Identifying Fraudulent Identity Documents by Analyzing Imprinted Guilloche Patterns

Musab Al-Ghadi[1], Tanmoy Mondal[2], Zuheng Ming[3], Petra Gomez-Krämer[1], Mickaël Coustaty[1], Nicolas Sidere[1], and Jean-Christophe Burie[1]

[1] L3i, University of La Rochelle, France
{musab.alghadi, petra.gomez, mickael.coustaty, nicolas.sidere, jean-christophe.burie}@univ-lr.fr
[2] IMT Atlantique, Brest, France
tanmoy.mondal@imt-atlantique.fr
[3] University Sorbonne Paris Nord, Paris, France
zuheng.ming@univ-paris13.fr

Abstract. Identity document (ID) verification is crucial in fostering trust in the digital realm, especially with the increasing shift of transactions to online platforms. Our research, building upon our previous work [1], delves deeper into ID verification by focusing on guilloche patterns. We present two innovative ID verification models leveraging contrastive and adversarial learning. These models enhance guilloche pattern detection, offering new insights into identifying counterfeit IDs. Each approach comprises two main components: (i) guilloche pattern recognition and feature generation using a convolutional neural network (CNN), and (ii) precise classification of input data as authentic or forged. We evaluate our models extensively on the MIDV and FMIDV datasets, achieving accuracy and F1-score results ranging from 68-92% and 75-100%, respectively. Our study, incorporating contrastive and adversarial learning, contributes significantly to the ongoing discourse on ID verification, specifically in analyzing guilloche patterns.

Keywords: Information security · Authentication · Identity documents · Contrastive learning · Adversary learning · Guilloche patterns.

1 Introduction

With the increase in internet usage and digitization, digital service providers are required to remotely perform several administrative processes e.g. user registration, identity verification, etc. Hence, it is required to automatically accept and verify administrative documents, such as *identity cards*, *passports*, *driving licenses*, etc. We need a secure, straightforward digital identification system to verify digitized administrative documents. Across the world, governments have been trying different measures to secure various administrative documents of their citizens from counterfeits and fraud. Several sophisticated security features are incorporated to combat any forgery style on IDs. These features make it theoretically difficult, if not impossible, to produce counterfeit or forged IDs [2]. *Holograms*, *Guilloche pattern*, *Optically Variable Ink*, *Anti-Scan Pattern*, *Watermark*, *Micro Type*, *Encoded Data*, and *Invisible Fluorescent Fibers* are examples of embedded security features in the IDs. Specifically, the *Guilloche* is defined as a geometrical pattern of computer-generated fine lines that are interlaced to form a unique shape [3]. This pattern is an important feature to confirm the authenticity of a given ID. This may be achieved directly by checking out the conformity of the guilloche pattern in the background of an ID (e.g. passport of the person) and its similarity to the guilloche pattern of an authentic (real) version of the same country (e.g. passport of France). There exist several works on the topic of fraud, anomaly detection, and verification of IDs [4], [5], [6], [7] but no works exist which aim at the technique of authenticating IDs, based on the guilloche pattern.

In our paper, we introduce two models for detecting anomalies. These models use contrastive and adversarial learning frameworks.

Contrastive learning is a framework that helps to learn a useful representation by using relations between samples. It helps to identify common attributes between data classes as well as the attributes that differentiate one data class from another [8], [9]. The embedding spaces of similar samples are positioned close to each other while dissimilar ones are positioned far away from each other.

Whereas, adversarial learning is a brute force supervised learning procedure where many adversarial examples are fed into the model which is explicitly labeled as fake samples. There exist many applications of machine learning that are adversarial. The primary goal is to distinguish instances that are *real* from those that are *fake*. Some of such applications are privacy-concerning [10], data hiding [11], and forgery detection [8]. Both of the proposed models, i.e., contrastive learning-based and adversarial learning-based models are designed in a manner to read the entire ID and to recognize the *guilloche* pattern to check its similarity against the pattern of an authentic ID of the same country.

The contribution of this paper can be summarized as follows: (i) *Novelty*: we have proposed two novel architectures (based on contrastive and adversarial learning) to detect forgery in IDs by using *guilloche* patterns. For the contrastive learning-based model, our network incorporates multiple loss functions which inherently help to improve the performance. Furthermore, for the adversarial learning-based model, we have proposed a new way of combining adversarial constraints in the network by incorporating the relevant losses. We also proposed the weighted combination of various losses for these two proposed models. (ii) *Reproducibility*: The code is made publicly available so that the results can be reproduced from the original data set.

The rest of this paper is organized as follows: In Section 2, we present some of the most relevant state-of-the-art works. Section 3 provides the details of the forgery detection models we propose. Section 4 details the experimental settings of our models. Experimental results and discussion are drawn in Section 5, while Section 6 shows the comparative study and discussion. The conclusions are given in Section 7.

Abbreviations

This section includes Table 1 that concisely defines all abbreviations employed throughout the manuscript, thereby fostering comprehension for readers unaccustomed to the specialized language employed.

2 Related Works

In this section, we explore some of the most relevant state-of-the-art works in five categories as follows.

2.1 Forgery Detection Approaches for IDs Authentication

The work in [1] focuses on guilloche patterns in ID. Two forgery detection models, CFD (Contrastive-based forgery Detection) and FsAFD (Fake-Sample-Adversary-based forgery Detection), are proposed. Both models utilize Siamese Neural Networks and aim to validate the authenticity of IDs by comparing guilloche patterns. CFD employs contrastive learning, while FsAFD utilizes adversarial learning. The research emphasizes the significance of guilloche pattern analysis for ID verification in digital environments, enhancing user trust through efficient forgery detection. In [12] the authors proposed a passport verification approach

Table 1: The abbreviations

Abbreviation	Meaning
ID	Identity document
ContFD	Contrastive forgery detection model
AdvFD	Adversary-based forgery detection model
FFT	Fast Fourier transform
LBP	Local binary pattern
SSIM	Structural similarity
ContFD-e2e	End-to-end learning scheme of contrastive forgery detection model
ContFD-dual	Dual learning scheme of contrastive forgery detection model
ContFD-FT	Fine-tuning learning scheme of contrastive forgery detection model
ContFD-e2e-SW	End-to-end learning scheme of contrastive forgery detection model with sub-weights
ContFD-e2e-FW	End-to-end learning scheme of contrastive forgery detection model with full-weights
TAR	True acceptance rate
FRR	False rejection rate
FAR	False acceptance rate
ROC	Receiver operating characteristics
AUC	Area under ROC curve
TP	True positive
FP	False Positive
FN	False negative
SOTA	State-of-the-art
GFLOPS	Giga-Floating-point operations per second

based on the detection of periodic patterns (i.e., logos) that are printed the Russian passports. The presence or absence of the periodic patterns on a given passport is studied through k peaks of the FFT to discriminate between a genuine and a fake passport. In [13], an authentication approach for IDs, based on the conformity of visual features and patterns is proposed. The approach is based on generating a visual descriptor called a *grid color-connected components descriptor*. These descriptors are generated from a set of visual features that are relevant enough to the color-connected components of the processed ID. The similarity between the descriptors of a genuine ID and a query ID is measured to decide whether the query ID is genuine or forged. In [14], the authors proposed a specific classifier to verify the authenticity and legitimacy of IDs. The classifier module started by extracting local and global features like gray-scale histograms, hue and saturation differences, structural similarity scores, and histograms of oriented gradients from the given ID. Then, these features are fed into the support vector machine (SVM) and random forest (RF) classifiers to test if the document is genuine or forged. In [15], the authors proposed two steps to verify the ID. The first step uses the oriented fast and rotated brief (ORB) method to localize the security features like seal, signature, and stamp on the processed ID. In contrast, the second step uses optical character recognition and LBP to extract significant features from the processed ID.

Another solution for ID authentication was designed in [5]. Here, two CNN models called *Siamese* and *Triplet* are adapted to design a technique for ID verification. The role of these models is to extract feature vectors of a pair of IDs and then the similarity between these vectors is measured to decide the genuineness of the IDs. The authors of [16] used *Siamese*, *Triplet*, and *PeleeNet* CNN models to design a verification approach for Spanish IDs. The approach performed a recurrent comparison between two textured background blocks; one block from the genuine ID and the other from the counterfeit ID. The difference between the two processed blocks is learned iteratively with an attention model into specific zones in the ID background.

However, these models are not end-to-end learning models; as they are concerned with specific regions of the ID to analyze and not whole the ID. The work in [17] aimed to enhance the detection capabilities of forensic document examiners in combating transnational document fraud. A profiling method is developed, leveraging visual characteristics of digitized images of fraudulent identity documents in the Interstate Database of Fraudulent Identity Documents (BIDIF) in Switzerland. The approach involves analyzing general and specific characteristics, comparing document numbers, and applying a systematic method for series detection. The work in [18] focused on the application of a generalized and transversal framework, known as the Transversal model, to develop forensic intelligence processes for the detection of modus operandi (M.O.) actions related to false identity documents. The study utilized image processing techniques to profile and compare visual features in datasets comprising 439 seized documents from various Swiss jurisdictions. The Transversal model employs feature selection, extraction, and similarity analysis. Two novel works are proposed for ID verification based on hologram detection [6, 7]. In [6], the approach is based on the shape and the color analysis thanks to the pixel properties to extract the hologram for a given ID and decide the presence of the hologram or not. The method of [7] used the LBP descriptor to represent the features of holograms in the ID and then recognized all hologram patterns to determine whether the ID is genuine. In Tables 2 and 3, we listed the characteristics, novelty, and pros and cons of various SOTA techniques.

Table 2: Characteristics of forgery detection approaches for ID authentication.

Method	**Target application**	**Target security object**	**Processed dataset/s**	**Used method/s**	**The performance**
[1]	Forgery detection IDs	on Guilloche	❐ MIDV2020 ❐ FMIDV2022	Contrastive and Adversarial deep learning	❐ Accuracy $< 65\%$ ❐ Precision $< 65\%$ ❐ F1-score $< 70\%$ ❐ AUC $< 57\%$
[12]	ID verification	Periodic patterns like holograms, logos	Russian passports	FFT	Accuracy (in avg.) $= 34\%$
[13]	ID verification	Specific logo in the ID	Private dataset (Italian, French ID, and French residence card)	❐ Grid-3CD descriptor ❐ Combination of image features ❐ SVM	Accuracy $> 85\%$
[14]	ID verification	Face and visual color analysis in the ID.	Colombian ID documents	❐ Combination of global and local image features ❐ SVM and RF	❐ Accuracy $> 97\%$ ❐ F1-score $> 90\%$

[15]	ID verification	Forgery detection on text, hologram, and stamps/seals	Azerbaijani passport images (MIDV-500 dataset)	❒ LBP ❒ ORB ❒ Fast from Accelerated Segment Test (FAST) ❒ CNN	❒ Accuracy $< 96\%$ ❒ Precision $< 93\%$ ❒ Specificity $< 90\%$
[5]	ID verification	Photo background and visual specific pattern	Private dataset of French IDs	Siamese and triplet CNN	❒ FRR $= 1.5\%$ ❒ FAR $= 3.2\%$
[16]	ID and Banknote verification	Textured zones in the document	❒ Private dataset of Spanish IDs ❒ Banknotes	❒ Recurrent comparator network	❒ AUC ranges $90 - 98\%$
[17]	Forgery detection on Text IDs		BIDIF	Inspection verification for visual characteristics like text mutations, irregularities in visual patterns, etc.	❒ TP $= 70\%$ ❒ FN $= 24\%$ ❒ FP $= 7\%$
[18]	Forgery detection on Text IDs		Private dataset (supplied by various police jurisdictions throughout Switzerland)	❒ Image processing techniques to profile and compare visual features in false identity documents ❒ Transversal model	❒ FP $0.05 - 0.30$ ❒ AUC $> 97\%$
[6]	Hologram detection for ID verification	Hologram	❒ MIDV-500 ❒ Private dataset	Shape and color analysis of the hologram pixels	❒ Precision $= 97\%$ ❒ Recall $= 92\%$ ❒ F1-score $= 94.6\%$

[7]	Hologram detection for ID and currency banknote verification	Hologram	❐ Private dataset of French passports ❐ Banknotes	Multi LBP model	❐ Precision 38-99% for hologram 1 ❐ Recall 6-53% for hologram 1 ❐ Precision 42-55% for hologram 2 ❐ Recall 5-8% for hologram 2

Table 3: Novelty, pros & cons of forgery detection approaches for ID authentication.

Method	**Novelty**	**Pros**	**Cons**
[1]	Proposed two forgery detection models for ID verification based on guilloche patterns	Introducing of the FMIDV dataset.	❐ Exhibit unsatisfactory performance in terms of accuracy, F1-score, and AUC.
[12]	❐ Introduces a peak-matching algorithm for comparing the periodic peaks in the FFT magnitude against a known pattern	❐ Flexibility for various periodic patterns ❐ Low complexity	❐ Dependency on knowing pattern nature ❐ Sensitivity to cropping accuracy ❐ Experimental testing on Russian citizen passport images is mentioned only
[13]	❐ Introduces a new visual descriptor called Grid-3CD for pattern comparison in ID verification. ❐ Demonstrates its effectiveness in ID verification through two strategies: • unsupervised, based on a distance measure • supervised, utilizing a one-class SVM	❐ Incorporation of color and spatial information. ❐ The dual approach provides flexibility in addressing different verification scenarios. ❐ Achieves an average accuracy of about 90% in ID verification.	❐ Dependency on document class definition: the second scenario requires defining verification zones for each document class. ❐ Limited dataset information

[14]	❐ Proposes a comprehensive pipeline for ID acquisition and verification, addressing real-life challenges in capturing document images with smartphones. ❐ The use of deep learning, specifically the UNETS architecture, for background removal.	❐ Combination of global and local features. ❐ Deep learning for semantic segmentation.	❐ The paper primarily focuses on a case study involving Colombian ID documents. ❐ The success of the template matching approach is sensitive to variations in document header appearance.
[15]	❐ System's capability to identify both forged text and manipulated images. ❐ The use of sliding window operations in the CNN training process.	❐ Process complex datasets and learn from various document features thanks to sliding windows.	❐ Trained only on Azerbaijani passport images in MIDV-500. ❐ Assumes that the uploaded documents are of high quality.
[5]	❐ Introduces a deep-learning-based framework for ID verification. ❐ Allows for the learning of generic similarity functions, reducing the need for re-training when dealing with new types of ID.	❐ Transferable learning: the Siamese and triplet models enable the learning of features that are transferable to new, unseen scenarios without requiring extensive re-training.	❐ Dependency on pre-alignment. ❐ Focuses on visual-level document verification, and its performance may be influenced by factors like image quality or lighting conditions.
[16]	❐ Examines different regions of security texture backgrounds to detect counterfeit documents produced by the scan-printing operation.	❐ End-to-end solution for detecting counterfeit documents, focusing on the lack of background details produced by the scan-printing operation. ❐ Introducing a new counterfeit document dataset.	❐ Dependency on image quality. ❐ The recurrent comparator architecture with attention mechanisms is computationally expensive.

[17]	Introduced a novel method for comparing and profiling fraudulent identity documents (FID) based on the visual characteristics of digitized images.	❒ Increased detection of series and links. ❒ Applicability across document types and forgery categories.	Dependency on Visual Characteristics.
[18]	Offering a generalized and transversal framework for developing forensic intelligence processes.	Versatility: the transversal model's flexibility is maintained by selecting ROIs that account for both counterfeits and forgeries, making the method adaptable to different types of false IDs.	❒ Higher error rates, particularly elevated FP rates. ❒ Dependency on feature selection.
[6]	A model for ID authentication by hologram shape and color analysis.	Real-time Processing: the analysis is fast and can be performed in real-time.	❒ Sensitivity to image quality. ❒ The Model's applicability is restricted to the authentication of ID with holographic patterns.
[7]	Proposed an approach to hologram authentication through Multi LBP model adapts to variations in hologram appearance.	❒ No need to choose a single reference image for each pattern thanks to the Multi LBP model. ❒ LBP descriptor robustness against illumination variations, making the model suitable for semi-constrained environments, such as smartphone-captured videos. ❒ Real-time processing due to the small size of the images.	❒ Influenced by the quality of the captured hologram images. ❒ The Model's applicability is restricted to the authentication of documents with holographic patterns.

2.2 Forgery Detection Approaches for Document Authentication

The work in [19] addressed the challenge of document forgery detection, specifically focusing on forged receipts, using natural language processing (NLP) techniques. The authors propose a regression-based approach, leveraging a pre-trained language model (CamemBERT) to represent textual content. Additionally, they enrich the representation with domain-specific ontology-based entities and relations. The study compares various input types, including raw text, extracted entities, and triple-based reformulation of document content. The experiments utilize a dataset of forged receipts, providing insights into the efficiency of the proposed methods. The work in [20] introduced a robust approach for detecting copy-move forgery in digital images, particularly focusing on challenges such as scaling, rotation, and compression forgeries. The method employs a keypoint-based image forensics approach utilizing a superpixel segmentation algorithm and Helmert transformation. The process involved keypoint extraction and matching using the SIFT algorithm, clustering and group merging based on spatial distance and geometric constraints, and forgery region localization and refining using zero mean normalized cross-correlation. The very fast copy-move forgery detection (VFCMFD) method in [21] addressed the challenge of copy-move forgery detection in digital images. It introduced a keypoint-based approach using the speeded-up robust features (SURF) detector and a novel fast feature matching algorithm based on the generalized two nearest-neighbor (g2NN) approach. The method efficiently extracts key points, performs a match search with reduced complexity, and applies clustering using the DBSCAN (density-based spatial clustering of applications with noise) algorithm to detect copied-moved areas. The final step involves computing convex hulls to identify forged regions. In Table 4 and 5, we have mentioned the characteristics, novelty, pros & cons of various SOTA techniques under this category.

Table 4: Characteristics of forgery detection approaches for document authentication.

Method	**The targeted application**	**The targeted object to analyze**	**The processed dataset/s**	**The used method/s**	**The performance**
[19]	Forgery detection on receipts	Address, Date, and price on the receipts.	Find it (Receipt dataset)	❐ Regression-based approach ❐ Leveraging a pre-trained language model (CamemBERT)	❐ F1-score = 96.7% ❐ Precision = 93.75% ❐ Recall = 100%
[20]	Copy-move forgery detection in digital images	Textual information	❐ CMH series datasets ❐ D0-D3 series datasets ❐ CMH5 (compressed dataset)	❐ Keypoint extraction and matching via SIFT ❐ Helmert transformation	❐ Recall = 79% ❐ Precision = 86% ❐ FPR = 0.99% ❐ F1-score = 83%
[21]	Copy-move forgery detection in digital images	Textual information	4K Ultra HD images	❐ SURF keypoints detection ❐ Feature-matching algorithm based on the generalized two-nearest-neighbor (g2NN)	F1-score>90%

Table 5: Summaries of forgery detection approaches for document authentication.

Method	Novelty	Pros	Cons
[19]	Introduced a domain-specific ontology for receipts, explicitly representing the relationships between entities and providing a more detailed understanding of the document's structure	❒ Transfer learning from NLP methods ❒ High performance especially when using triples as input	Bias in the dataset such as a high frequency of receipts from a specific company (Carrefour)
[20]	Incorporated keypoint-based forensics, advanced clustering and merging strategies, and robust forgery region localization techniques to achieve accurate and reliable copy-move forgery detection in digital images	❒ Robustness against transformations ❒ Forgery localization accuracy	❒ Threshold dependence ❒ Computational complexity when dealing with high-resolution images
[21]	Introducing a new fast feature-matching algorithm based on the generalized two-nearest-neighbor (g2NN) approach	❒ Fast computation ❒ Applicability to large images ❒ Robust keypoint extraction	Dependency on parameters (e.g., thresholds for angles, distances, and radius).

2.3 Quality Detection Approaches for Document and ID Authentication

The work in [22] introduced CheckScan, a document liveness detection method for ID verification, focusing on quality verification. The proposed approach involves FFT-based feature extraction and reference hashing to discriminate between original ID templates and scanned versions. The feature extraction identifies discriminative FFT peaks, while the hash construction step maps these peaks into binary codes using a novel quantization scheme. The work in [23] focused on detecting forgeries in paper documents, particularly those created using the scan-edit and print (SEP) technique, where genuine documents are digitized and manipulated using image processing software. The authors propose an automatic forgery detection method based on intrinsic features at the character level. The approach involves outlier character detection in a discriminant feature space and identifying strictly similar characters. The method utilized shape descriptors and features like character size, principal inertia axis, and horizontal alignment for forgery detection. The work in [24] focused on automatic document authentication through the Delaunay layout descriptor (DLD), a method based on spatial relationships of document regions. The authors propose a refined matching algorithm for the DLD, combining global and local matching to address issues with different numbers of segmented regions in authentic copies. In Tables 6 and 7, we have provided details on the characteristics, novelty, advantages, and disadvantages of various SOTA techniques.

2.4 Contrastive Learning-based Approaches

An anomaly detection model, based on contrastive learning is proposed in [4]. This work, which represents a scheme of self-supervised learning, adopts two steps to detect cut-paste defects in local regions of an image without anomalous data; the first step aims to learn deep representations from normal data based on contrastive learning, and the second step is to build a generative one-class classifier by using the learned representations. In [8], a novel decoder-encoder framework for forgery detection is introduced which consists of three components: a generative network, a contrastive network, and a mutual information estimator. The generative network (i.e., decoder) learns by mapping the latent vector to a high-dimensional image space, while

Table 6: Characteristics of quality detection approaches for ID/document authentication.

Method	The targeted application	The targeted object to analyze	The processed dataset/s	Used method/s	Performance
[22]	Original or scan ID verification	Quality of Guilloche pattern	MIDV2020	❒ FFT ❒ Hashing	Accuracy $> 95\%$
[23]	Forgery detection on scan-edit and print documents	Text	A synthetic fraudulent document	❒ Shape descriptors ❒ Intrinsic features at the character level	❒ Recall = $22.0 - 100\%$ ❒ Precision = $28.0 - 70.0\%$ r
[24]	Printed and scanned document authentication	Text	❒ LayoutCopies ❒ DocCopies	Refined Delaunay layout descriptor	❒ FN = 0.011 (printed and scanned layouts) ❒ FP = 0.0 (printed and scanned layouts) ❒ FN = 0.3978 (real document dataset) ❒ FP = 0.0029 (real document dataset)

Table 7: Summaries of quality detection approaches for ID/document authentication.

Method	Novelty	Pros	Cons
[22]	Introduced a novel reference hashing method to discern between the original template of an ID image and its scanned version	❒ Discrimination capability ❒ FFT-based feature extraction and a novel quantization scheme, contribute to a well-anti-collision capability for distinct IDs	❒ An assumption on reference blocks ❒ Model's specificity is evaluated only on the MIDV dataset
[23]	Introduced an automatic forgery detection method for paper documents based on intrinsic features at the character level	❒ Applicability to various document types ❒ Adaptive threshold	❒ Lack of real-world data ❒ The method relies on certain assumptions, such as the existence of misalignments or skew in forged documents
[24]	Addressing the limitation of the DLD, which fails to match authentic copies with originals when the number of segmented regions differs between the documents.	❒ Improved matching performance ❒ Stability and robustness factors	❒ Limited tolerance to region variations ❒ Dependency on parameters such as similarity thresholds and quantization factors

the contrastive network (i.e., encoder) maps a positive/negative image to local feature maps and a global latent feature. The mutual information estimator is used to distinguish between the features from the target samples and their corresponding hard negative samples to effectively learn more discriminating latent feature presentation. The proposed framework helps to overcome the problems of generative adversarial networks and self-supervised approaches like instability training, mode dropping, and low discriminative ability. Furthermore, in [9], a simple approach based on contrastive learning of visual representations is proposed. The framework consists of three components : (i) data augmentations for enhancing contrastive predictions, (ii) nonlinear transformation between the representation and contrastive loss to improve the quality of learned representations, and (iii) contrastive learning. These three components are combined for effective visual representation learning. For further exploration of image recognition and contrastive learning, we refer the reader to the following works [25–29].

2.5 Adversarial Learning-based Approaches

More recent research in the literature has been focused on the provision of adversarial training for different applications. In [30], a novel anomaly detection technique is proposed by employing a framework of encoder-decoder-encoder sub-networks which is based on a conditional generative adversarial network. The generation of high-dimensional image space and the inference of latent space are jointly learned to help the model learning and the data distribution for the normal samples. In the domain of data hiding, the adversarial network is applied [11] to encode a rich amount of useful information (secret message) as invisible perturbations in the encoded image. This technique jointly trains the encoder and the decoder; the encoder produces a visually indistinguishable encoded image by using the given input message and cover image, and the decoder works to extract the original message from the encoded image. The adversary network works to improve the quality of an encoded image by minimizing the ability of an adversary to distinguish the encoded images.

3 Proposed Models

In this paper, we propose two models to distinguish between forged and non-forged documents in the first paradigm. In the second paradigm, we aimed to learn how to differentiate between real and fake IDs. The first proposed model learns the representations in a contrastive learning manner, named the contrastive forgery detection model (ContFD), and the second model learns the representations based on an adversarial setting, named the constrained adversary-based forgery detection model (AdvFD).

3.1 ContFD Model: Contrastive-based Forgery Detection Model

This model employs an encoder-decoder-classifier sub-network which enables the model to map the input image into a lower-dimension feature vector, and then reconstruct the output image. The objective of the classifier is to classify the input image into a real or a fake image.

An encoder network $E_\theta(.)$, parameterized by θ, receives a pair of IDs $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ and extracts latent feature vectors $z_x \in \mathbb{R}^{d\times 1}$ and $z_y \in \mathbb{R}^{d\times 1}$ where $z_x = E_\theta(x)$ and $z_y = E_\theta(y)$ respectively. These latent vectors z_x and z_y are used to compute the contrastive loss (denoted as $\mathcal{L}_{contrastive}^{z_x,z_y,\ell_x,\ell_y}$) to discriminate between z_x and z_y. Here, an image pair (x, y) is fed into the model as input and the objective is to discriminate between the image pair, where ℓ_x and ℓ_y are the labels of input x and y. A decoder network $D(.)$ receives the latent spaces z_x and z_y respectively to reconstruct the same input image pair, fed into the encoder. The decoder output is given by $\hat{x}$ and $\hat{y}$. The decoder's objective is to ensure the reconstruction of the input to the network which inherently operates to generate better latent feature vectors z_x and z_y which carry a better representation of the network's input. We use the point-wise $L1$ loss and SSIM loss [31] term for image reconstruction by the decoder. SSIM loss is a commonly used metric for image reconstruction tasks. It has been recently shown to be a good loss term for depth estimating CNNs [32]. The other part of the model is a classifier $f(.)$ network, whose task is to classify the latent feature vectors z_x and z_y into the class of "real" or "fake". An overview of the ContFD model is depicted in Figure 1.

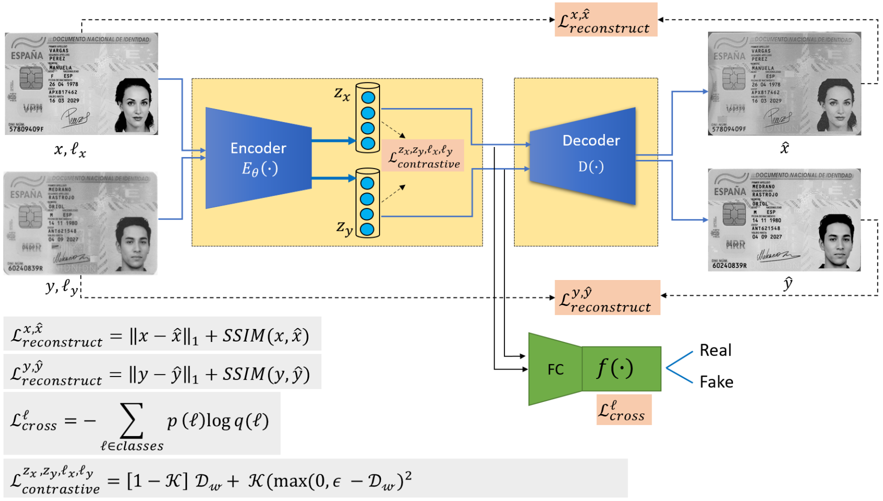

$\mathcal{L}^{x,\hat{x}}_{reconstruct}$
x, ℓ_x
y, ℓ_y
Encoder
$E_\theta(\cdot)$
z_x
z_y
$\mathcal{L}^{z_x,z_y,\ell_x,\ell_y}_{contrastive}$
Decoder
D(·)
$\hat{x}$
$\hat{y}$
$\mathcal{L}^{y,\hat{y}}_{reconstruct}$
FC
$f(\cdot)$
Real
Fake
$\mathcal{L}^{\ell}_{cross}$
$\mathcal{L}^{x,\hat{x}}_{reconstruct} = \|x - \hat{x}\|_1 + SSIM(x, \hat{x})$
$\mathcal{L}^{y,\hat{y}}_{reconstruct} = \|y - \hat{y}\|_1 + SSIM(y, \hat{y})$
$\mathcal{L}^{\ell}_{cross} = -\sum_{\ell \in classes} p(\ell) \log q(\ell)$
$\mathcal{L}^{z_x,z_y,\ell_x,\ell_y}_{contrastive} = [1 - \mathcal{K}]\, \mathcal{D}_w + \mathcal{K}(\max(0, \epsilon - \mathcal{D}_w)^2$

Fig. 1: An overview of ContFD for forgery detection of IDs.

3.1.1 ContFD model training The objective of the proposed model is to perform the classification of IDs by (i) minimizing the distance between the latent spaces (z_x and z_y) of an input pair of IDs if the pair belongs to the same class (i.e., the pair of IDs are either real or both are fake), or maximizing the distance between them if the pair does not belong to the same class (i.e., one ID is real and the other one is fake); (ii) minimizing the difference between the input and decoded (reconstructed) images; (iii) maximizing the ability of the classifier $f(.)$ to classify the input pair correctly. Three loss functions are introduced: contrastive loss (i.e., $\mathcal{L}^{z_x,z_y,\ell_x,\ell_y}_{contrastive}$), reconstruction loss (i.e., $\mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct}$), and cross-entropy loss or classification loss (i.e., $\mathcal{L}^{\ell}_{cross}$) of the classifier $f(.)$.

The *contrastive loss* is a metric learning loss, which imposes constraints on the distribution of the model's inner representation (i.e., latent vectors) of the input data, i.e., the model can learn any features regardless of whether after transformation the similar latent features would be located close to each other or not. The contrastive loss enforces the embedding function of the encoder $E_\theta(.)$ to learn how to encode features such that samples from the same classes have similar features, and samples from different classes have very different ones. In so doing, the contrastive loss helps to minimize the embedding distance between the embedding spaces (z_x, z_y) if they are from the same class or instead to maximize the distance between them if they belong to a different class. Mathematically, the contrastive loss is represented in eq. 1.

$$\begin{aligned} \mathcal{L}^{z_x,z_y,\ell_x,\ell_y}_{contrastive} = \{(1-\mathcal{K}) \times {\mathcal{D}_{\mathcal{W}}}^2\} + \\ \{\mathcal{K} \times max(0, \epsilon - \mathcal{D}_{\mathcal{W}})^2\} \end{aligned} \tag{1}$$

where

$$\mathcal{D}_{\mathcal{W}} = \|z_x - z_y\|_2 \ , \ \ \mathcal{K} = \begin{cases} 0 & \text{if } [\ell_x = \ell_y] \\ 1 & \text{if } [\ell_x \neq \ell_y] \end{cases}$$

The constant ϵ is a margin, defining the lower bound distance between samples of different classes. The $\mathcal{K}$ term here specifies whether ℓ_x and ℓ_y are similar then $\mathcal{K} = 0$ or dissimilar then $\mathcal{K} = 1$. The $\mathcal{D}_{\mathcal{W}}$ term represents similarity (or, rather, dissimilarity) between the latent feature vectors.

The second loss i.e., *reconstruction loss* ($\mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct}$) consists of two terms i.e., $\mathcal{L}^{x,\hat{x}}_{reconstruct}$ and $\mathcal{L}^{y,\hat{y}}_{reconstruct}$ respectively (see eq. 2). Whereas, each of the $\mathcal{L}^{x,\hat{x}}_{reconstruct}$ and $\mathcal{L}^{y,\hat{y}}_{reconstruct}$ term is consisting of point-wise $L1$ loss (defined over x v/s $\hat{x}$ and y v/s $\hat{y}$) and the $SSIM$ term which computes the structural similarity between the input pair (x, y) and the reconstructed pair ($\hat{x}$, $\hat{y}$).

$$\mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct} = \mathcal{L}^{x,\hat{x}}_{reconstruct} + \mathcal{L}^{y,\hat{y}}_{reconstruct} \tag{2}$$

where

$$\begin{aligned} \mathcal{L}^{x,\hat{x}}_{reconstruct} &= \|x - \hat{x}\|_1 \ + \ SSIM(x, \hat{x}) \\ \mathcal{L}^{y,\hat{y}}_{reconstruct} &= \|y - \hat{y}\|_1 \ + \ SSIM(y, \hat{y}) \end{aligned}$$

The third loss i.e., *cross-entropy* loss ($\mathcal{L}^{\ell}_{cross}$) is a metric that is used to measure how well a classification function $f(.)$ in machine learning performs. With $\mathcal{L}^{\ell}_{cross}$ we try to maximize the classification accuracy by using our training data. Mathematically, the cross-entropy loss is expressed in eq. 3.

$$\mathcal{L}^{\ell_x,\ell_y}_{cross} = \mathcal{L}^{\ell_x}_{cross} + \mathcal{L}^{\ell_y}_{cross} \tag{3}$$

where

$$\begin{aligned} \mathcal{L}^{\ell_x}_{cross} &= - \sum_{\ell_x \in classes} p(\ell_x) \, log \, q(\ell_x) \\ \mathcal{L}^{\ell_y}_{cross} &= - \sum_{\ell_y \in classes} p(\ell_y) \, log \, q(\ell_y) \end{aligned}$$

and $p(\ell)$ is the true probability distribution and $q(\ell)$ is the model's predicted probability distribution of the output classes. In the following section, we train the ContFD model in three different schemes by using distinctive objective functions.

3.1.2 End-to-end learning scheme (ContFD-e2e) End-to-end learning is a technique where the model is trained to learn a mapping from the input to the output in a single step. The goal is to train a single model that can handle the entire task without the need for any additional processing or pre-processing. The input and output data are fed into the model, and the model learns to produce the correct output of a given input. Here, we train the model by optimizing the total loss. Hence, our objective function is expressed in eq. 4.

$$\mathcal{L}_{e2e} = \mathcal{L}^{z_x,z_y,\ell_x,\ell_y}_{contrastive} + \mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct} + \mathcal{L}^{\ell}_{cross} \tag{4}$$

3.1.3 Dual learning scheme (ContFD-dual) Dual learning, on the other hand, is a technique where two models are trained simultaneously in a way that they can learn from each other. The goal is to train two models that can each handle a different aspect of the task and then combine their outputs to produce the final results. This approach can be especially useful in cases where the input and output data are not easily connected, or when the task requires multiple steps. Here, we train the model in the following two phases:

Phase 1

We train the model using a partial loss $\mathcal{L}_{P_1}$, then all parameters of the obtained model in this phase are frozen before starting *phase 2*. Therefore, the objective function in *phase 1* is expressed in eq. 5.

$$\mathcal{L}_{P_1} = \mathcal{L}_{contrastive}^{z_x,z_y,\ell_x,\ell_y} + \mathcal{L}_{reconstruct}^{x,\hat{x},y,\hat{y}} \tag{5}$$

Phase 2

The model training is continued for supervised classification based on $\mathcal{L}_{P_2}$. Therefore, the objective function in *phase 2* is represented by eq. 6.

$$\mathcal{L}_{P_2} = \mathcal{L}_{cross}^{\ell} \tag{6}$$

3.1.4 Fine-tuning scheme (ContFD-FT) Fine-tuning works by using a pre-trained model as a starting point and then continuing to train it on the new task, typically with a smaller learning rate than was used to train the original model. The pre-trained model's weights are used as initialization, and the model is updated with backpropagation to better fit the new task. The goal is to modify the model so that it is better suited to the new task, while still retaining the information learned from the original task. Here, the model is trained in two phases as well:

Phase 1

We train the model using a partial loss $\mathcal{L}_{P_1}$, which is then considered as a pre-trained model before starting phase 2. Hence, our objective function is the same as in eq. 5.

Phase 2

The learning of the model is done in a supervised manner by optimizing (by fine-tuning) the pre-trained model that is obtained in phase 1 and takes into consideration $\mathcal{L}_{cross}^{\ell}$, which gives the final objective function as in eq. 4.

End-to-end learning and dual learning are two popular techniques used in machine learning and artificial intelligence. Both are used to improve the performance of models by optimizing the inputs and outputs of the system. However, there are some key differences between the two. The fine-tuning technique works to leverage a pre-trained model and adjusts its weights to better suit a specific task. The goal is to improve the performance of the model for the new task while still retaining the knowledge learned from the original task.

Likewise, we also perform the analysis of weighted loss functions for all of these proposed "ContFD" models.

3.2 AdvFD Model: Adversarial Model for Forgery Detection

The second model is very similar to the network architecture, shown in Fig. 1. The only difference is that the classifier network $f(.)$ is replaced here by a constrained adversarial model $A(.)$. In the case of this model, we arbitrarily select the first input x as real (labeled as $\ell_{x:real}$) and the second input y as fake (labeled as $\ell_{y:fake}$). Like a classifier, the adversary network $A(.)$ also predicts whether a given input is real or fake and this provides the adversarial loss which helps to extract better-quality latent feature vectors. An overview of the AdvFD model is depicted in Figure 2. Contrary to the classical adversarial model, the proposed constrained adversarial network is trained in an adversarial way by feeding only the fake images. Training in this manner not only helps to obtain good-quality image reconstruction but also helps to generate a better discriminatory latent space (compared to the latent space, generated from real images) of the fake input. In this manner, we avoid any interference in generating the embedding space (or latent space) from the real image. We hypothesize that a significant difference/dissimilarity exists between the latent vector z_x (mapped from real image x) and z_y (mapped from fake image y). Such dissimilarity between z_x and z_y helps the adversarial model to classify.

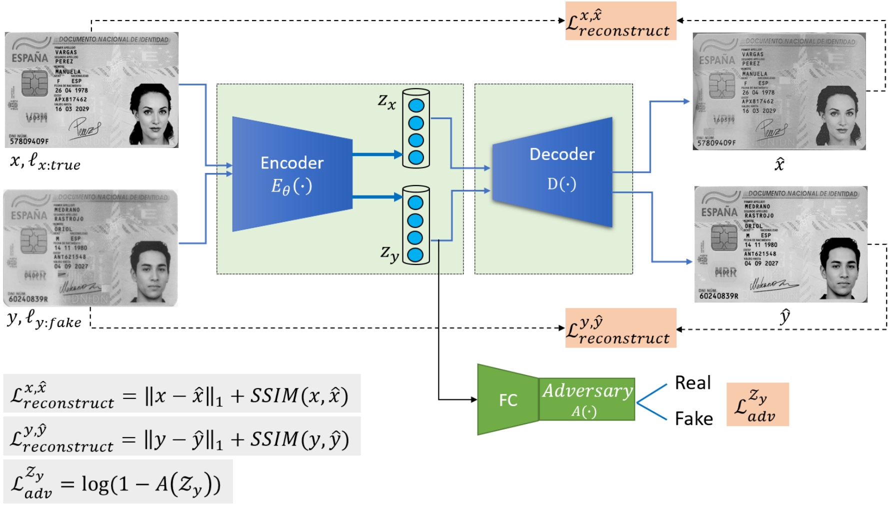

$\mathcal{L}^{x,\hat{x}}_{reconstruct}$
$x, \ell_{x:true}$
Encoder
$E_\theta(\cdot)$
z_x
z_y
Decoder
D(·)
$\hat{x}$
$\hat{y}$
$y, \ell_{y:fake}$
$\mathcal{L}^{y,\hat{y}}_{reconstruct}$
FC
Adversary
$A(\cdot)$
Real
Fake
$\mathcal{L}^{z_y}_{adv}$
$\mathcal{L}^{x,\hat{x}}_{reconstruct} = \|x-\hat{x}\|_1 + SSIM(x,\hat{x})$
$\mathcal{L}^{y,\hat{y}}_{reconstruct} = \|y-\hat{y}\|_1 + SSIM(y,\hat{y})$
$\mathcal{L}^{z_y}_{adv} = \log(1 - A(z_y))$

Fig. 2: An overview of our constrained-adversary-based model (AdvFD).

3.2.1 AdvFD model training As we arbitrarily select the first input x as real and the second input y as a fake of the whole network, the adversarial technique is adequate to fool the classification process by taking the latent space z_y of the fake input only. By following the principle of adversarial learning, here also the encoder $E_\theta(.)$ learns to minimize the ability of the adversarial network $A(.)$ to correctly detect the target label ($t\ell_y$) of the input y, but rather $E_\theta(.)$ learns to maximize the ability of the adversary to detect the ground truth label (ℓ_y) of the input y. It is worth noting that the input of the adversarial network is a latent space z_y (generated from a fake image) whose target label ($t\ell_y$) is real. Figure 3 depicts an overview of the constrained adversarial network.

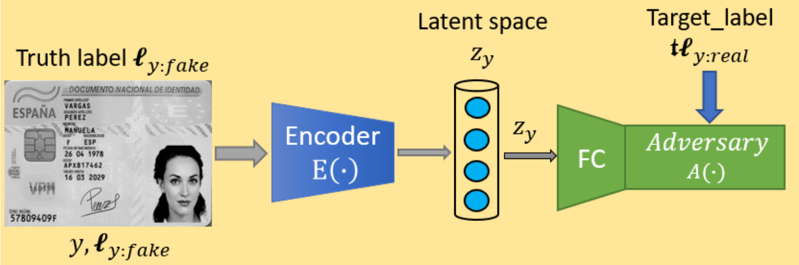


Fig. 3: Constrained-adversarial network architecture.

Simply speaking, employing the constrained adversarial network enhances the quality of latent spaces implicitly by maximizing the distance between the latent spaces z_x and z_y by considering only z_y as input. This leads also to minimizing the difference between the input and the reconstructed image (output of $D(.)$). The training process incorporates the following two loss functions, as articulated in eq. 7.

$$\mathcal{L}_{adversarial}^{x,\hat{x},y,\hat{y},z_y} = \mathcal{L}_{adv}^{z_y} + \mathcal{L}_{reconstruct}^{x,\hat{x},y,\hat{y}} \tag{7}$$

The first loss is *Adversarial loss* ($\mathcal{L}_{adv}^{z_y}$), which operates in an adversarial fashion by penalizing itself for misclassifying a fake instance (i.e., the latent space) as real and rewarding for correctly classifying the fake instance as fake. Therefore, the adversarial loss, indicating the discriminator's capability to detect z_y, is formulated in eq. 8.

$$\mathcal{L}_{adv}^{z_y} = \quad log\,(1 - A(z_y)) \tag{8}$$

The second loss is *Reconstruction* loss ($\mathcal{L}_{reconstruct}^{x,\hat{x},y,\hat{y}}$) (mentioned before in eq. 2). With the only adversarial loss, the generation of latent spaces by $E_\theta(.)$ is not optimized toward learning contextual information about the input data. Penalizing the generator by measuring this distance between the input and the generated images, remedies this problem.

Likewise, we also perform the analysis of weighted loss functions for all of these proposed "AvdFD" models.

4 Experimental Settings

This section details the information about the datasets, a sample of training and testing sets, the experiment settings, and finally the experimental results of the proposed models.

4.1 Real and Fake Datasets

To evaluate the performance of the proposed approaches, we require real and fake samples of IDs. *MIDV-2020* (mobile identity document video) [33] is used as a dataset has real samples of IDs, and FMIDV[4] (forged mobile identity document video) [1] is used as a dataset that has fake samples of IDs.

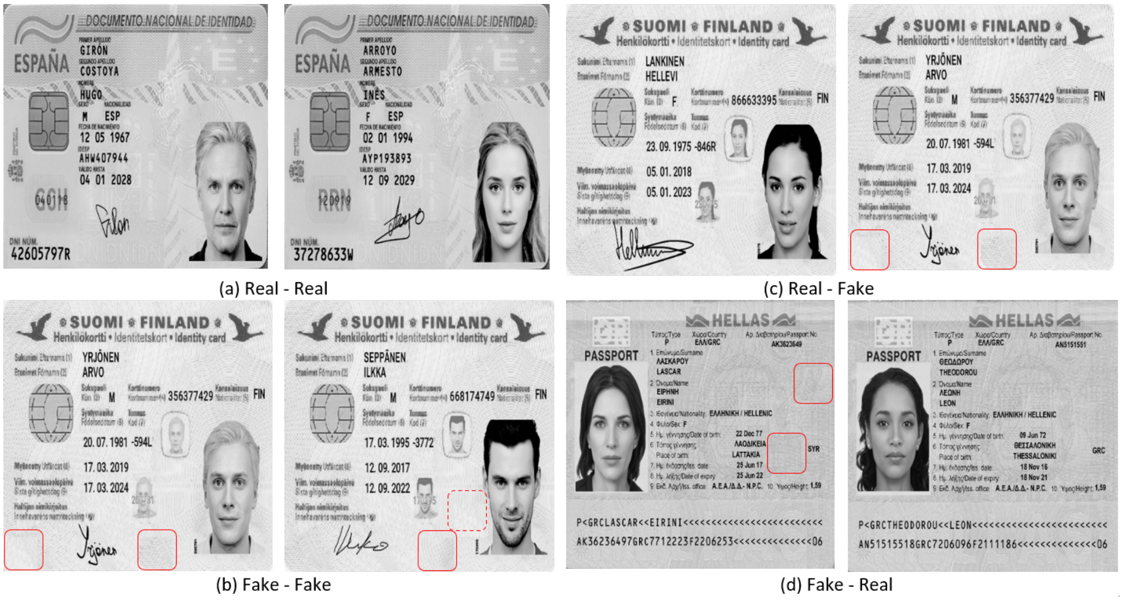
MIDV-2020 consists of 1000 unique dummy IDs which contain unique text fields and artificially generated faces. These dummy IDs are from 10 different countries; i.e., Albania (alb), Azerbaijan (aze), Spain

[4] http://l3i-share.univ-lr.fr/2022FMIDV/2022FMIDV.html

(esp), Estonia (est), Finland (fin), Greece (grc), Latvia (iva), Russia (rus), Serbia (srb), and Slovakia (svk). Based on these dummy IDs, 2000 scanned images were obtained from the flatbed scanner, and another 1000 images were captured by smartphones. *FMIDV* consists of 28000 fake IDs of the same 10 countries that were generated by applying the "copy-move" type of forgeries on the 4000 images i.e., 1000 dummy IDs, 2000 scanned images, and 1000 photos of *MIDV-2020*.

4.2 Sample of Training and Testing Sets

In the proposed models, the inputs are pairs of IDs. These pairs could be similar pairs or dissimilar pairs. A similar pair signifies two IDs that belong to the same class (either real or fake class) and the same country, while a dissimilar pair signifies two IDs that belong to different classes (one document belongs to the real class and the other one belongs to the fake class), but also belong to the same country. A sample of similar and dissimilar pairs of IDs, which are used as input for our proposed models, are shown in Figure 4. Note that the samples in Figure 4 are sourced from the public dataset MIDV.


DOCUMENTO NACIONAL DE IDENTIDAD
ESPAÑA
SUOMI FINLAND
Henkilökortti • Identitetskort • Identity card
HELLAS
PASSPORT

(a) Real - Real

(c) Real - Fake

(b) Fake - Fake

(d) Fake - Real

Fig. 4: (a, b) Sample of similar and (c, d) dissimilar pairs; red boxes present the forgery locations.

4.3 Experimental Setup

The training and testing of the proposed models are carried out on the dataset of each country separately. The training process takes place by taking pairs of samples, while the testing process takes one single sample as the input for verification. As mentioned, we have 4000 real samples in MIDV and 24000 fake samples in FMIDV (after removing the generated fake samples where the copy-move operation is performed on the zones of 64 × 64 because these samples are easily detectable by naked eyes). For each country, we have

400 real samples and 2400 fake samples. We have randomly selected $2/3$ of these samples and used them as training data and the remaining $1/3$ images are used as testing data. We choose the following size of training and testing samples from the aforementioned whole dataset of each country to prevent high computational time during training and testing as well as to strike a balance between the size of actual and fake samples.

For training, we have randomly selected 20 real samples and 20 fake samples from the training data set of each country. Hence, in the ContFD model, we have 1160 pairs for training which are distributed as $20 \times 19 = 380$ (real-real) pairs, and $20 \times 20 = 400$ (real-fake) pairs, $20 \times 19 = 380$ (fake-fake) pairs. While, for the AdvFD model, we have $20 \times 20 = 400$ (real-fake) pairs for training, as for this model, we have arbitrarily defined the first input as real and the second input as fake (can be taken in reversed order also). To test the performances of the proposed models, for all of our experiments, we choose different sizes of testing samples e.g. {30, 60, 90, 120, 150, 180, 210, 240}, which are selected randomly from the testing dataset. Moreover, all results presented, utilizing various evaluation metrics, are derived from a rigorous process involving 5 rounds of calculation, ensuring the robustness of the findings.

4.4 Evaluation Metrics

Three metrics are used in this article to evaluate the performance of the proposed models.

4.4.1 Accuracy is a metric for classification models that measures the number of predictions that are correct as a percentage of the total number of predictions that are made. The accuracy is expressed in eq. 9.

$$Accuracy = \frac{\#\, of\, correct\, predictions}{\#\, of\, total\, predictions} \tag{9}$$

4.4.2 F1-score it is one of the most important evaluation metrics to measure a model's accuracy on a data set. It sums up the predictive performance of a model by combining the precision and recall of a model into a single metric by considering its harmonic mean. The F1-score gives a better measure of the incorrectly classified cases than the accuracy metric. The F1-score is expressed in eq. 10.

$$F1 - score = \frac{TAR}{TAR + \frac{1}{2}(FAR + FRR)} \tag{10}$$

where the TAR, the FRR, and the FAR metrics are calculated according to the eq. 11, as reported in [34].

$$TAR = \frac{x_1}{X_1}, \qquad FRR = \frac{x_2}{X_1}, \qquad FAR = \frac{x_3}{X_2} \tag{11}$$

where x_1 is the number of real IDs that are predicted as real documents, x_2 is the number of real IDs that are predicted as fake documents, x_3 is the number of fake IDs that are predicted as real documents, X_1 and X_2 correspond to the total number of real and fake IDs, respectively.

4.4.3 AUC-ROC The receiver operating characteristics (ROC) curve is another evaluation metric for binary classification problems. It visualizes the trade-off probabilities between *FAR* and *TAR* at various thresholds. The AUC measures the ability of a classifier to distinguish between the classes. Hence, a higher AUC indicates a better performance of the model which has a high capability of distinguishing (or discriminating) between the real and fake classes. Indeed, the AUC is used as a summary of the ROC curve.

The training and testing are carried out on an in-house GPU server (28 CPUs, 128 Go RAM, 4 GPU Nvidia RTX 2080Ti cards), with the batch size = 8, and the number of epochs = 100. The "Adam" optimizer is used, where the learning rate (lr) equals $10e-4$ and the weight decay equals 0. The learning rate lr is scheduled at every 20 epoch by $gamma = 0.1$.

5 Experimental Results

This section presents the performance tests of the proposed models in two schemes: (i) with no-weights joint objective functions, and (ii) with weights joint objective functions.

5.1 No-Weights Joint Objective Functions

Here, we report the performance of predicting the correct label (either real or fake) of the test data in terms of accuracy, F1-score, and AUC of the proposed models.

Figure 5 presents the obtained accuracy of the proposed models on the different sizes of test samples from 10 countries. The AdvFD model achieves interesting results in comparison to the ContFD model, regardless of different training schemes. The accuracy values of the AdvFD model are greater than 60% for 8 countries and also exceed 50% for countries like Greece (grc) and Russia (rus) (due to the bad quality of the guilloche patterns of the IDs of these countries, the accuracy values are little lesser).

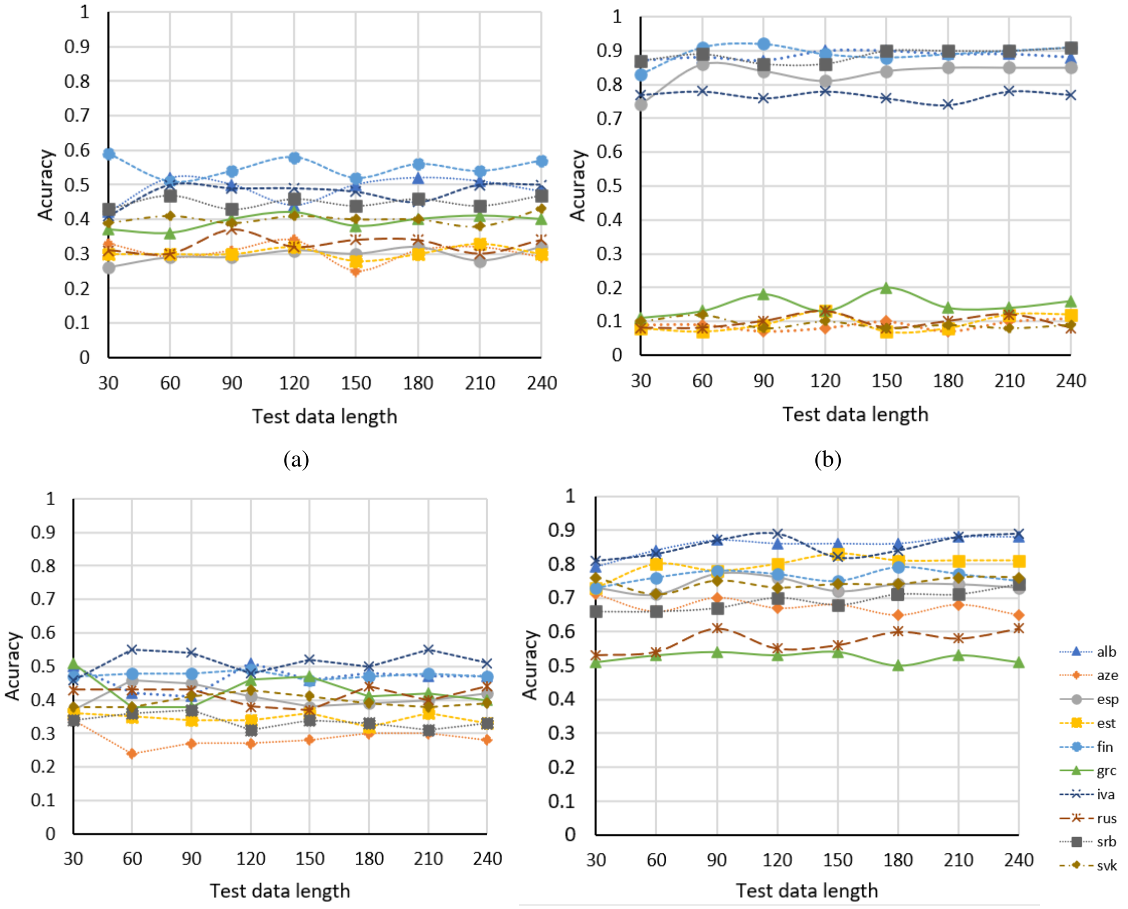
From the accuracy results of the ContFD model, we can see that the dual training scheme i.e., ContFD-dual achieves better accuracy than the end-to-end scheme (ContFD-e2e) and the fine-tuning scheme (ContFD-FT) for some countries. Indeed, the accuracy exceeds 75% for countries like Albania (alb), Spain (esp), Finland (fin), Latvia (iva), and Serbia (srb) and could not exceed 20% for the countries like Azerbaijan (aze), Estonia (est), Greece (grc), Russia (rus), Slovakia (svk). In ContFD-e2e and ContFD-FT, the accuracy values are within 25%-60%.

Figure 6 presents the F1-score results of the proposed models. We can see that these results are coherent with the aforementioned accuracy results in Figure 5. We can see that the AdvFD model achieves interesting F1-score results compared to the achieved F1-score results of the ContFD model. The F1-scores of the AdvFD model exceed 55% for all countries.

Similarly, the F1-scores of the ContFD-dual achieve better F1-scores compared to ContFD-e2e and ContFD-FT for some countries. Indeed, the F1-scores exceed 75% for countries like *alb*, *esp*, *fin*, *iva*, *srb*, and do not exceed 10% for countries like *aze*, *est*, *grc*, *rus*, *svk*. That can be explained due to the bad quality of the guilloche patterns in these countries. For ContFD-e2e and ContFD-FT, the F1-scores range between 25% and 60%. From the previously presented results in Figures 5 and 6, we can conclude that the constrained-adversary-based training model (AdvFD) achieves better performance than the contrastive-based training model (ContFD) in differentiating between the real and fake classes. Figure 7 summarizes the performances of the proposed models for distinguishing between the real and fake classes in terms of AUC. Specifically, the AUC of AdvFD exceeds 55% for five countries i.e., *alb*, *aze*, *grc*, *srb*, *svk*, and the AUC of AdvFD model is not less than 50%. For *est*, *fin*, *iva*, and *rus*, the AUC for ContFD-dual shows a better performance compared to other proposed models. Also, the ContFD-e2e model outperforms the other suggested models in terms of AUC for the results of Spain's (esp) IDs.

5.2 Weights Joint Objective Functions

Loss functions are used to gauge how well the model's predictions correspond to ground truth. The technique to add weights to the loss function is to multiply by a weight factor before summing up the individual losses.



(a)

(b)

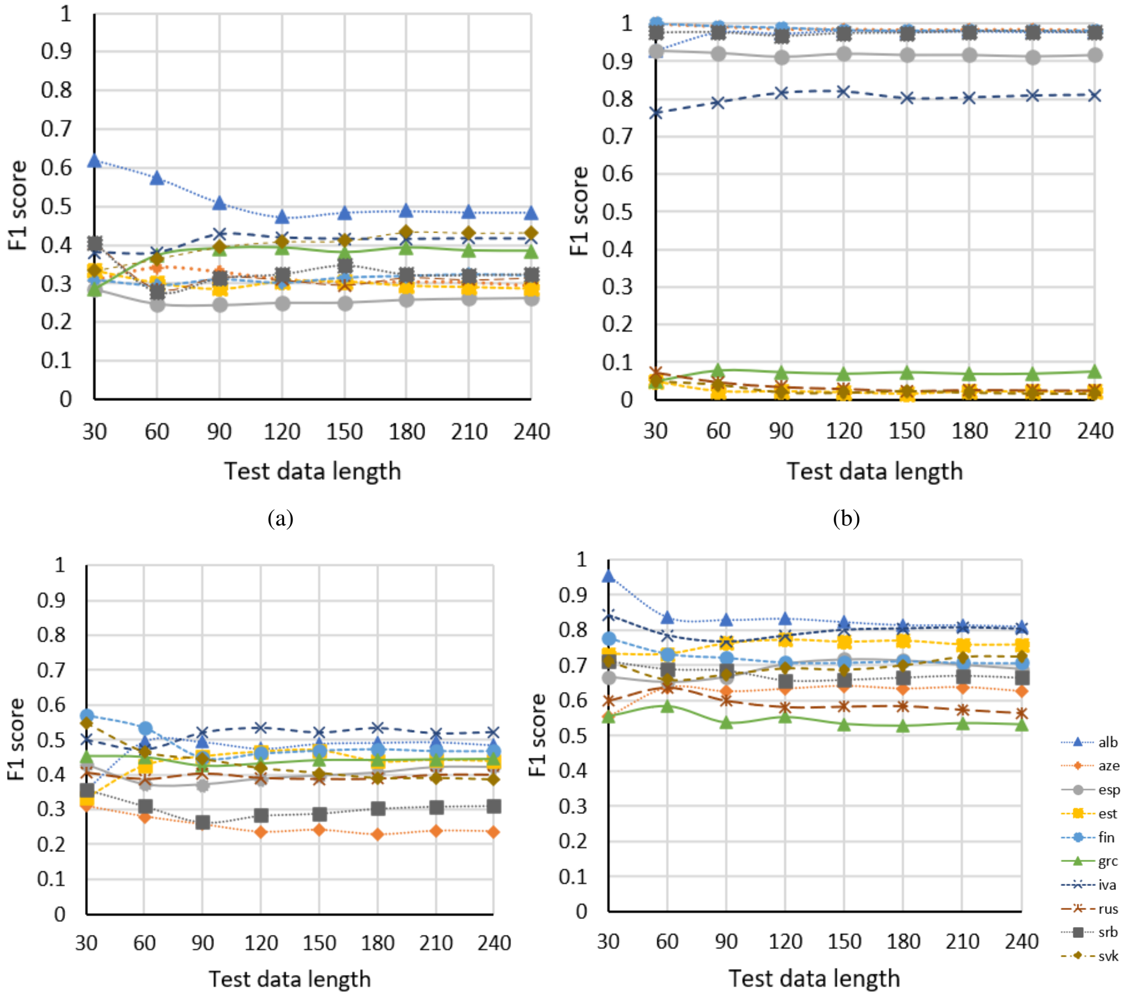
(c)

(d)

Fig. 5: Accuracy results of (a) ContFD-e2e, (b) ContFD-dual, (c) ContFD-FT, (d) AdvFD.

To this end, firstly, we add weights into the joint partial loss functions before summing all the individual losses in each of the proposed models (expressed as adding sub-weights into the partial loss functions). Secondly, we add weights into the computation of total loss, which represents the overall loss of all the individual losses for each of the proposed approaches (expressed as adding full weights into the total loss functions). The purpose of adding sub-weights and full-weights can be seen as a way to employ variable weights to the different individual losses of the network to regulate and improve the overall performance of the model.

5.2.1 Applying sub-weights and full-weights in the ContFD model Here, we explain in detail how we have added the sub-weights and full-weights to the partial and total loss functions of the ContFD model, followed by the performance evaluation.



(a)

(b)

(c)

(d)

Fig. 6: F1-score results of (a) ContFD-e2e, (b) ContFD-dual, (c) ContFD-FT, (d) AdvFD.

5.2.1.1 Adding the sub-weights into the partial loss function:

In continuation with the description in Section 3 about the ContFD Model which consists of three partial loss functions i.e., reconstruction loss ($\mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct}$), contrastive (encoded) loss ($\mathcal{L}^{z_x,z_y,\ell_x,\ell_y}_{contrastive}$) and cross-entropy loss ($\mathcal{L}^{\ell}_{cross}$). Here, we give a thorough description of how we included sub-weights in the partial loss functions of several proposed models, followed by the performance evaluation of these models. As previously stated, the *"ContFD"* model has three distinct schemes. We have outlined the technique which we used to integrate sub-weights into each of these three schemes. To implement the desired weighted loss function, we incorporated four ($w \in \{w_{z_x^1}, w_{z_x^2}, w_{z_y^1}, w_{z_y^2}\}$) weights (a pair of weights is computed from

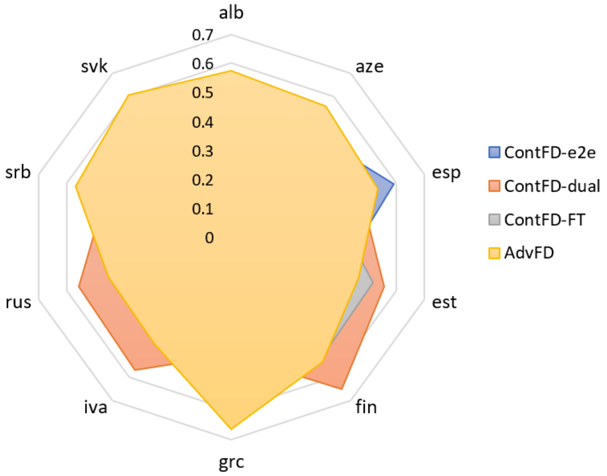


Fig. 7: AUC results of the proposed models.

each latent vector i.e., z_x and z_y) into the partial loss functions of *"ContFD-e2e"* model. Accordingly, equations 2 and 3 are revised to equations 12 and 13, respectively.

$$\mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct} = w_{z_x^1}\,\mathcal{L}^{x,\hat{x}}_{reconstruct} + (1-w_{z_x^1})\,\mathcal{L}^{y,\hat{y}}_{reconstruct} \tag{12}$$

where

$$\begin{aligned}\mathcal{L}^{x,\hat{x}}_{reconstruct} &= w_{z_x^2}\,\|x-\hat{x}\|_1 \,+\, (1-w_{z_x^2})\,SSIM(x,\hat{x})\\ \mathcal{L}^{y,\hat{y}}_{reconstruct} &= w_{z_y^1}\,\|y-\hat{y}\|_1 \,+\, (1-w_{z_y^1})\,SSIM(y,\hat{y})\end{aligned}$$

$$\mathcal{L}^{\ell}_{cross} = w_{z_y^2}\,\mathcal{L}^{\ell_x}_{cross} + (1-w_{z_y^2})\,\mathcal{L}^{\ell_y}_{cross} \tag{13}$$

For the second scheme i.e., in the *"Dual"* learning scheme of *"ContFD"* model, the weighted loss function is computed in two steps. In the first step, eq. 2 gets updated by adding two weights i.e., $w_{z_x^1}, w_{z_y^1}$ as in eq. 14.

$$\mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct} = \mathcal{L}^{x,\hat{x}}_{reconstruct} + \mathcal{L}^{y,\hat{y}}_{reconstruct} \tag{14}$$

where

$$\begin{aligned}\mathcal{L}^{x,\hat{x}}_{reconstruct} &= w_{z_x^1}\,\|x-\hat{x}\|_1 \,+\, (1-w_{z_x^1})\,SSIM(x,\hat{x})\\ \mathcal{L}^{y,\hat{y}}_{reconstruct} &= w_{z_y^1}\,\|y-\hat{y}\|_1 \,+\, (1-w_{z_y^1})\,SSIM(y,\hat{y})\end{aligned}$$

In the second step, eq. 3 is modified to eq. 15 by the inclusion of the weight $w_{z_x^2}$.

$$\mathcal{L}^{\ell}_{cross} = w_{z_x^2}\,\mathcal{L}^{\ell_x}_{cross} + (1-w_{z_x^2})\,\mathcal{L}^{\ell_y}_{cross} \tag{15}$$

Hence, these three weights i.e., $w_{z_x^1}$, $w_{z_x^2}$ and $w_{z_y^1}$ are generated ($w_{z_x^1}$, $w_{z_x^2}$ comes from the latent vector z_x and $w_{z_y^1}$ comes from the latent vector z_y) in the same manner as explained before.

For the third scheme i.e., the *"Fine-tuning"* scheme, we need only one weight i.e., $w_{z_x^1}$, which is obtained from the latent vector z_x in the same manner as explained before. As a result, eq. 3 is modified to eq. 16.

$$\mathcal{L}^{\ell}_{cross} = w_{z^1_x}\,\mathcal{L}^{\ell_x}_{cross} + (1 - w_{z^1_x})\,\mathcal{L}^{\ell_y}_{cross} \tag{16}$$

5.2.1.2 Adding full weights into the total loss functions:

The individual weights are added in the following manner to compute the final loss. In the case of the *End-to-end (ContFD-e2e)* learning scheme of training, it is required to generate three weight values i.e., $\lambda \in \{\lambda_1, \lambda_2, \lambda_3\}$ for optimizing the total loss. Please note that here we need 3 individual weights instead of 2 because we need to maintain the constraint; $\sum\limits_{i=1}^{3} \lambda_i = 1$. Accordingly, eq. 4 is modified to eq. 17.

$$\mathcal{L}_{e2e} = \lambda_1 \mathcal{L}^{z_x, z_y, \ell_x, \ell_y}_{contrastive} + \lambda_2 \mathcal{L}^{x, \hat{x}, y, \hat{y}}_{reconstruct} + \lambda_3 \mathcal{L}^{\ell}_{cross} \tag{17}$$

For the of *Dual (ContFD-dual)* learning scheme of training, only one weight (i.e., $\lambda = \{\lambda_1\}$) is required in *phase 1*. As a result, eq. 5 is updated to eq. 18.

$$\mathcal{L}_{P_1} = \lambda_1 \mathcal{L}^{z_x, z_y, l_x, l_y}_{contrastive} + (1 - \lambda_1) \mathcal{L}^{x, \hat{x}, y, \hat{y}}_{reconstruct} \tag{18}$$

Whereas, for phase 2, the training is performed based on eq. 6, where we don't have any possibility/necessity to add weights in the final computation of total loss.

In the case of *Fine-tuning (ContFD-FT)* scheme, the model training in $phase1$ is performed in the same manner as it is done in eq. 18, whereas, in $phase2$, the model training is continued by using eq. 6. Here also, we don't have any possibility to add weights in the final computation of the total loss.

5.2.1.3 Computing sub-weights and full-weights for ContFD:

Here, we explain the technique to automatically compute the optimized weights to add them to the partial and the total loss functions of ContFD models.

❒ **Generating sub-weights for the loss function:** The technique used for generating the sub-weights in our models starts by passing the generated corresponding latent vectors (either z_x or z_y[5]) from each image in the input image pair through two **Conv-BN-RELU** layers, one "Average Pooling"[6] layer and multiple **DropOut-FC(i_vect, o_vect)-RELU** layers. Finally, the output is passed through the **Sigmoid** () activation function to obtain the weight value between 0 to 1. Where, **Conv(**$in_{ch} \rightarrow out_{ch}; k$**)**, **BN**, **RELU**, and **FC** represent, convolutional, batch normalization layers, activation function, and fully connected layers respectively. The **i_vect** and **o_vect** in **FC(i_vect, o_vect)** represent the number of input and output vectors of the fully connected layer respectively. Whereas, in_{ch}, out_{ch}, and k represent the number of input, output channels, and kernel size respectively.
The structure of above mentioned first and second **Conv-BN-RELU** layers are **Conv (1664, 512, 7)-BN-ReLU** and **Conv (512, 256, 5)-BN-ReLU** respectively. There are a total of 5 **DropOut-FC(i_vect, o_vect)-RELU** blocks which are added after. Then one "Average Pooling" (i.e., **AdaptiveAvgPool2d(**$pool_{size}$**)**) layer is applied to downsample the input along its spatial dimensions (height × width) by taking the average value over an input window (of size defined by $pool_{size}$) for each channel of the input. In this case, the **AdaptiveAvgPool2d(1)** layer takes an input of size $(256 \times 3 \times 3)$ and transforms it into a vector of size $(256 \times 1 \times 1)$ which is then flattened to obtain a vector of size 256 only. Then, this 256-dimensional

[5] The dimensions of each latent vector are $1664 \times 7 \times 7$.

[6] Here we have used the "Adaptive Average Pooling" algorithm from the PyTorch library. For more details, see: https://pytorch.org/cppdocs/api/classtorch_1_1nn_1_1_adaptive_avg_pool1d.html.

vector is passed through the first block of **DropOut**-**FC** (256 → 128)-**RELU**; the second block of **DropOut**-**FC** (128 → 64)-**RELU**; the third block of **DropOut**-**FC** (64 → 32)-**RELU**; the fourth block of **DropOut**-**FC** (32 → 16)-**RELU**; and the fifth block of **DropOut**-**FC** (16 → 2)-**RELU** respectively. Depending on the batch size (b), used for training, finally, we will obtain an output of dimension $b \times 2$, which is then averaged row-wise to finally obtain a vector of size 2. These 2 values are then passed through the **Sigmoid ()** activation function to obtain 2 weight values in the range of 0 to 1.
In the same manner, we obtain another 2 weight value from the second image of the input image pair. In this way, we will be able to compute 4 weights i.e., $w_{z_x^1}, w_{z_x^2}, w_{z_y^1}$, and $w_{z_y^2}$.

❒ **Generating full weights for the loss function:** We follow a similar approach to generate the full weights. At first, the two latent vectors, obtained from the input image pair are concatenated (i.e., z_x and z_y are concatenated along the first dimension[7].) and then passed into two **Conv-BN-RELU** layers, one "Average Pooling" layer and multiple **DropOut-FC(i_vect, o_vect)-RELU** layers. Finally, in the same manner, the output is passed through the **Sigmoid ()** activation function to obtain the weight value between 0 to 1. The structure of the first and second **Conv-BN-RELU** layers are: **Conv (1664×2, 512, 7)-BN-RELU**, **Conv (512, 256, 5)-BN-RELU**. After that, a 2D adaptive average pooling **AdaptiveAvgPool2d(1)** layer is applied over the input of size $256 \times 1 \times 1$ to transform it into a vector of size $(256 \times 1 \times 1)$ which are the flattened to obtain a vector of size 256 only. Then, this 256 dimensional vector is passed through the five blocks of:
DropOut-**FC** (256 → 128)-**RELU**; **DropOut**-**FC** (128 → 64)-**RELU**; **DropOut**-**FC** (64 → 32)-**RELU**; **DropOut**-**FC** (32 → 16)-**RELU**; **DropOut**-**FC** (16 → 3)-**RELU** respectively. After the above-mentioned operation, we obtain an output of dimension $b \times 3$, which is then averaged row-wise to finally obtain a vector of size 3. These 3 values are then passed through the **Sigmoid ()** activation function to obtain 3 weight values in the range of 0 to 1.

5.2.2 Applying the sub-weights and full-weights in the AdvFD Model Here, we explain in detail how we added the sub-weights and full-weights to the partial loss functions and the total loss functions of the AdvFD model, followed by the performance evaluation.

5.2.2.1 Adding the sub-weights into the partial loss function:

In continuation with Section 3.2.1, here in this section, we define the technique to compute the weighted loss function for *"AdvFD Model"*. The *"AdvFD"* model integrates two loss functions i.e., the reconstruction loss ($\mathcal{L}_{reconstruct}^{x,\hat{x},y,\hat{y}}$), and adversarial loss ($\mathcal{L}_{adv}^{z_y}$), as depicted in eq. 19.

$$\mathcal{L}_{adversarial}^{x,\hat{x},y,\hat{y}} = \mathcal{L}_{adv}^{z_y} + \mathcal{L}_{reconstruct}^{x,\hat{x},y,\hat{y}} \tag{19}$$

In eq. 19, the sub-weights can be exclusively applied to the reconstruction loss ($\mathcal{L}_{reconstruct}^{x,\hat{x},y,\hat{y}}$). In contrast, there is no provision for introducing sub-weights into the adversarial loss ($\mathcal{L}_{adv}^{z_y}$), given its singular term $A(z_y)$, as presented in eq. 20.

$$\mathcal{L}_{adv}^{z_y} = \quad log\,(1 - A(z_y)) \tag{20}$$

The formulation of reconstruction loss remains the same as in eq. 12. Hence, three weights (i.e., $w_{z_x^1}, w_{z_x^2}, w_{z_y^1}$) are generated from each input z_x and z_y respectively. Finally, based on this sub-weights-based loss function, the weighted adversarial loss ($\mathcal{L}_{adversarial}^{x,\hat{x},y,\hat{y}}$) is computed.

[7] The concatenated vector is of dimension $3328 \times 7 \times 7$

5.2.2.2 Adding the full-weights into the total loss functions of AdvFD Model:

This model of training requires two full weights $\lambda \in \{\lambda_1, \lambda_2\}$ and two sub-weights i.e., $w_{z_x^1}$ and $w_{z_x^2}$ are required for computing the total loss. Accordingly, the aforementioned eq. 19 is revised to eq. 21.

$$\mathcal{L}^{x,\hat{x},y,\hat{y}}_{adversarial} = \mathcal{L}^{z_y}_{adv} + \mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct} \tag{21}$$

where

$$\mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct} = \lambda_1 \mathcal{L}^{x,\hat{x}}_{x,reconstruct} + \lambda_2 \mathcal{L}^{y,\hat{y}}_{y,reconstruct}$$

$$\mathcal{L}^{x,\hat{x}}_{x,reconstruct} = w_{z_x^1} \|x - \hat{x}\|_1 + (1 - w_{z_x^1})\, SSIM(x, \hat{x})$$

$$\mathcal{L}^{y,\hat{y}}_{y,reconstruct} = w_{z_y^1} \|y - \hat{y}\|_1 + (1 - w_{z_y^1})\, SSIM(y, \hat{y})$$

hence

$$\mathcal{L}^{x,\hat{x},y,\hat{y}}_{adversarial} = (1 - (\lambda_1 + \lambda_2))\mathcal{L}^{z_y}_{adv} + \lambda_1 \mathcal{L}^{x,\hat{x}}_{x,reconstruct} + \lambda_2 \mathcal{L}^{y,\hat{y}}_{y,reconstruct}$$

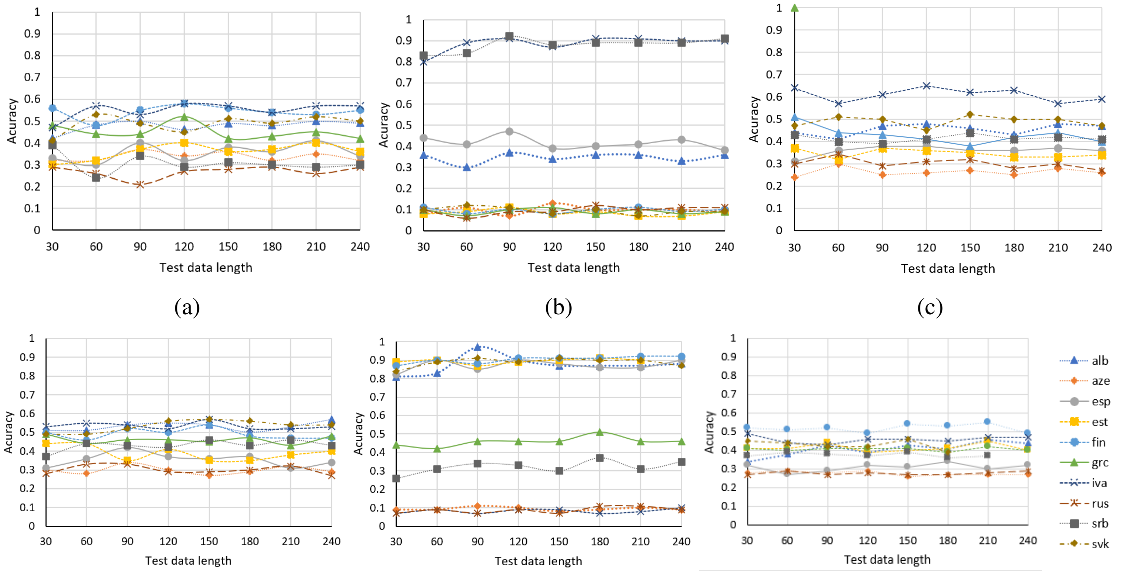
Here also the required number of sub-weights and full weights are computed by using the same techniques which are mentioned in Section 5.2.1.3.

5.2.3 The testing performance of adding sub-weights and full-weights in ContFD model: The performance of predicting the correct label (either real or fake) of the test data in terms of accuracy and F1-score after adding sub-weights into the partial loss functions and full-weights into the total loss functions in the ContFD model are shown in Figures 8 and 9. It can be seen from these results (see Figure 8) that there is no noticeable improvement can be observed between adding the sub-weights in partial loss functions and adding the full-weights in the total loss function. However, it can be seen that the accuracy is slightly better after adding full-weights over adding sub-weights for some countries like *Finland (fin)*, *Estonia (est)*, *Albania (alb)*, *Greece (grc)*, *Slovakia (svk)*.

Likewise, it can also be seen from the results of F1-score in Figure 9 that it is consistent and coherent with the accuracy results (shown in Figure 8). The F1-score results also show that there is no noticeable improvement in terms of the F1-score between adding the sub-weights into the partial loss function and adding full weights into the total loss functions. However, the F1-score results of adding full-weights over sub-weights in the ContFD-dual model show slight improvement for some countries like *Finland (fin)*, *Spain (esp)*, *Estonia (est)*, and *Slovakia (svk)*.

5.2.4 The testing performance of adding the sub-weights and full-weights in the AdvFD model: The performance of predicting the correct label (either real or fake) of the test data in terms of accuracy and F1-score after adding sub-weights into the partial loss functions and full-weights into the total loss functions in the AdvFD model are shown in Figures 10 and 11.

The accuracy results in Figure 10 show that adding the sub-weights into the partial loss functions of the “AdvFD” model (i.e., “AdvFD-SW” model) outperforms the accuracy of adding the full-weights (i.e., “AdvFD-FW” model) into the total loss functions. The accuracy results of “AdvFD-SW” in Figure 10(a) range between $0.60 - 0.90$, while the accuracy results of “AdvFD-FW” in Figure 10(b) don’t exceed 0.78. A better regularization is the potential reason, which explains that adding sub-weights into the partial loss functions of the “AdvFD” model offers better accuracy than adding full weights into the total loss functions. Indeed, by adding sub-weights into the partial loss functions, we apply better-targeted regularization to specific parts of the model. Additionally, the F1-score results in Figure 11 are coherent and consistent with the accuracy results in Figure 10. However, the F1-score results of “AdvFD-SW” in Figure 11(a) range between 0.65-0.93, and the “AdvFD-FW” in Figure 11(b) do not exceed 0.85.



(a)

(b)

(c)

(d)

(e)

(f)

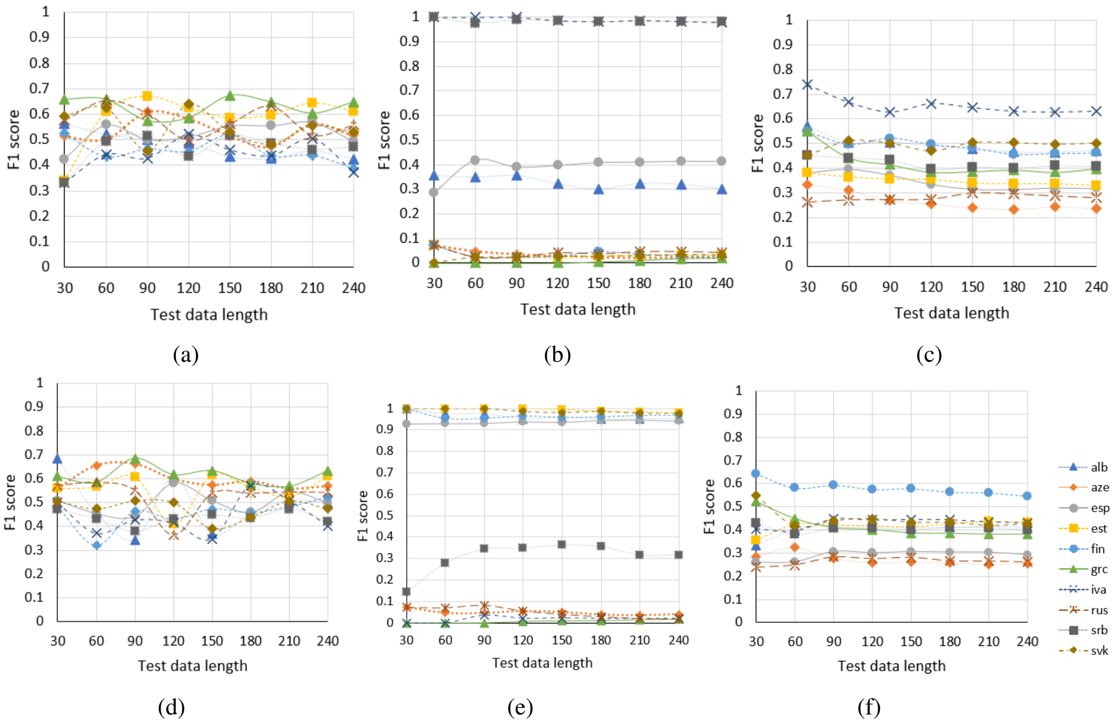
Fig. 8: Accuracy results of (a) ContFD-e2e sub-weights (SW), (b) ContFD-dual-SW), (c) ContFD-FT-SW (d) ContFD-e2e full weights (FW), (e) ContFD-dual-FW, (f) ContFD-FT-FW.

5.3 Comprehensive Study of the Performances of all the Models

In this section, we present a comprehensive study of the performances of all the proposed models. Tables 8, 9, and 10, present the accuracy values and F1 scores of all the proposed models by using 30, 150, and 240 testing samples respectively. Furthermore, Figure 12 presents the AUC results of the proposed models. In Tables 8, 9, 10, the accuracy and F1-score results are presented in ("accuracy | F1-scores") format, and the bold font represents the best accuracy and best F1-score among the proposed models.

Table 8: Inclusive (accuracy|F1-score) results for the proposed models on 30 samples.

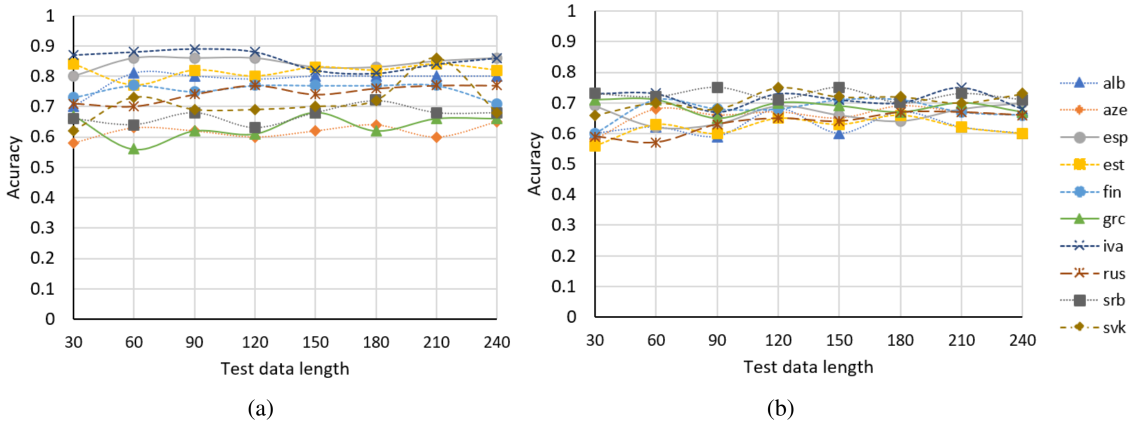
Country	Alb	Aze	Esp	Est	Fin	Grc	Iva	Rus	Srb	Svk
ContFD-e2e	0.42 | 0.62	0.33 | 0.31	0.26 | 0.29	0.30 | 0.33	0.59 | 0.31	0.37 | 0.29	0.41 | 0.38	0.31 | 0.40	0.43 | 0.40	0.39 | 0.33
ContFD-e2e-SW	0.42 | 0.56	0.32 | 0.51	0.33 | 0.42	0.30 | 0.34	0.56 | 0.53	0.48 | 0.66	0.47 | 0.33	0.29 | 0.58	0.39 | 0.33	0.41 | 0.59
ContFD-e2e-FW	0.51 | 0.68	0.30 | 0.56	0.31 | 0.51	0.44 | 0.56	0.50 | 0.48	0.49 | 0.61	0.53 | 0.50	0.28 | 0.57	0.37 | 0.47	0.49 | 0.51
ContFD-dual	**0.87** | 0.93	0.09 | **1.00**	0.74 | 0.93	0.08 | 0.05	0.83 | 1.00	0.11 | 0.05	0.77 | 0.76	0.08 | 0.07	**0.87** | **0.98**	0.10 | 0.05
ContFD-dual-SW	0.36 | 0.36	0.08 | 0.07	0.44 | 0.29	0.08 | 0.07	0.11 | 0.07	0.10 | 0.00	0.80 | **1.00**	0.10 | 0.07	0.83 | 1.00	0.10 | 0.00
ContFD-dual-FW	0.81 | **1.00**	0.09 | 0.07	**0.82** | **0.93**	**0.87** | **1.00**	**0.87** | **1.00**	0.44 | 0.00	0.07 | 0.00	0.07 | 0.07	0.26 | 0.14	**0.84** | **1.00**
ContFD-FT	0.50 | 0.36	0.34 | 0.31	0.37 | 0.43	0.36 | 0.33	0.47 | 0.57	0.51 | 0.45	0.46 | 0.50	0.43 | 0.40	0.34 | 0.36	0.38 | 0.55
ContFD-FT-SW	0.44 | 0.57	0.24 | 0.33	0.31 | 0.38	0.37 | 0.38	0.44 | 0.55	0.51 | 0.55	0.64 | 0.74	0.30 | 0.26	0.43 | 0.45	0.47 | 0.45
ContFD-FT-FW	0.34 | 0.33	0.28 | 0.29	0.32 | 0.26	0.41 | 0.36	0.52 | 0.64	0.41 | 0.52	0.49 | 0.40	0.27 | 0.24	0.37 | 0.43	0.46 | 0.55
AdvFD	0.79 | 0.96	**0.71** | 0.56	0.73 | 0.67	0.73 | 0.73	0.73 | 0.78	0.51 | 0.56	0.81 | 0.84	0.53 | 0.60	0.66 | 0.71	0.76 | 0.71
AdvFD-SW	0.70 | 0.93	0.58 | 0.69	0.80 | 0.87	0.84 | 0.91	0.73 | 0.82	0.67 | 0.69	**0.87** | 0.91	**0.71** | **0.87**	0.66 | 0.73	0.62 | 0.76
AdvFD-FW	0.60 | 0.67	0.58 | 0.71	0.69 | 0.76	0.56 | 0.69	0.60 | 0.62	**0.71** | **0.71**	0.73 | 0.84	0.59 | 0.56	0.73 | 0.84	0.66 | 0.80



(a) (b) (c)

(d) (e) (f)

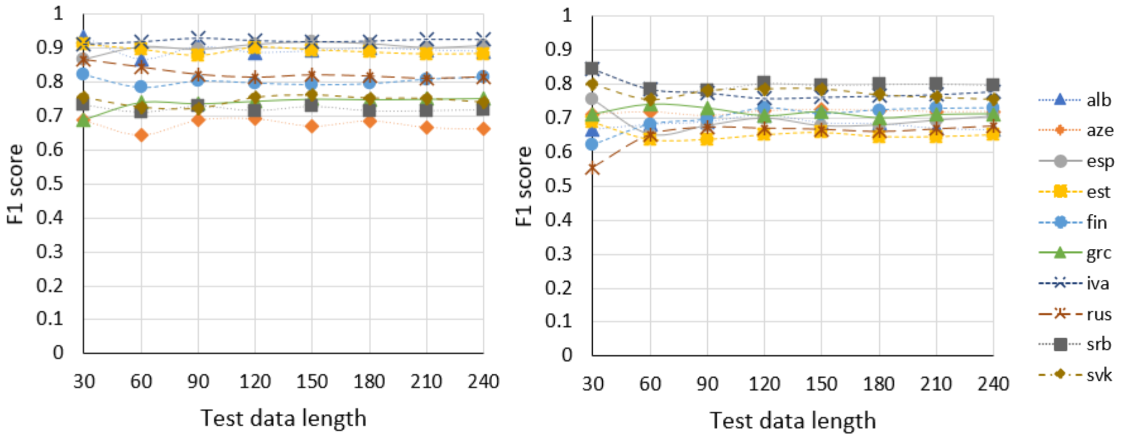
Fig. 9: F1-score results of (a) ContFD-e2e-SW, (b) ContFD-dual-SW, (c) ContFD-FT-SW, (d) ContFD-e2e-FW, (e) ContFD-dual-FW, (f) ContFD-FT-FW.



(a) (b)

Fig. 10: Accuracy results of (a) AdvFD-SW, (b) AdvFD-FW.

According to the aforementioned results in Tables 8, 9, and 10, the ”ContFD-dual” and ”AdvFD” models achieved the best testing performance for almost all of the countries. More specifically, the ”ContFD-dual”



(a)

(b)

Fig. 11: F1-score results of (a) AdvFD-SW, (b) AdvFD-FW.

Table 9: Inclusive (accuracy|F1-score) results for the proposed models on 150 samples.

Country	Alb	Aze	Esp	Est	Fin	Grc	Iva	Rus	Srb	Svk
ContFD-e2e	0.50 | 0.48	0.25 | 0.30	0.30 | 0.25	0.28 | 0.31	0.52 | 0.32	0.38 | 0.38	0.48 | 0.42	0.34 | 0.30	0.44 | 0.35	0.40 | 0.41
ContFD-e2e-SW	0.49 | 0.43	0.36 | 0.52	0.38 | 0.56	0.36 | 0.59	0.56 | 0.52	0.42 | 0.67	0.57 | 0.46	0.28 | 0.56	0.31 | 0.52	0.51 | 0.53
ContFD-e2e-FW	0.54 | 0.37	0.27 | 0.57	0.36 | 0.51	0.35 | 0.61	0.54 | 0.47	0.45 | 0.63	0.57 | 0.35	0.29 | 0.54	0.46 | 0.45	0.57 | 0.39
ContFD-dual	**0.90** | **0.98**	0.10 | **0.98**	0.84 | 0.92	0.07 | 0.02	0.88 | **0.98**	0.20 | 0.07	0.76 | 0.80	0.08 | 0.02	**0.90** | **0.98**	0.08 | 0.02
ContFD-dual-SW	0.36 | 0.30	0.10 | 0.02	0.40 | 0.41	0.09 | 0.03	0.10 | 0.05	0.08 | 0.00	**0.91** | **0.98**	0.12 | 0.04	0.89 | 0.98	0.10 | 0.03
ContFD-dual-FW	0.87 | 0.96	0.08 | 0.05	**0.88** | **0.94**	**0.89** | **1.00**	**0.91** | 0.96	0.46 | 0.01	0.09 | 0.02	0.07 | 0.04	0.30 | 0.36	**0.91** | **0.98**
ContFD-FT	0.46 | 0.49	0.28 | 0.24	0.38 | 0.40	0.36 | 0.47	0.46 | 0.47	0.47 | 0.44	0.52 | 0.52	0.37 | 0.39	0.34 | 0.29	0.41 | 0.40
ContFD-FT-SW	0.46 | 0.48	0.27 | 0.24	0.36 | 0.31	0.35 | 0.34	0.47 | 0.48	0.38 | 0.39	0.62 | 0.65	0.32 | 0.30	0.44 | 0.40	0.52 | 0.50
ContFD-FT-FW	0.43 | 0.41	0.26 | 0.26	0.31 | 0.31	0.40 | 0.42	0.54 | 0.58	0.41 | 0.39	0.46 | 0.45	0.27 | 0.28	0.37 | 0.40	0.42 | 0.43
AdvFD	0.86 | 0.82	**0.68** | 0.64	0.72 | 0.72	0.83 | 0.77	0.75 | 0.71	0.54 | 0.53	0.82 | 0.80	0.56 | 0.58	0.68 | 0.66	0.74 | 0.69
AdvFD-SW	0.80 | 0.89	0.62 | 0.67	0.83 | 0.92	0.83 | 0.89	0.77 | 0.79	0.68 | **0.75**	0.82 | 0.92	**0.74** | **0.82**	0.68 | 0.73	0.70 | 0.76
AdvFD-FW	0.60 | 0.69	0.65 | 0.72	0.66 | 0.68	0.63 | 0.66	0.71 | 0.71	**0.69** | 0.72	0.71 | 0.76	0.64 | 0.67	0.75 | 0.80	0.72 | 0.79

Table 10: Inclusive (accuracy|F1-score) results for the proposed models on 240 samples.

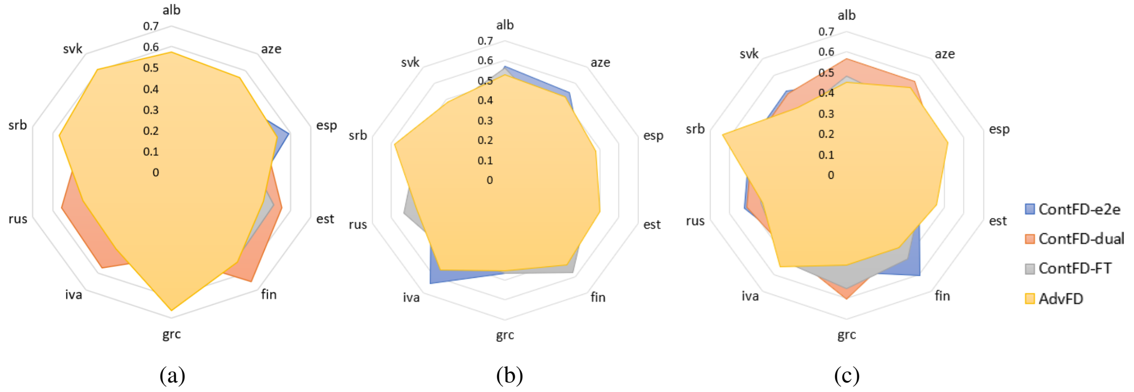
Country	Alb	Aze	Esp	Est	Fin	Grc	Iva	Rus	Srb	Svk
ContFD-e2e	0.48 | 0.48	0.29 | 0.29	0.32 | 0.26	0.30 | 0.29	0.57 | 0.32	0.40 | 0.38	0.50 | 0.42	0.34 | 0.32	0.47 | 0.32	0.43 | 0.43
ContFD-e2e-SW	0.49 | 0.43	0.32 | 0.52	0.34 | 0.49	0.36 | 0.61	0.55 | 0.39	0.42 | 0.65	0.57 | 0.37	0.29 | 0.56	0.30 | 0.47	0.50 | 0.53
ContFD-e2e-FW	0.57 | 0.50	0.29 | 0.57	0.34 | 0.50	0.40 | 0.61	0.47 | 0.52	0.48 | 0.63	0.53 | 0.40	0.27 | 0.54	0.43 | 0.42	0.54 | 0.47
ContFD-dual	**0.88** | **0.98**	0.11 | **0.98**	0.85 | 0.92	0.12 | 0.02	0.91 | **0.98**	0.16 | 0.07	0.77 | 0.81	0.08 | 0.03	**0.91** | **0.98**	0.09 | 0.02
ContFD-dual-SW	0.36 | 0.30	0.09 | 0.02	0.38 | 0.41	0.09 | 0.03	0.10 | 0.03	0.09 | 0.02	**0.90** | **0.98**	0.11 | 0.04	0.91 | 0.98	0.09 | 0.03
ContFD-dual-FW	0.88 | 0.95	0.09 | 0.04	**0.90** | **0.94**	**0.90** | **0.98**	**0.92** | 0.97	0.46 | 0.02	0.10 | 0.02	0.09 | 0.02	0.35 | 0.32	**0.87** | **0.98**
ContFD-FT	0.47 | 0.48	0.28 | 0.24	0.42 | 0.42	0.33 | 0.44	0.47 | 0.47	0.40 | 0.44	0.51 | 0.52	0.44 | 0.40	0.33 | 0.31	0.39 | 0.39
ContFD-FT-SW	0.47 | 0.47	0.26 | 0.24	0.36 | 0.32	0.34 | 0.33	0.47 | 0.46	0.40 | 0.40	0.59 | 0.63	0.27 | 0.28	0.41 | 0.41	0.47 | 0.50
ContFD-FT-FW	0.44 | 0.41	0.27 | 0.26	0.32 | 0.29	0.40 | 0.43	0.49 | 0.55	0.40 | 0.38	0.47 | 0.43	0.29 | 0.26	0.37 | 0.40	0.39 | 0.43
AdvFD	0.88 | 0.81	0.65 | 0.63	0.73 | 0.69	0.81 | 0.76	0.75 | 0.71	0.51 | 0.53	0.89 | 0.80	0.61 | 0.57	0.74 | 0.66	0.76 | 0.72
AdvFD-SW	0.80 | 0.89	0.65 | 0.66	0.86 | 0.91	0.82 | 0.88	0.71 | 0.81	0.66 | **0.75**	0.86 | 0.93	**0.77** | **0.81**	0.68 | 0.72	0.68 | 0.74
AdvFD-FW	0.60 | 0.67	**0.66** | 0.72	0.70 | 0.71	0.60 | 0.65	0.66 | 0.73	**0.67** | 0.71	0.68 | 0.78	0.66 | 0.68	0.71 | 0.80	0.73 | 0.76

model achieved high testing performance for countries like *Alb* and *Srb*, while the "ContFD-dual-FW" model achieved higher testing performance for countries like Esp, Est, Fin, and Svk. The ”AdvFD-SW” model achieved a higher testing performance for Rus, while the two configurations of the AdvFD model (i.e., by adding sub-weights and full-weights) achieved better performance for Greece. In terms of F1 scores, the ”ContFD-dual” model achieved better results, while the AdvFD model achieved better accuracy for some countries. The ”ContFD-e2e” and ”ContFD-FT” models were not able to outperform the ”ContFD-dual” and ”AdvFD” models, which showed the best testing performance for almost all of the countries.

Here are some specific observations from the results:

- ContFD-dual achieved the highest accuracy (0.87-0.91) for all sample sizes in Alb, Srb, and several other countries. It also achieved the highest F1 score (0.93-1.00) in Alb and Srb for all sample sizes.
- AdvFD achieved the highest accuracy (0.79-0.89) for Grc in all sample sizes. It also achieved the highest F1 score (0.82-0.96) for Aze with 30 samples.
- ContFD-dual-FW achieved the highest accuracy (0.87-0.92) for Est, Fin, and Svk with 150 and 240 samples. It also achieved the highest F1 score (1.00) for Fin with 150 samples.
- AdvFD-SW achieved the highest accuracy (0.74-0.77) for Rus with 150 and 240 samples. It also achieved the highest F1 score (0.87) for Rus with 30 samples.

It is important to note that the best-performing model can vary depending on the specific country and evaluation metric (accuracy vs. F1 score). However, both ”ContFD-dual” and ”AdvFD” consistently achieved strong performance across a variety of testing conditions, making them promising candidates for real-world applications.



(a) (b) (c)

Fig. 12: AUC results of the proposed models (a) without adding weights, (b) after adding sub-weights, (c) after adding full weights.

Furthermore, in Figure 12, the AUC results of the various countries via different configurations of the proposed models (with/without adding weights into the loss functions) are shown. It can be seen from the results that the AUC values range between $0.55 - 0.65$. More importantly, it can be seen from the AUC results that the testing performance of both the “ContFD-dual” and “AdvFD” models are superior to the “ContFD-e2e” and “ContFD-FT” models for most of the countries. The analysis of results from AUC curves is fully coherent with the results mentioned in Tables 8, 9, 10.

6 Comparative Study and Discussion

This section presents a comprehensive comparison and discussion of the performance of the proposed models in this paper, with other relevant studies in the literature. Due to the limited overlap in the characteristics and methodologies of the proposed models and the other works, particularly those listed in Tables 2 and 3, this comparison will be conducted across multiple disciplines.

6.1 Performance Comparison

To further evaluate the effectiveness of the proposed $ContFD$ and $AdvFD$ models, we undertake a comparative analysis against the CFD and $FsAFD$ models introduced in [1]. This comparison is particularly relevant as [1] is the only existing work that aligns with our models in terms of the security objects to be verified (i.e., guilloche) and utilizes similar datasets (i.e., MIDV and FMIDV) for performance evaluation.

Table 11 provides a detailed comparison between our research in this paper and the previous study conducted by [1] using sample sizes of 30, 150, and 240, focusing on the metrics of accuracy and F1-score. This comparative analysis assesses the performance of our best model/s using datasets comprising 30, 150, and 240 samples, as outlined in the comprehensive results showcased in Tables 8, 9, and 10.

Referring to the results in Table 11, a notable disparity emerges in the performance of our proposed models in this paper compared to the models presented in the previous work [1], particularly in terms of accuracy and F1-score. This performance gap remains consistent across all sample sizes, i.e, 30, 150, and 240. Remarkably, the performance tests reveal a substantial discrepancy, with accuracy differing by more than 40% and F1-score by over 30% for all countries.

Upon examining the data in Table 11, we draw several noteworthy conclusions regarding the performance of various learning schemes. The ContFD model, utilizing a dual learning approach, known as ContFD-dual, emerges as the top-performing (Excellent) model for all countries except Grc and Rus. In the cases of Greece (Grc) and Russia (Rus), the AdvFD model, enhanced with sub-weights (AdvFD-SW) within its objective function, stands out as the optimal choice.

Furthermore, the results presented in Table 11 affirm the advantages of incorporating weights into the objective functions of the ContFD-dual model. This enhancement significantly improves the overall performance across all eight countries. Specifically, when full weights are integrated into the objective function of the ContFD-dual model (ContFD-dual-FW), it becomes the best-performing (excellent) model for the countries Esp, Est, Fin, and Svk. Conversely, introducing sub-weights into the ContFD model-based dual learning scheme (ContFD-dual-SW) yields the best results for the Iva country.

In Figure 13, we present a comparative analysis showcasing the AUC results of our top-performing models in comparison to the models introduced in [1].

The results displayed in Fig. 13 reveal a significant disparity in AUC performance between our top-performing models and the models introduced in [1] across most countries. Our proposed models in this paper surpass the AUC results achieved by the models in [1] (CFD and FsAFD) for all countries except Est and Rus. However, it is worth noting that the performance gap in terms of AUC is substantial, exceeding 10% for all countries except Est and Rus. In contrast, for Est and Rus, the AUC performance of all models shows remarkable convergence, with the margin in performance not exceeding 1%.

While AUC provides a valuable performance metric, it's not the sole factor when choosing a model for a specific task. It is essential to assess other metrics like accuracy, and F1-score and consider the specific requirements of the targeted application.

Table 11: Comparison of accuracy and F1-scores.

Country		30 samples	150 samples	240 samples
	Models	**accuracy|F1**	**accuracy|F1**	**accuracy|F1**
Alb	**ContFD-dual**	**0.87** | **0.93**	**0.90** | **0.98**	**0.88** | **0.98**
	CFD in [1]	0.46 | 0.63	0.58 | 0.68	0.56 | 0.64
	FsAFD in [1]	0.51 | 0.65	0.57 | 0.70	0.57 | 0.70
Aze	**ContFD-dual**	0.09 | **1.00**	0.10 | **0.98**	0.11 | **0.98**
	AdvFD	**0.71** | 0.56	**0.68** | 0.64	**0.65** | 0.63
	CFD in [1]	0.59 | 0.72	0.56 | 0.64	0.58 | 0.57
	FsAFD in [1]	0.54 | 0.71	0.56 | 0.71	0.55 | 0.71
Esp	**ContFD-dual-FW**	**0.82** | **0.93**	**0.88** | **0.94**	**0.90** | **0.94**
	CFD in [1]	0.50 | 0.59	0.44 | 0.59	0.46 | 0.57
	FsAFD in [1]	0.56 | 0.71	0.57 | 0.73	0.56 | 0.72
Est	**ContFD-dual-FW**	**0.87** | **1.00**	**0.89** | **1.0**	**0.90** | **0.98**
	CFD in [1]	0.48 | 0.70	0.57 | 0.66	0.58 | 0.65
	FsAFD in [1]	0.62 | 0.77	0.57 | 0.73	0.57 | 0.73
Fin	**ContFD-dual-FW**	**0.87** | **1.00**	**0.91** | **0.96**	**0.92** | **0.97**
	CFD in [1]	0.47 | 0.42	0.48 | 0.58	0.51 | 0.56
	FsAFD in [1]	0.61 | 0.72	0.56 | 0.69	0.56 | 0.69
Grc	**AdvFD-SW**	**0.67** | 0.69	**0.68** | **0.75**	**0.66** | **0.75**
	CFD in [1]	0.54 | 0.62	0.57 | 0.62	0.55 | 0.66
	FsAFD in [1]	0.56 | **0.71**	0.55 | 0.69	0.55 | 0.69
Iva	**ContFD-dual-SW**	**0.80** | **1.00**	**0.91** | **0.98**	**0.90** | **0.98**
	CFD in [1]	0.34 | 0.50	0.42 | 0.54	0.41 | 0.65
	FsAFD in [1]	0.60 | 0.75	0.58 | 0.73	0.57 | 0.73
Rus	**AdvFD-SW**	**0.71** | **0.87**	**0.74** | **0.82**	**0.77** | **0.81**
	CFD in [1]	0.46 | 0.52	0.47 | 0.49	0.46 | 0.48
	FsAFD in [1]	0.61 | 0.73	0.57 | 0.70	0.56 | 0.70
Srb	**ContFD-dual**	**0.87** | **0.98**	**0.90** | **0.98**	**0.91** | **0.98**
	CFD in [1]	0.52 | 0.53	0.56 | 0.62	0.57 | 0.60
	FsAFD in [1]	0.51 | 0.68	0.57 | 0.72	0.58 | 0.73
Svk	**ContFD-dual-FW**	**0.84** | **1.00**	**0.91** | **0.98**	**0.87** | **0.98**
	CFD in [1]	0.43 | 0.63	0.54 | 0.59	0.53 | 0.57
	FsAFD in [1]	0.52 | 0.69	0.57 | 0.72	0.57 | 0.72

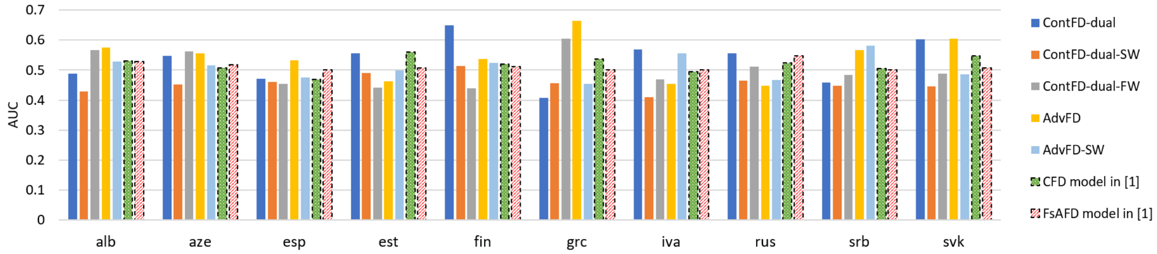


Fig. 13: Comparative analysis of AUC results.

6.2 Systematic Comparisons

Our research represents the pioneering effort in developing a forgery detection model tailored to IDs, specifically focusing on guilloche patterns. Notably, there is a notable absence of prior experiments or reported results concerning the MIDV and the FMIDV datasets within the existing literature, rendering a comparative study impractical. Consequently, our study serves as foundational groundwork for future endeavors in this domain. Nevertheless, the novelty of our approach can be underscored by conducting a systematic comparison between our work and pertinent studies referenced in [1], [5–7], [12–18], which are deemed highly relevant to our proposal. Key facets for comparison include the generality of the learning scheme, the necessity of the original document as a reference for forgery detection, the simplicity predicated on the number of utilized CNNs, the establishment of a new dataset, and the complexity. Table 12 provides a comparative analysis of state-of-the-art forgery detection methods for documents and IDs and the proposed models. To assess the efficacy of the forgery detection system, we will employ a standardized evaluation criterion to gauge its performance, typically measured in terms of accuracy or other relevant metrics. Accordingly, if the model achieves a performance level between 70% and below 80%, it will be categorized as a ”fair” model. If the performance falls within the range of 80% to below 90%, the model will be deemed ”good”. Finally, if the performance exceeds or equals 90%, the model will be classified as ”very good”.

Table 12: Comparative analysis of SOTA methods of forgery detection for IDs w.r.t the proposed models.

Characteristics	**SOTA Techniques**											
	[1]	**[12]**	**[13]**	**[14]**	**[15]**	**[5]**	**[16]**	**[17]**	**[18]**	**[6]**	**[7]**	**Proposed Method**
Threshold setting	No	Yes	Yes	Yes	No	No	No	No	No	Yes	Yes	No
Dependency on knowing pattern nature	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
Dependency on pre-processing	No	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	No
Dependency on visual characteristics	No	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	No
Sensitivity to image quality	No	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes Yes	No	
Requirement of the original pattern/document (as a reference)	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
End-to-end solution	Yes	No	No	No	Yes	No	No	No	No	No	No	Yes
Forgery detection performance	Low (fair model)	Average (good model)	High (better model)	High (better model)	Average (good model)	Average (good model)	Average (good model)	Average (good model)	Low (fair model)	High (better model)	Average (good model)	High (better model)
Generality for different types of ID	Yes	No	No	No	No	No	No	No	No	No	No	Yes
Simplicity	Yes (used 1 CNN)	Yes	Yes	Yes	Yes	Yes (used 1 CNN)	Yes (used 3 CNN)	Visual inspection	Visual inspection	Yes	Yes	Yes (used 1 CNN)
Creation of new dataset	Yes	No	No	No	No	No	Yes	No	No	No	No	No
Types of processed documents	Global IDs	Limited on Russian passport	Limited (Italian, French IDs)	Limited to Colombian IDs	Limited to Azerbaijani IDs of MIDV	Limited to French IDs	Limited to Spain IDs and banknotes	Global IDs	Global IDs	Limited to MIDV500	Limited French passport	Global IDs (MIDV and FMIDV)
Complexity	Average	Low	Low	High	High	Average	High	High	High	Average	Average	Average

Table 12 provides a comprehensive comparative analysis of state-of-the-art forgery detection methods for documents and IDs, alongside our proposed models. Our models offer end-to-end solutions without requiring the original pattern or document as a reference, ensuring a high level of generality across different types of IDs. Unlike several existing methods, our approach does not depend on pre-processing or visual characteristics, making it robust to variations in image quality. Moreover, our models employ a single CNN for simplicity and have demonstrated high forgery detection performance, outperforming many existing methods. Additionally, our work contributes to the field by introducing new datasets and addressing the complexity associated with forgery detection in IDs.

6.3 The Pros and Cons of the Proposed Models

According to the mentioned elements in Table 13, the proposed models offer several advantages.

ContFD provides robust feature representation through an encoder-decoder structure, facilitating contrastive learning for discrimination and ensuring high-quality reconstruction. Additionally, its multi-objective training enhances the model's comprehensiveness. AdvFD, on the other hand, leverages a constrained adversarial network to enhance the quality of latent spaces, leading to improved discriminatory features. However, both models suffer from complexity due to their intricate architectures, which can pose challenges during training and deployment. Moreover, the interpretability of AdvFD may be compromised due to the adversarial nature of its training, making it difficult to interpret the decision-making process.

6.4 Complexity Analysis

In the following section, we have categorically analyzed the complexities of the proposed ContFD and AdvFD models and the previously introduced CFD and FsAFD models, outlined in [1]. The following complexity analysis reveals several key insights into the computational demands of the proposed ContFD and AdvFD models compared to the previously introduced CFD and FsAFD models. Both ContFD and AdvFD models exhibit higher computational complexity primarily due to their intricate architectures involving encoder-decoder networks and adversary networks. Specifically, the ContFD model employs a dense neural network architecture for feature extraction and reconstruction, contributing to a significant increase in computational requirements, as evidenced by the substantially higher number of FLOPS and parameters compared to the CFD and FsAFD models. Similarly, the AdvFD model introduces additional computational overhead with its inclusion of an adversary network for adversarial training, further amplifying the complexity. While these models offer enhanced discriminatory features and improved representations, their computational demands pose challenges in terms of training and deployment. **Method Name :** CFD model [1]

I. **Complexity Analysis :** The complexity analysis of the CFD model involves considering the computational cost associated with training and making predictions. We break down the complexity analysis into different components:
   i. Encoder Network $E_\theta(.)$ (i.e. Siamese Neural Network) :
      ❐ *Forward Pass:* It involves computing the latent feature vectors $z_x = E_\theta(x)$ and $z_y = E_\theta(y)$ for a pair of IDs (x,y). The encoder network consists of fully connected layers with d dimensions, the complexity is typically $O(d)$.
      ❐ *Contrastive Loss:* It involves computing the distance between $E_\theta(x)$ and $E_\theta(y)$ based on the specified conditions. This distance calculation has a complexity of $O(d)$.
      ❐ *Backward Pass (Gradient Descent):* The back-propagation involves computing of gradients, concerning the parameters of the encoder network. The complexity is $O(d)$ per parameter.

Table 13: The pros and cons of the proposed models.

Method	**Pros**	**Cons**
ContFD	❒ *Robust feature representation:* It employs an encoder-decoder structure to map input images into lower-dimensional feature vectors and helps in extracting robust and meaningful representations of IDs. ❒ *Contrastive learning for discrimination:* It facilitates discrimination between genuine and fake IDs. ❒ *Reconstruction quality:* The inclusion of point-wise $L1$ loss and $SSIM$ loss in the reconstruction objective ensures that the decoder generates high-quality reconstructed images. ❒ *Multi-Objective training:* helps address various aspects of the forgery detection task, making the model more comprehensive. ❒ *Enforcement of similarity constraints:* The contrastive loss enforces constraints on the distribution of the model's inner representation, encouraging similar features for samples from the same class and dissimilar features for samples from different classes. This enhances the discriminative power of the model.	❒ *Complexity:* The model's architecture, involving an encoder-decoder and a classifier network, introduces complexity in terms of training, and deployment.
AdvFD	❒ A constrained adversarial network $(A(.))$ enhances the quality of latent spaces by maximizing the distance between the latent spaces for real and fake IDs. This can contribute to improved discriminatory features. ❒ Employing the constrained adversarial network implicitly improves the quality of latent spaces by focusing on the latent space (z_y) generated from fake images. This, in turn, minimizes the difference between the input and the reconstructed image, leading to better representations. ❒ Adversarial loss $(\mathcal{L}^{z_y}_{adv})$ for discrimination: helps in the discrimination process by penalizing the network for misclassifying a fake instance as real and rewarding correct classification. This promotes the ability of the discriminator to detect features indicative of forgery.	❒ *Complexity:* Like the ContFD model, the AdvFD model involves a complex architecture, which may introduce challenges in terms of training, and interpretability. ❒ *Interpretability:* Understanding the decision-making process of an adversarial model can be challenging. The adversarial nature of the training might make it harder to interpret the features contributing to the model's decisions.

ii. Classifier Network $f(.)$:

❒ *Forward Pass:* It includes computing the classification probabilities based on the latent feature vectors. The classifier consists of fully connected layers with d dimensions, the complexity is typically $O(d)$.

❒ *Cross-Entropy Loss (i.e., $\mathcal{L}^{\ell}_{cross}$):* It involves computing the difference between predicted probabilities and true labels. The complexity is $O(1)$ for each sample, making it $O(N)$ for N samples.

❒ *Backward Pass:* It involves computing gradients concerning its parameters. The complexity is usually $O(d)$ per parameter.

iii. Training Iterations :

❒ The overall training complexity is influenced by the number of training iterations. Each iteration involves a forward and backward pass for both the encoder and the classifier.

II. **Speed and Number of Parameters:**

❒ *GFLOPS:* 0.10253851

❒ *Number of parameters (million):* 46.091677

Method Name : FsAFD model [1]

I. **Complexity Analysis :** The complexity analysis of the FsAFD model involves considering the computational cost associated with training and making predictions. We break down the complexity analysis into different components:

i. Encoder Network $E_\theta(.)$ (i.e. Siamese Neural Network)

❒ *Forward Pass:* Similar to the encoder in CFD model, the complexity is typically $O(d)$.

❒ *Contrastive Loss:* Similar to the CFD model, the complexity of this operation is $O(d)$.

❒ *Backward Pass:* The complexity is $O(d)$ per parameter.

ii. Adversary Network (Fake-sample Adversary):

❒ *Forward Pass:* It involves computing the predicted probabilities based on the encoded representation $E_\theta(y)$. The adversary consists of fully connected layers with d dimensions, the complexity is typically $O(d)$.

❒ *Adversary Loss:* It involves computing the loss based on the predicted probabilities and the contrary target labels. The complexity is $O(N)$ for N samples.

❒ *Backward Pass:* It involves computing gradients concerning its parameters. The complexity is usually $O(d)$ per parameter.

iii. Training Iterations :

❒ The overall training complexity is influenced by the number of training iterations. Each iteration involves a forward and backward pass for both the encoder and the adversary.

II. **Speed and Number of Parameters:**

- *GFLOPS:* 0.102541264
- *Number of parameters (million):* 46.094458

Method Name : Proposed ContFD model

I. **Complexity Analysis :** The complexity analysis of the ContFD model involves considering the computational cost associated with training and making predictions. We break down the complexity analysis into different components:

i. Encoder Network (i.e. DenseNet Neural Network)

❒ *Forward Pass:* It involves computing the latent feature vectors $z_x = E_\theta(x)$ and $z_y = E_\theta(y)$ for a pair of real and fake IDs. If the encoder network consists of fully connected layers with d dimensions, the complexity is typically $O(d)$.

❒ *Contrastive loss (i.e., $\mathcal{L}^{z_x,z_y,\ell_x,\ell_y}_{contrastive}$):* It involves computing the loss based on the specified conditions to discriminate between z_x and z_y. The complexity is $O(d)$.

❒ *Backward Pass:* The complexity is usually $O(d)$ per parameter.

ii. Decoder Network

❒ *Forward Pass:* It involves reconstructing the input image pair from the latent spaces z_x and z_y. The complexity is $O(d)$.

- *Reconstruction loss (i.e., $\mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct}$):* It involves computing the point-wise $L1$ loss and $SSIM$ loss for image reconstruction. The complexity is $O(d)$.
- *Backward Pass:* The back-propagation for the decoder involves computing gradients concerning its parameters. The complexity is usually $O(d)$ per parameter.

iii. Classifier Network:
- *Forward Pass:* It involves classifying the latent feature vectors z_x and z_y into the class of "real" or "fake". If the classifier consists of fully connected layers with d dimensions, the complexity is typically $O(d)$.
- *Cross-Entropy Loss (i.e., $\mathcal{L}^{\ell}_{cross}$):* It involves computing the difference between predicted probabilities and true labels. The complexity is $O(1)$.

iv. Training Iterations :
- The overall training complexity is influenced by the number of training iterations. Each iteration involves a forward and backward pass for both the encoder, decoder, and classifier.

II. **Speed and Number of Parameters:**
- *GFLOPS:* 23.109468608
- *Number of parameters (million):* 126.286971

Method Name : Proposed AdvFD model

I. **Complexity Analysis :** The complexity analysis of the AdvFD model involves considering the computational cost associated with training and making predictions. We break down the complexity analysis into different components:

i. Encoder Network (i.e. DenseNet Neural Network) :
- *Forward Pass:* It involves computing the latent feature vectors $z_x = E_\theta(x)$ and $z_y = E_\theta(y)$ for a pair of real and fake IDs. If the encoder network consists of fully connected layers with d dimensions, the complexity is typically $O(d)$.
- *Backward Pass:* The complexity is usually $O(d)$ per parameter.

ii. Decoder Network :
- *Forward Pass:* It involves reconstructing the input image pair from the latent spaces $z_x = E_\theta(x)$ and $z_y = E_\theta(y)$. The complexity is $O(d)$.
- *Reconstruction loss (i.e., $\mathcal{L}^{x,\hat{x},y,\hat{y}}_{reconstruct}$):* It involves computing the point-wise $L1$ loss and $SSIM$ loss for image reconstruction. The complexity is $O(d)$.
- *Backward Pass:* The back-propagation for the decoder involves computing gradients concerning its parameters. The complexity is usually $O(d)$ per parameter.

iii. Adversary Network (Fake-sample Adversary):
- *Forward Pass:* It involves computing the predicted probabilities based on the encoded representation E_y. If the adversary consists of fully connected layers with d dimensions, the complexity is typically $O(d)$.
- *Adversary Loss (i.e., $\mathcal{L}^{x,\hat{x},y,\hat{y},z_y}_{adversarial}$):* It involves computing the loss based on the predicted probabilities and the contrary target labels. The complexity is $O(N)$ for N samples.
- *Backward Pass:* The back-propagation for the adversary involves computing gradients concerning its parameters. The complexity is usually $O(d)$ per parameter.

iv. Training Iterations :
- The number of training iterations influences the overall training complexity. Each iteration involves a forward and backward pass for both the encoder and the adversary.

II. **Speed and Number of Parameters:**

- *GFLOPS:* 23.109468608
- *Number of parameters (million):* 126.286971

6.5 The Limitations and Challenges

This section delves into the limitations and challenges that impede scientific researchers in their pursuit of novel and impactful research in the realm of ID forgery detection.

- *Lack of public datasets:* The study is limited by the lack of publicly available datasets of forged IDs. This makes it difficult to train and evaluate the proposed forgery detection method on a variety of forged documents.
- *Difficulty in simulating forgeries:* It is challenging to simulate all possible forgeries that could be applied to IDs. This includes artificially created forgeries that may not be detectable by the naked eye.
- *Range of security features:* The wide range of security features incorporated into modern IDs, such as guilloches, holograms, micro-prints, watermarks, and optical effects, contributes to multiple layers of identity security. However, this diversity also poses challenges in designing and implementing effective innovative solutions for enhancing identity security.
- *Limited comparative studies:* The field of forgery detection on IDs is relatively young and a limited body of research is available for comparison. This makes it difficult to assess the performance of the proposed method definitively.
- *Focus on private datasets:* Many existing studies in this domain have been conducted using private datasets, which makes it difficult to compare results and establish the method's generalization ability.

The limitations of current forgery detection methods suggest the need for further research to develop more robust and effective approaches. This includes efforts to expand and diversify the datasets used for training and testing these methods. Additionally, it is crucial to develop techniques for simulating a broader range of forgeries, enabling the development of more comprehensive and generalizable detection algorithms.

7 Conclusions and Future Works

In this paper, we have presented two novel ID verification models based on contrastive learning and adversarial learning, respectively, for detecting fraudulent IDs using guilloche patterns. The models leverage the distinctive features of guilloche patterns to effectively distinguish between genuine and forged documents. The proposed models, ContFD and AdvFD, achieve remarkable performance, with accuracy and F1-score surpassing the results reported in the existing literature. Comparative analysis reveals that ContFD-dual emerges as the overall best-performing model, outperforming the previous models, CFD and FsAFD, across all sample sizes and countries except Grc and Rus. For these two countries, AdvFD-SW outperforms the other models. The incorporation of weights into the objective functions of both ContFD and AdvFD models further enhances their performance.

The contributions of this paper include: (i) introducing two novel ID verification models based on contrastive learning and adversarial learning for detecting fraudulent IDs using guilloche patterns. (ii) employing contrastive learning to capture the discriminative features of guilloche patterns and distinguish between genuine and forged documents. (iii) utilizing adversarial learning to improve the model's generalization capability and robustness against adversarial attacks. (iv) achieving state-of-the-art performance in ID verification using guilloche patterns.

Future work in this field should focus on addressing the limitations identified above. One important area of research is to develop publicly available datasets of forged IDs. This would allow researchers to train and evaluate their methods on a wider range of forgeries and make it easier to compare results. Additionally, researchers should continue to develop methods for simulating forgeries that are more realistic and challenging. This would help to ensure that the methods developed are robust to a wider range of forgeries. Another important area of research is to develop multi-modal forgery detection methods that can analyze multiple security features simultaneously. This would be more effective than single-modal methods, which can only analyze one security feature at a time. Additionally, researchers should continue to develop comparative studies that evaluate the performance of different forgery detection methods. This would help to establish the best practices for forgery detection and make it easier for researchers to choose the right method for their applications. Finally, researchers should focus on developing methods that can be deployed in real-time. This would be valuable for applications such as border control and identity verification.

Funding This work has been financed by (le fonds unique interministériel) FUI IDECYS+ project (No.:DOS0098984/00)

Data Availability Both the data and forgery detection code are made available at `https://github.com/malghadi/CheckID`.

Declarations

- The ID samples and the photos of individuals throughout this manuscript are sourced exclusively from the publicly available MIDV dataset.
- We have no conflict of interest to declare.

References

1. M. Al-Ghadi, Z. Ming, P. Gomez-Krämer, J.-C. Burie, M. Coustaty, and N. Sidere, "Guilloche detection for ID authentication: A dataset and baselines," in *Proceedings of the International Workshop on MultiMedia Signal Processing (MMSP)*. IEEE, 2023, pp. 1–6.
2. C. Jung, G. Kim, M. Jeong, J. Jang, Z. Dong, T. Badloe, J. K. Yang, and J. Rho, "Metasurface-driven optically variable devices," *Chemical Reviews*, vol. 121, pp. 13 013–13 050, 11 2021.
3. P. J. Stepien, R. Gajda, and A. Marszalek, "Guilloche in diffractive optically variable image devices," in *Optical Security and Counterfeit Deterrence Techniques II, SPIE Conference Series*, vol. 3314. SPIE, 4 1998, pp. 231–236.
4. C. L. Li, K. Sohn, J. Yoon, and T. Pfister, "Cutpaste: Self-supervised learning for anomaly detection and localization," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 9659–9669, 2021.
5. N. Ghanmi, C. Nabli, and A. M. Awal, "Checksim: A reference-based identity document verification by image similarity measure," in *Document Analysis and Recognition – ICDAR 2021 Workshops*, vol. 12916 LNCS. Springer Science and Business Media Deutschland GmbH, 2021, pp. 422–436.

6. O. Kada, C. Kurtz, C. van Kieu, and N. Vincent, "Hologram detection for identity document authentication," in *Pattern Recognition and Artificial Intelligence*, M. El Yacoubi, E. Granger, P. C. Yuen, U. Pal, and N. Vincent, Eds. Cham: Springer International Publishing, 2022, pp. 346–357.
7. M.-N. Chapel, M. Al-Ghadi, and J.-C. Burie, "Authentication of holograms with mixed patterns by direct LBP comparison," in *Proceedings of the International Workshop on MultiMedia Signal Processing (MMSP).* IEEE, 2023, pp. 7–12.
8. C. Chen, Y. Xie, S. Lin, R. Qiao, J. Zhou, X. Tan, Y. Zhang, and L. Ma, "Novelty detection via contrastive learning with negative data augmentation," in *IJCAI International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence, 2021, pp. 606–614.
9. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*, 2020.
10. T. Ng, H. J. Kim, V. T. Lee, D. DeTone, T.-Y. Yang, T. Shen, E. Ilg, V. Balntas, K. Mikolajczyk, and C. Sweeney, "Ninjadesc: Content-concealing visual descriptors via adversarial learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 797–12 807.
11. J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11219 LNCS, pp. 682–697, 2018.
12. T. S. Chernov, D. P. Nikolaev, V. M. Kliatskine, T. S. Chernov, D. P. Nikolaev, and V. M. Kliatskine, "A method of periodic pattern localization on document images," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 9875. SPIE, 12 2015, p. 987508.
13. N. Ghanmi and A. M. Awal, "A new descriptor for pattern matching: application to identity document verification," in *Proceedings - 13th IAPR International Workshop on Document Analysis Systems, DAS 2018*. Institute of Electrical and Electronics Engineers Inc., 6 2018, pp. 375–380.
14. A. Castelblanco, J. Solano, C. Lopez, E. Rivera, L. Tengana, and M. Ochoa, "Machine learning techniques for identity document verification in uncontrolled environments: A case study," in *Mexican Conference on Pattern Recognition*, vol. 12088 LNCS. Springer, 2020, pp. 271–281.
15. M. Sirajudeen and R. Anitha, "Forgery document detection in information management system using cognitive techniques," *Journal of Intelligent & Fuzzy Systems*, vol. 39, pp. 8057–8068, 12 2020.
16. A. B. Centeno, O. R. Terrades, J. L. Canet, and C. C. Morales, "Recurrent comparator with attention models to detect counterfeit documents," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*. IEEE Computer Society, 9 2019, pp. 1332–1337.
17. S. Lugon Moulin, C. Weyermann, and S. Baechler, "An efficient method to detect series of fraudulent identity documents based on digitised forensic data," *Science & Justice*, vol. 62, no. 5, pp. 610–620, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1355030622001113
18. B. Talbot-Wright, S. Baechler, M. Morelato, O. Ribaux, and C. Roux, "Image processing of false identity documents for forensic intelligence," *Forensic Science International*, vol. 263, pp. 67–73, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0379073816301414
19. B. Martínez Tornés, E. Boros, A. Doucet, P. Gomez-Krämer, and J.-M. Ogier, "Detecting forged receipts with domain-specific ontology-based entities & relations," in *Document Analysis and Recognition - ICDAR 2023*, G. A. Fink, R. Jain, K. Kise, and R. Zanibbi, Eds. Cham: Springer Nature Switzerland, 2023, pp. 184–199.
20. B. Martínez Tornés, E. Boros, A. Doucet, P. Gomez-Krämer, and J.-M. Ogier, "Detecting forged receipts with domain-specific ontology-based entities & relations," in *Document Analysis and Recognition - ICDAR 2023: 17th International Conference, San José, CA, USA, August 21–26, 2023, Proceedings, Part III*. Berlin, Heidelberg: Springer-Verlag, 2023, pp. 184–199. [Online]. Available: https://doi.org/10.1007/978-3-031-41682-8_12
21. L. Bertojo, C. Néraud, and W. Puech, "A very fast copy-move forgery detection method for 4k ultra hd images," *Frontiers in Signal Processing*, vol. 2, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frsip.2022.906304
22. M. Al-Ghadi, P. Gomez-Krämer, and J.-C. Burie, "CheckScan: a reference hashing for identity document quality detection," in 14^{th} *International Conference on Machine Vision (ICMV 2021)*, W. Osten, D. Nikolaev, and J. Zhou, Eds., vol. 12084, International Society for Optics and Photonics. SPIE, 2022, p. 120840J. [Online]. Available: https://doi.org/10.1117/12.2623887

23. R. Bertrand, P. Gomez-Krämer, O. R. Terrades, P. Franco, and J.-M. Ogier, “A system based on intrinsic features for fraudulent document detection,” in 12^{th} *International Conference on Document Analysis and Recognition*, 2013, pp. 106–110.
24. P. Gomez-Krämer, K. Rouis, A. O. Diallo, and M. Coustaty, “Printed and scanned document authentication using robust layout descriptor matching,” *Multimedia Tools and Applications*, 2023. [Online]. Available: https://doi.org/10.1007/s11042-023-17021-1
25. Z. Li, H. Tang, Z. Peng, G.-J. Qi, and J. Tang, “Knowledge-guided semantic transfer network for few-shot image recognition,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.
26. H. Tang, C. Yuan, Z. Li, and J. Tang, “Learning attention-guided pyramidal features for few-shot fine-grained recognition,” *Pattern Recognition*, vol. 130, p. 108792, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320322002734
27. H. Tang, Z. Li, Z. Peng, and J. Tang, “Blockmix: Meta regularization and self-calibrated inference for metric-based meta-learning,” in *Proceedings of the* 28^{th} *ACM International Conference on Multimedia*, ser. MM ’20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 610–618. [Online]. Available: https://doi.org/10.1145/3394171.3413884
28. H. Tang, J. Liu, S. Yan, R. Yan, Z. Li, and J. Tang, “M3net: Multi-view encoding, matching, and fusion for few-shot fine-grained action recognition,” in *Proceedings of the* 31^{st} *ACM International Conference on Multimedia*, ser. MM ’23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 1719–1728. [Online]. Available: https://doi.org/10.1145/3581783.3612221
29. Z. Zha, H. Tang, Y. Sun, and J. Tang, “Boosting few-shot fine-grained recognition with background suppression and foreground alignment,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 3947–3961, Aug. 2023. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2023.3236636
30. S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, “Ganomaly: Semi-supervised anomaly detection via adversarial training,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11363 LNCS, pp. 622–637, 5 2018.
31. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 4 2004.
32. C. Godard, O. M. Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January. Institute of Electrical and Electronics Engineers Inc., 9 2016, pp. 6602–6611.
33. K. Bulatov, E. Emelianova, D. Tropin, N. Skoryukina, Y. Chernyshova, A. Sheshkus, S. Usilin, Z. Ming, J.-C. Burie, M. M. Luqman, and V. V. Arlazarov, “Midv-2020: A comprehensive benchmark dataset for identity document analysis,” *Computer Optics*, vol. 46, pp. 252–270, 7 2021.
34. J. Ouyang, G. Coatrieux, and H. Shu, “Robust hashing for image authentication using quaternion discrete fourier transform and log-polar transform,” *Digital Signal Processing*, vol. 41, pp. 98–109, 6 2015.