mean-shifted contrastive loss for anomaly detection”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 2. 2023, pp. 2155–2162.

[248] Yaroslav Ganin and Victor Lempitsky. “Unsupervised domain adaptation by backpropagation”. In: *International conference on machine learning*. PMLR. 2015, pp. 1180–1189.

[249] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[250] Dorin Comaniciu and Peter Meer. “Mean shift: A robust approach toward feature space analysis”. In: *IEEE Transactions on pattern analysis and machine intelligence* 24.5 (2002), pp. 603–619.

[251] Mark Sandler et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[252] Timo Lüddecke and Alexander Ecker. “Image segmentation using text and image prompts”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 7086–7096.

[253] Xiaohua Zhai et al. “Sigmoid loss for language image pre-training”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 11975–11986.

[254] Yang Zou et al. “SPot-the-Difference Self-Supervised Pre-training for Anomaly Detection and Segmentation”. In: *arXiv preprint arXiv:2207.14315* (2022).

[255] Yue Zhang et al. “Common sense reasoning for deepfake detection”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 399–415.

[256] Niki M Foteinopoulou, Enjie Ghorbel, and Djamila Aouada. “A hitchhiker’s guide to fine-grained face forgery detection using common sense reasoning”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 2943–2976.

[257] Ke Sun et al. “Towards General Visual-Linguistic Face Forgery Detection”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. 2025, pp. 19576–19586.

[258] Hong-Shuo Chen et al. “Geo-DefakeHop: High-Performance Geographic Fake Image Detection”. In: *APSIPA Transactions on Signal and Information Processing* 13.3 (2024). DOI: 10.1561/116.00000072. URL: http://dx.doi.org/10.1561/116.00000072.

[259] Siddharth Solaiyappan and Yuxin Wen. “Machine learning based medical image deepfake detection: A comparative study”. In: *Machine Learning with Applications* 8 (2022), p. 100298.

[260] Jiazhi Guan et al. “TALK-Act: Enhance Textural-Awareness for 2D Speaking Avatar Reenactment with Diffusion Model”. In: *SIGGRAPH Asia 2024 Conference Papers*. SA ’24. Tokyo, Japan: Association for Computing Machinery, 2024. ISBN: 9798400711312. DOI: 10.1145/3680528.3687571. URL: https://doi.org/10.1145/3680528.3687571.

[261] Zhiyuan You et al. “A unified model for multi-class anomaly detection”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 4571–4584.

[262] Jia Guo et al. “Dinomaly: The less is more philosophy in multi-class unsupervised anomaly detection”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 20405–20415.