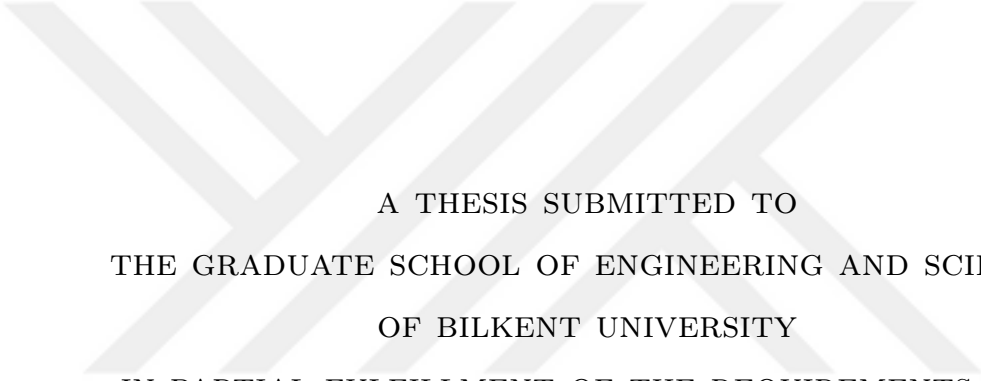


FACE MANIPULATION DETECTION

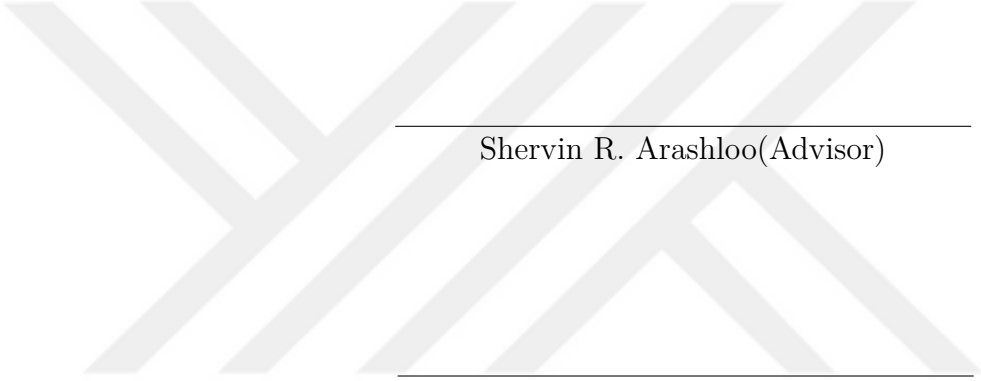


A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Sepehr Nourmohammadi
September 2023

Face manipulation detection
By Sepehr Nourmohammadi
September 2023

We certify that we have read this thesis and that in our opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.



Shervin R. Arashloo(Advisor)

Ayşegül Dünder

Çiğdem Gündüz Demir

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

FACE MANIPULATION DETECTION

Sepehr Nourmohammadi
M.S. in Computer Engineering
Advisor: Shervin R. Arashloo
September 2023

Advancements in deep learning have facilitated the creation of highly realistic counterfeit human faces, ushering in the era of deepfakes. The potential to generate such convincingly authentic fake content prompts concerns due to the potential harm it could inflict on individuals and societies alike. Current studies predominantly focus on binary approaches that differentiate between real and fake images or videos. However, this approach can be time-consuming, requiring a multitude of diverse fake examples for training. Furthermore, unique deepfake content generated using different models may elude detection, making it challenging to apprehend all deepfakes. We propose two potential solutions. First, we suggest a one-class classification method, a purist approach that trains solely on real data and tests on both real and fake data. Second, using a cross-manipulation technique as a non-purist approach, which refers to the application of image manipulations to a use unseen/unknown manipulated samples during the training of the machine learning model. Efficacy in this process can be achieved by using a combination of different models, which enhances the detection of deep fakes. This is done by merging learning-based systems involving an ℓ_p -norm constraint with adjustable p-norm rules, thereby providing both sparse and non-sparse solutions to enhance discriminatory information between based learners in ensemble learning. Contrary to conventional subject-independent learning methods employed in deep fake detection, we propose a subject-dependent learning approach. Our preliminary findings suggest that this multifaceted approach can effectively detect deepfakes, demonstrating impressive results on the FaceForensics++ dataset as well as on generic one-class classification datasets including the UCI, and Keel datasets in both pure and non-pure approaches.

Keywords: Deepfake detection, One-class classification, Ensemble learning, ℓ_p -norm regularization.

ÖZET

YÜZ MANİPÜLASYONU TESPİTİ

Sepehr Nourmohammadi
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Shervin R. Arashloo
Eylül 2023

Derin öğrenmedeki ilerlemeler, son derece gerçekçi sahte insan yüzlerinin yaratılmasını kolaylaştırdı ve deepfake çağını başlattı. Bu kadar inandırıcı derecede orijinal sahte içerik üretme potansiyeli, hem bireylere hem de toplumlara verebileceği potansiyel zarar nedeniyle endişelere yol açıyor. Mevcut çalışmalar ağırlıklı olarak gerçek ve sahte görseller veya videolar arasında ayrım yapan ikili yaklaşımlara odaklanmaktadır. Ancak bu yaklaşım zaman alıcı olabilir ve eğitim için çok sayıda farklı sahte örnek gerektirebilir. Ayrıca, farklı modeller kullanılarak oluşturulan benzersiz deepfake içeriği tespitten kaçabilir ve bu da tüm deepfake'lerin yakalanmasını zorlaştırabilir. İki potansiyel çözüm öneriyoruz. İlk olarak, yalnızca gerçek veriler üzerinde eğitim veren ve hem gerçek hem de sahte veriler üzerinde testler yapan saf bir yaklaşım olan tek sınıflı bir sınıflandırma yöntemi öneriyoruz. İkincisi, saf olmayan bir yaklaşım olarak çapraz manipülasyon tekniğinin kullanılması; bu, makine öğrenimi modelinin eğitimi sırasında görünmeyen/bilinmeyen manipüle edilmiş örneklerin kullanımına görüntü manipülasyonlarının uygulanması anlamına gelir. Bu süreçteki etkinlik, derin sahtekarlıkların tespitini artıran farklı modellerin bir kombinasyonu kullanılarak elde edilebilir. Bu, ℓ_p -norm kısıtlamasını içeren öğrenme tabanlı sistemleri ayarlanabilir p-norm kurallarıyla birleştirerek yapılır, böylece topluluk öğreniminde tabanlı öğrenenler arasındaki ayırmacı bilgileri geliştirmek için hem seyrek hem de seyrek olmayan çözümler sağlanır. Derin sahte tespitte kullanılan geleneksel konudan bağımsız öğrenme yöntemlerinin aksine, konuya bağlı bir öğrenme yaklaşımı öneriyoruz. İlk bulgularımız, bu çok yönlü yaklaşımın deepfake'leri etkili bir şekilde tespit edebildiğini, FaceForensics++ veri setinin yanı sıra UCI ve Keel veri setlerini de içeren genel tek sınıf sınıflandırma veri setlerinde hem saf hem de saf olmayan yaklaşımlarda etkileyici sonuçlar ortaya koyduğunu gösteriyor.

Anahtar sözcükler: Deepfake tespiti, Tek sınıf sınıflandırma, Topluluk öğrenimi,

ℓ_p -norm düzenlemesi.



Acknowledgement

Without the unwavering patience, deep field expertise, and professional guidance from my advisor, Professor Shervin R. Arashloo, who served as the ideal mentor, the completion of this thesis would have remained an unattainable goal.

The journey would have also been far more challenging without the extraordinary encouragement and understanding of my beloved parents. Their support has been a pivotal force in my life, making the realization of this study possible. My friends, too, deserve heartfelt thanks for their steadfast companionship and support, enriching my life in good times and bad.

Mathematical Notations

Symbol	Meaning
ℓ_p	Norm corresponding to p value
γ_{KPCA}	Controls the non-linear mapping
γ_{SVDD}	The width or scale of the kernel function in SVDD
K	Kernel matrix
σ	Presumed variance of the noise
I	Identity matrix
μ	Mean
ϕ	Projection function
p	Parameter controlling the vector norm for $1 \leq p < \infty$

Acronyms

Acronym	Meaning
AUC	Area under the ROC curve
ACC	Accuracy
EER	Equal Error Rate
CDA	Central Difference Attention
CDIN	Complementary Dynamic Interaction Network
CCDFM	Complementary Cross Dynamics Fusion Module
FFR	Frequency-based Feature Refinement
GANs	Generative Adversarial Networks
GLI	Global-Local Interaction
GDD	Generalization Deepfake Detector
GAD-branch	two parallel branches: Global Adaptive Dynamics-aware
GIMP	GNU Image Manipulation Program
HFI-Net	Hierarchical Frequency-assisted Interactive Networks
HWS	High-frequency Wavelet Sampler
MTCNN	Multi-Task Cascaded Convolutional Neural Networks
OCC	One-Class Classification
LRL	Local Relation Learning
LRM	Local Representative Lip Movements-aware
LSTM	Long-Short Term Memory
LTW	Learning-to-Weight
OCSVM	One-class Support Vector Machine
PCA	Principal Component Analysis
RMSE	Root Mean Square Error
SRM	Steganalysis rich model
SVM	Support Vector Machine
SVDD	Support Vector Data Description
VAE	Variational Autoencoder

Contents

1	Introduction	1
1.1	Motivation	1
1.1.1	Contributions	5
1.2	Organization	6
2	Related Work	7
2.1	Introduction	7
2.2	Categorization	7
2.2.1	Non-Pure approach	8
2.2.2	Pure approach	15
3	Proposed Approach	17
3.1	Introduction	17
3.2	Classifier fusion	17
3.2.1	Parallel classifier fusion	18

3.2.2	Sequential classifier fusion	22
3.3	Classifier Fusion for One-Class Learning	23
3.4	Objective Function	24
3.4.1	Multiple One-Class Classifier Fusion Subject to Sparsity Constraints	25
3.4.2	Frank-Wolfe method	27
3.5	Subject dependent/independent modeling	30
3.6	Normalization	30
4	Experimental Evaluation	33
4.1	Introduction	33
4.2	Datasets	33
4.2.1	genuine images	34
4.2.2	manipulated images	34
4.3	OCC models	36
4.3.1	Support Vector Data Description (SVDD)	36
4.3.2	Gaussian Process (GP)	37
4.3.3	Kernel PCA (KPCA)	38
4.3.4	Gaussian Mixture Model (GMM)	38
4.4	Metrics	41

4.5	Implementation Details	41
4.5.1	Model's details	42
4.5.2	one-class classifier fusion details	44
4.6	Results	45
4.6.1	Evaluation on the Faceforensics++ dataset using single models	45
4.6.2	uniform weights	47
4.6.3	Pure approach based on one-class classification	53
4.6.4	Non-pure approach of unseen/unknown method	58
4.6.5	Evaluation results on the datasets of Table 4.1 based for Frank-Wolfe method	64
4.7	Summary	66
4.7.1	Non-pure method (cross-manipulation) comparisons	69

5 Conclusion

List of Figures

4.1	from left to right, original, deepfake, faceswap, face2face, neural texture, and face shifter	35
4.2	from left to right, full face, nose and mouth and mouth regions . .	40
4.3	from left to right, full face, nose and mouth and mouth regions . .	40

List of Tables

4.1	datasets of UCI, and Keel	34
4.2	The hyper-parameters for SVDD	43
4.3	The hyper-parameters in the proposal approach	44
4.4	Model's AUC(%) in DeepFakes High quality frame based	45
4.5	Model's AUC(%) in FaceShifter High quality frame based	46
4.6	Model's AUC(%) in FaceSwap High quality frame based	46
4.7	Model's AUC(%) in Face2Face High quality frame based	46
4.8	Model's AUC(%) in Neuraltextures High quality frame based	46
4.9	AUC's(mean $\pm$ std%) for the datasets in Table 4.1 for a uniform weight fusion scheme	47
4.10	AUC(%) on the DeepFakes dataset using an uniform weight fusion scheme	48
4.11	AUC(%) on the FaceShifter dataset using an uniform weight fusion scheme	48

4.12 AUC(%) on the FaceSwap dataset using an uniform weight fusion scheme	49
4.13 AUC(%) on the Face2Face dataset using an uniform weight fusion scheme	49
4.14 AUC(%) on the Neuraltextures dataset using an uniform weight fusion scheme	49
4.15 ACC(%) on the DeepFakes dataset using an uniform weight fusion scheme	49
4.16 ACC(%) on the FaceShifter dataset using an uniform weight fusion scheme	50
4.17 ACC(%) on the FaceSwap dataset using an uniform weight fusion scheme	50
4.18 ACC(%) on the Face2Face dataset using an uniform weight fusion scheme	50
4.19 ACC(%) on the Neuraltextures dataset using an uniform weight fusion scheme	50
4.20 EER(%) on the DeepFakes dataset using an uniform weight fusion scheme	51
4.21 EER(%) on the FaceShifter dataset using an uniform weight fusion scheme	51
4.22 EER(%) on the FaceSwap dataset using an uniform weight fusion scheme	51
4.23 EER(%) on the Face2Face dataset using an uniform weight fusion scheme	51

4.24 EER(%) on the Neuraltextures dataset using an uniform weight fusion scheme	52
4.25 Average AUC(%) on the Faceforensics++ dataset through frame based results in the uniform weights	52
4.26 Average AUC(%) on the Faceforensics++ dataset through $Z - score$ results in the uniform weights	52
4.27 AUC's(mean $\pm$ std%) for the datasets in Table 4.1 for a pure approach learning based fusion scheme	53
4.28 AUC(%) on the DeepFakes dataset using pure approach learning based fusion scheme	53
4.29 AUC(%) on the FaceShifter dataset using pure approach learning based fusion scheme	54
4.30 AUC(%) on the FaceSwap dataset using pure approach learning based fusion scheme	54
4.31 AUC(%) on the Face2Face dataset using pure approach learning based fusion scheme	54
4.32 AUC(%) on the Neuraltextures dataset using pure approach learning based fusion scheme	54
4.33 ACC(%) on the DeepFakes dataset using pure approach learning based fusion scheme	55
4.34 ACC(%) on the FaceShifter dataset using pure approach learning based fusion scheme	55
4.35 ACC(%) on the FaceSwap dataset using pure approach learning based fusion scheme	55

4.36 ACC(%) on the Face2Face dataset using pure approach learning based fusion scheme	55
4.37 ACC(%) on the Neuralttextures dataset using pure approach learning based fusion scheme	56
4.38 EER(%) on the DeepFakes dataset using pure approach learning based fusion scheme	56
4.39 EER(%) on the FaceShifter dataset using pure approach learning based fusion scheme	56
4.40 EER(%) on the FaceSwap dataset using pure approach learning based fusion scheme	56
4.41 EER(%) on the Face2Face dataset using pure approach learning based fusion scheme	57
4.42 EER(%) on the Neuralttextures dataset using pure approach learning based based fusion scheme	57
4.43 Average AUC(%) of Faceforensics++ dataset through frame based results in Pure approach one-class classification	57
4.44 Average AUC(%) of Faceforensics++ dataset through $Z - score$ results in Pure approach one-class classification	58
4.45 AUC's(mean $\pm$ std%) for the datasets in Table 4.1 for a non-pure approach learning based fusion scheme	59
4.46 AUC(%) on the DeepFakes dataset using non-pure approach learning based fusion scheme	59
4.47 AUC(%) on the FaceShifter dataset using non-pure approach learning based fusion scheme	59

4.48 AUC(%) on the FaceSwap dataset using non-pure approach learning based fusion scheme	60
4.49 AUC(%) on the Face2Face dataset using non-pure approach learning based fusion scheme	60
4.50 AUC(%) on the Neuraltextures dataset using non-pure approach learning based fusion scheme	60
4.51 ACC(%) on the DeepFakes dataset using non-pure approach learning based fusion scheme	60
4.52 ACC(%) on the FaceShifter dataset using non-pure approach learning based fusion scheme	61
4.53 ACC(%) on the FaceSwap dataset using non-pure approach learning based fusion scheme	61
4.54 ACC(%) on the Face2Face dataset using non-pure approach learning based fusion scheme	61
4.55 ACC(%) on the Neuraltextures dataset using non-pure approach learning based fusion scheme	61
4.56 EER(%) on the DeepFakes dataset using non-pure approach learning based fusion scheme	62
4.57 EER(%) on the FaceShifter dataset using non-pure approach learning based fusion scheme	62
4.58 EER(%) on the FaceSwap dataset using non-pure approach learning based fusion scheme	62
4.59 EER(%) on the Face2Face dataset using non-pure approach learning based fusion scheme	62

4.60 EER(%) on the Neuraltextures dataset using non-pure approach learning based fusion scheme	63
4.61 Average AUC(%) of Faceforensics++ dataset through frame based results in Non-pure approach cross-manipulation	63
4.62 Average AUC(%) of video-base Faceforensics++ dataset through $Z - score$ results in Non-pure approach cross-manipulation	64
4.63 norm difference of weights	65
4.64 Average ranking of different approaches based on Freidman's test for the dataset of Table 4.1 ($p - value = 0.0005$) according to uniform weights, pure (CVX package), and non-pure (CVX package)	65
4.65 Elapsed time (seconds)	66
4.66 Comparison of the mean AUC(%) for datasets referenced in Table 4.1 using the pure learning-based approach(One-Class Classifica- tion) with Tanh score normalization in our methodology.	67
4.67 AUC (%) of Faceforensics++ dataset through video based (VB) results in Pure approach one-class classification	68
4.68 Comparing performance in One-Class Classification method on Faceforensics++ dataset in Low quality	69
4.69 AUC (%) of Faceforensics++ dataset through video based (VB) results in Non-pure method (cross-manipulation)	70
4.70 Comparing AUC (%) in cross-manipulation method on Faceforen- sics++ dataset in Low quality	70
4.71 Comparing AUC (%) in cross-manipulation method on Faceforen- sics++ dataset in High quality	71

Chapter 1

Introduction

1.1 Motivation

Manipulating visual media is both an art and a science, tracing its roots back to the inception of the medium itself. Over time, the evolution of sophisticated image manipulation and editing software, including but not limited to Adobe Photoshop, GIMP, and After Effects Pro, has enabled profound transformations of visual content. These software packages offer a wide array of manipulative operations, such as object addition, duplication, or deletion, providing users with the means to generate realistic and credible visual outcomes that can deceive even the most observant viewer. In recent times, however, the field of image manipulation has witnessed significant advances. The integration of state-of-the-art computer graphics and deep learning methodologies has taken this craft to a new level. These modern technologies facilitate manipulations with a superior degree of semantic consistency, further enhancing the authenticity of the manipulated visual content. The improvement in semantic consistency, a measure of the logical and contextual correlation in the image, indicates a promising trajectory for the future of image manipulation technologies. Innovations in deep learning methodologies have facilitated the development of remarkably realistic synthetic

human facial images, thereby initiating a novel epoch of image and video manipulation. The technology known as deepfakes, which combines deep learning and fake, utilizes sophisticated deep learning algorithms to effectively modify a person's facial appearance in an image or video, thereby transferring their distinctive characteristics and attributes. The occurrence of deepfakes has experienced a significant increase in recent years, indicating a notable transformation in the application of this technology. The term "deepfake" has expanded in its connotation, encompassing a broader range of digitally created content. This process commonly entails altering the subject depicted in a video or image, in order to replicate the characteristics or behaviors of another subject [1]. The remarkable ability of deepfakes to blur the lines between reality and artificiality marks a significant milestone in the trajectory of digital content manipulation technologies. Deepfakes typically make use of two key technologies: Autoencoders and Generative Adversarial Networks (GANs) [2]. These tools take an existing image (the source) and blend it onto a different image (the target), creating a new, forged picture. By combining autoencoders and GANs on a single-face image, false photos and videos can be produced. These deep learning techniques generate fake content or alter existing media to such an extent that it's very hard, sometimes even impossible, for people to tell if they are real or fake. Thanks to the large data sets now available and the advancements in deep learning, it's become easier to either change existing faces or create completely new ones without the need for a lot of manual editing. Several tools, such as Adobe Photoshop, StyleGAN [3], Faceswap, and PGGAN [4], are used to make these fake images. However, this growing ability to create fake videos and images, especially those that look very real, has raised concerns. The potential harmful impacts on individuals, businesses, and society are real. A well-known example is the deepfake of former U.S. President Barack Obama, highlighting the potential for misuse of this technology. Deepfakes were initially developed as part of a research project [5] that aimed to create a high-quality video of Barack Obama with accurate lip synchronization, superimposed onto a target video. This event exemplified the concern surrounding the potential for video manipulation, thereby presenting potential implications for privacy and security. For example, the potential for misuse arises when cyber criminals exploit deepfakes to propagate detrimental misinformation by means

of a fabricated video featuring a prominent individual. However, it is important to distinguish deepFakes from adversarial machine learning [6]. The objective of adversarial machine learning is to deceive machine learning algorithms through the deliberate manipulation of data with a fraudulent purpose. In contrast, the primary objective of deepFakes is to deceive human observers rather than computational systems.

Several notable techniques of facial modification include:

- **Comprehensive Facial Image Generation:** Through the application of sophisticated deep learning methods like Generative Adversarial Networks (GAN), including approaches such as StyleGAN[3], one can fabricate facial imagery that is entirely novel. These methods have proven highly effective, generating high-definition facial images that exhibit an impressive degree of realism.
- **Identity Substitution:** This form of manipulation involves the substitution of one individual's face with another's within a video sequence. Two common techniques are typically utilized to achieve this: firstly, conventional computer graphics techniques such as FaceSwap; and secondly, innovative deep learning methods known as DeepFakes, with the FaceApp smartphone application being a prime example.
- **Attribute Modification:** Also referred to as face editing, this form of facial image alteration involves making minor yet impactful changes to facial characteristics, such as modifying the person's gender, adding accessories like glasses, altering age appearance, skin color, or hair characteristics [7]. Generative Adversarial Networks (GANs), particularly methods like StarGAN, are often employed to execute such modifications [8]. Tools such as FaceApp have made executing these attribute modifications increasingly accessible to users.
- **Expression Transformation:** Also known as "facial reanimation," this method involves adjusting a subject's facial expression to correspond with the emotional state of another individual from a separate footage. While

there are other methodologies for such modifications [9], the techniques most commonly utilized include Face2Face [10] and Neural Textures [11]. Common forms of facial reanimations encompass mouth synchronization, often referred to as dubbing, as well as mirroring appearance and posture.

The prevailing trend in research efforts for deepfake detection involves building a binary classifier that differentiates between authentic and manipulated images or videos. However, this modeling paradigm presents several limitations as enumerated below.

- The training of a binary classifier necessitates the utilization of both authentic and manipulated data samples. While authentic images are relatively abundant, the generation of manipulated data can be a time-consuming process. It often requires the deployment of numerous deepfake models to ensure a diversified set of manipulated training examples.
- Crucially, the array of deepfake models employed during the training phase can substantially impact the versatility of the resulting binary classifiers. Consequently, novel and previously unobserved deepfake content, generated via a distinct deepfake model, may successfully bypass the detection system. This presents a significant challenge to maintaining the robustness and effectiveness of deepfake detection methods.

One approach to mitigate these challenges involves the application of one-class classification. In this scenario, only authentic data is employed for training, while both authentic and fabricated data are used for testing. However, the efficacy of existing one-class methods leaves room for improvement, particularly for manipulations involving expressions and face swapping, where accuracy and recall rates may be suboptimal. According to this perspective, more effective classifiers should be tailor-made for the task at hand, potentially augmenting the detection system’s ability to generalize to novel and unseen/unknown deepfake content.

Moreover, although the majority of current research relies on a single detection classifier, the expectation is that an ensemble of diverse classifiers could enhance the generalization capability of deepfake detection. The incorporation of techniques like stacking-based ensembles [12], multiple classifier combination strategies [13], or other fusion processes could be promising. Such performance-enhancing techniques represent a potent area of research to be further explored within this study, as they have not been extensively investigated in the literature concerning deepfake detection.

1.1.1 Contributions

The primary contributions of this research can be outlined as follows:

- Breaking away from the predominant subject-independent learning methodologies used in deepfake detection, our research introduces a subject-dependent learning approach. This approach leverages individual attributes of the subjects to enhance the performance of deepfake detection.
- We suggest using a strategy that combines multiple classifiers to improve detection performance. This strategy combines the information from several learning systems that work in situations where the type of manipulation is not known beforehand.
- In this research, we observe that some learners within an ensemble display superior discriminatory capabilities. Addressing this, we incorporate the task’s inherent sparsity into the learning phase, suggesting the weight vector could better represent the problem. Traditional optimization methods may not capture this. Hence, we introduce a sparsity-inducing mechanism, denoted as ℓ_p -norm, which determines sparsity from the data itself. This not only boosts model accuracy but also helps counteract overfitting or underfitting.

- In this research, we explored various score normalization techniques to better tailor our models to specific contexts. By adapting normalization methods to the unique characteristics of each context, we aim to improve the accuracy and robustness of our ensemble learners. The objective is to identify which normalization strategy offers the best trade-off between computational efficiency and predictive performance in different scenarios.

1.2 Organization

This research is divided into several chapters. In Chapter 2, we classify techniques into two primary categories: non-pure and pure techniques. This classification is based on the use of fake and real or only real data. Chapter 3 evaluates various ensemble learning methods, focusing on fixed-rule and learning-based approaches. This chapter also discusses our contributions, along with various score normalization methods. Chapter 4 examines different machine learning models, and datasets like UCI, Keel, and Faceforensics++. It also discusses the feature extraction process and implementation specifics. Additionally, this chapter presents results based on AUC (Area Under Curve)%, ACC (Accuracy)%, and EER (Equal Error Rate)%. In the summary subsection of Chapter 4, we evaluate the results of our approach and compare them to the existing methods. The final chapter, Chapter 5, provides a summary of our research.

Chapter 2

Related Work

2.1 Introduction

We will present a brief summary of the most notable and state-of-the-art strategies for detecting forged/manipulated face images in this chapter. These techniques may be divided into two categories: pure and non-pure approaches, discussed next.

2.2 Categorization

Numerous researchers have developed an interest in examining deepfakes, a process that encompasses generating and identifying manipulated videos that distort authentic content. As mentioned previously, deepfake detection methods can be roughly categorised into two main groups: pure and non-pure.

2.2.1 Non-Pure approach

The non-pure approach is a method of making predictions by using both positive and negative samples. There are two subcategories within this technique: seen/known and unseen/unknown detection. The non-pure detection problem can be cast as a binary classification problem where the objective is to assign the data to one of two possible classes. The task of binary classification can be challenging because the boundary between the two classes is often not well-defined, and the data can be noisy or imbalanced. To address these issues, various machine learning techniques, such as support vector machines, logistic regression, and neural networks, have been developed to improve the accuracy of binary classification models.

2.2.1.1 Seen/known classification

The seen/known classification and prediction involve using labeled data during the training phase of the model. In this approach, the model is trained on a labeled dataset, where each data point is associated with a known label or class. The model is then evaluated on the *same* type of data that it has been trained on. The seen/known approach is commonly employed in supervised learning tasks, such as classification and prediction, where the goal is to learn a mapping between the input data and the corresponding output labels. By utilizing labeled data, the model can learn to make accurate predictions and classifications, and the evaluation process can measure the model's performance in a meaningful way. The seen/known approach is particularly useful when sufficient labeled data is available, as it can lead to high accuracy and robustness of the trained model. However, it may not be suitable for tasks where labeled data is scarce or expensive to obtain or the type of test data may differ from that at training samples.

In binary classification, a standard machine learning task, the goal is to learn a decision boundary separating the two classes based on the patterns and relationships found in the training data. One notable example can be found [14] the

authors propose an innovative approach for identifying deepfake videos. They utilize a combination of facial features extracted from videos and Long-Short Term Memory (LSTM) deep networks to effectively detect deepfake content. The training process involves using a dataset containing both genuine and manipulated videos, from which the facial features are extracted. These features are then fed into the LSTM network, which is trained to classify videos as real or deepfake.

In the study by To et al. [15], the researchers located the facial area of the subject in a video using the MTCNN face detector and turned it into a facial image. Data augmentation was applied and the EfficientNet algorithm was used to detect images as genuine or fake. EfficientNet is a type of convolutional neural network design and scaling technique that maintains uniform scaling across all depth, width, and resolution dimensions by employing a compound coefficient. This means that EfficientNet models are able to achieve higher levels of accuracy while also improving efficiency when compared to other convolutional neural networks [16]. The prediction process was performed on the raw label and frame selection to achieve the finalized label. The overall procedure of this research includes four stages: first, cropping faces using landmarks obtained with MTCNN; second, augmenting data randomly; third, employing the EfficientNet algorithm (binary classifier) for detection; and finally, comparing the predicted label for each frame to determine the video’s final label.

In the study by Ma et al. [17], the contrastive learning method is employed to address the limitations present in previous approaches. The authors incorporate a contrastive loss function into the total loss function of a Siamese network, enabling the calculation of similarity distances between samples. This approach facilitates the comparison of a diverse set of images and ultimately yields general features that can be applied to other tasks.

Data distribution complexity and intra-class variance pose challenges for detection. In a different study [18], the authors propose a two-stage learning approach that adopts a divide-and-conquer strategy. Initially, multiple expert models are used to capture distinct manipulated patterns individually. Each model is trained

on a subset of forgery data based on manipulation-aware partitioning and optimized using binary cross-entropy loss. In the final stage, the models are merged into a unified model. Since the models are trained in different forgery domains, they possess varying discrimination abilities for each sample. Consequently, a weight configuration method is considered.

Luo et al. [19] presents a new approach for detecting facial manipulation in images using a combination of bilateral filtering and joint adversarial training. The proposed method enhances the robustness of facial manipulation detection systems by incorporating bilateral filtering to preprocess input images, effectively reducing noise and artifacts. Additionally, joint adversarial training is used to train the model, which involves a generator and a discriminator learning concurrently in a minimax game framework. The generator produces manipulated images, while the discriminator differentiates between genuine and manipulated images. This training process enables the model to better understand and detect various types of manipulations.

The work in [20] introduces a novel technique known as Local Relation Learning (LRL) for the purpose of detecting face forgery. The authors propose a deep learning-based method that identifies the unique patterns of manipulations in facial regions by learning local relations from a set of forgery features. The LRL model is trained using a combination of supervised and unsupervised learning techniques, which enables it to effectively detect a wide range of forgery types. By focusing on the fine-grained spatial information within facial regions, this approach achieves superior performance in detecting both known and unknown forgery techniques compared to existing methods. The training process involves a two-stage pipeline: first, the network is pre-trained on a large dataset of unlabeled face images using a self-supervised task, and then it is fine-tuned with labeled data to improve the forgery detection accuracy. This comprehensive approach to training allows the LRL model to robustly detect face forgeries in various scenarios.

2.2.1.2 Unseen/unknown classification

Identifying unseen/unknown manipulated fake faces is a challenging task, which is often evaluated using cross-manipulation testing scenario to gauge a model’s generalization capabilities. Cross-manipulation refers to the process of applying image manipulations or transformations to a dataset that was not used during the training of a machine learning model, in order to evaluate the model’s robustness and generalization capabilities [21].

In an attempt to address the challenges of identifying manipulated fake faces and enhancing the performance of the trained model, researchers have explored various techniques. In a recent study conducted by [22], a new model called Learning-to-Weight (LTW) is introduced. This model aims to detect counterfeit faces across various domains, including target domains that have not been previously encountered. The task of identifying counterfeit faces in unfamiliar domains presents a significant challenge due to the intricate and skewed data distributions resulting from diverse face manipulation techniques. The authors claim that various facial features make distinct contributions to a detection model that has been trained on multiple domains. This can result in the model being influenced by biases specific to each domain. In order to tackle this issue, the authors suggest the utilization of the LTW approach which involves the assignment of distinct weights for face images originating from various domains, achieved through the implementation of a meta-weight learning algorithm. This approach enhances the model’s capacity to generalize across various domains. Furthermore, the authors propose the incorporation of an intra-class compact loss in order to augment the detection capability of the network. During training, the authors select N source domains for training, where each subset is based on the attack method, and evaluate the trained model on an unseen/unknown target domain test set. The goal is to train the detection model on the source domain, and assess its performance on the unseen/unknown test domain without any model updating.

Miao et al. [21] emphasize that while current methods for identifying manipulated faces work well on evaluations within a specific dataset, they lack robustness

when it comes to generalization across multiple datasets. The authors propose a new approach called Hierarchical Frequency-assisted Interactive Networks (HFI-Net) to address this issue which aims to detect forgery cues related to frequency. The HFI-Net is comprised of a dual-branch network that integrates the advantages of both CNN and transformer models in order to effectively capture local details and global context, respectively. In order to improve the network’s capacity to detect manipulated faces, the authors propose the integration of a Frequency-based Feature Refinement (FFR) module. This module is designed to acquire frequency-based attention from RGB features, with the specific objective of emphasizing forgery cues while simultaneously suppressing pristine semantic information. This is achieved by selectively preserving middle-high frequency features and discarding low-frequency features. The authors additionally present a co-sharing Global-Local Interaction (GLI) module, which facilitates frequency-assisted interactions between the two branches of the network and captures their complementary nature. In addition, the GLI module is incorporated at every stage of the network in order to investigate frequency artifacts at multiple levels and enhance the network’s capability to identify manipulated faces. In a subsequent publication authored by Miao et al. [23], the researchers present a novel neural network architecture known as F2Trans that incorporates two modules, namely Central Difference Attention (CDA) and High-frequency Wavelet Sampler (HWS). The process of identifying manipulation patterns using CDA involves the analysis of both the intensity and gradient information associated with each pixel. This analysis allows for the generation of key and value pairs. The HWS method eliminates low-frequency components and prioritizes high-frequency cues in order to avoid model confusion and emphasize local frequency information. Additionally, it has the capability to function as a specialized pooling layer, generating hierarchical feature representations within the spatial-frequency domain. Extensive experiments demonstrate that the F2Trans framework is effective in detecting face forgery across datasets, manipulations, and perturbations.

This study in [24] introduces a technique for detecting fake faces that enhances the model’s generalization capabilities. The proposed method, termed Generalization Deepfake Detector (GDD), is designed to recognize manipulated videos

and images created by a specific method and is able to instantly detect new unseen/unknown domains without requiring updates to the model. The GDD algorithm employs a meta-learning-based approach to improve the model’s effectiveness and generalization. During the training phase, the model learns from N tasks, each with its own set of labels and extracted feature vectors. In the testing phase, the model is evaluated on one or more target domains that have not been seen/known before. The meta-learner process enables the model to learn quickly and exhibit high generalization capabilities by learning from related tasks. By optimizing the technique for source domains, the model can generalize the parameters to predict the target domain.

In a subsequent study, the authors [25] present a technique to improve the detection of fake media by using supervised contrastive learning. The proposed method is designed to handle unknown manipulations and datasets by introducing a cross-modality data augmentation approach that combines Steganalysis rich model (SRM) and RGB features to comprehensively extract detection clues. Additionally, a multi-scale feature enhancement module is suggested to improve textural and semantic information. The experiments conducted by the researchers demonstrate that the proposed approach enhances the generalization of the model in both intra- and cross-dataset scenarios.

In another research effort [26], the authors developed the Complementary Dynamic Interaction Network (CDIN) for forgery detection, focusing on analyzing dynamic facial feature movements. The network consists of two parallel branches: Global Adaptive Dynamics-aware (GAD-branch) and Local Representative Lip Movements-aware (LRM-branch). The GAD-branch examines the entire face, while the LRM-branch targets the mouth region. A Complementary Cross Dynamics Fusion Module (CCDFM) combines feature maps from both branches, enabling the global branch to detect unusual mouth movements and the local branch to gather global context. The network is optimized using a multi-task learning scheme with both global and local information.

In a different study [27], the authors have developed a new method for detecting face forgery. Their approach considers a face image to be the result of the

interaction between 3D geometry and the lighting environment. To disentangle the components of the image, they use a 3D decomposition-based method that separates it into four graphics components: 3D shape, lighting, common texture, and identity texture. These components are each constrained by a different model, including a 3D morphable model, a harmonic reflectance illumination model, and a PCA texture model. To improve the accuracy of the 3D shape prediction, they have also built a fine-grained morphing network that can predict shapes at the pixel level. Finally, they propose a composition search strategy that automatically constructs an architecture to detect forgery by mining forgery-relevant components.

Haliassos et al. [28] present a novel approach, called LipForensics, for detecting fake videos. This method is designed to be both generalizable to new manipulations and robust against various distortions. The initial stage of LipForensics involves the identification of prominent linguistic inconsistencies within lip movements, commonly observed in professionally produced videos. The method begins by pre-training a CNN that consists of spatiotemporal feature extractor to conduct lipreading, resulting in the acquisition of rich inner representations associated with genuine lip movements. Subsequently, freezing the feature extractor and optimizing a temporal network on manipulated data because unwanted artifacts that are not related to lip movements will affect the learning process.

The main objective of research in this field is to create a binary classifier that can distinguish between authentic and fake images or videos. However, this method faces some challenges. Firstly, it requires both genuine and fake data to train the classifier, and creating counterfeit data can be a lengthy process that demands the use of multiple deepfake models to ensure a varied training dataset. Secondly, the ability of binary classifiers to detect new and unknown deepfake content largely relies on the deepfake models employed during training. As a result, any innovative fake material generated using a distinct deepfake data might elude detection by the classifier.

2.2.2 Pure approach

The pure approach refers to the task of assigning a label or category to an input data point based on a set of predefined classes or categories. One of the related methods for this approach is one-class classification (OCC). One-class classification is a supervised learning problem, where the model is trained on labeled data to predict the class of new, unseen/unknown data points. On the other hand, one-class classification is also applicable to the anomaly detection problem where the goal is to identify data points that belong to a specific class, called the target class, while disregarding all other data points [29]. In this sense, one class classification is a special case of pure approach where there is only one class of interest.

While most studies in deepfake detection rely on two/multi-class classification, a few researchers consider the problem as a one-class anomaly detection challenge, where the learning process primarily depends on observations from a single class. This is because the concept of One-Class Classification (OCC) is centered on detecting patterns that adhere to a particular condition or behavior, which is referred to as "normal" or inliers, while identifying any other sample that deviates from this normal state, referred to as outliers or novelties, and deepfake detection falls under this category [30].

It is important to note that one-class classification approaches have been investigated widely, as they provide a distinct perspective for detecting deviations from normal states or behaviors. As one such instance, in the work in [31], the authors initially track facial and head movements from a single video input using and subsequently, extract the presence and intensity of specific action units, as described by Ekman and Friesen [32]. The researchers then construct a novelty detection model, utilizing a one-class support vector machine (SVM)[33], which is method for estimating the support of high-dimensional probability distributions. The authors provide empirical results that demonstrate the effectiveness of their method in various applications, including novelty detection, outlier detection, and noise reduction. The method offers a powerful tool for handling high-dimensional

data and identifying patterns in complex datasets.

Khalid et al. [34], introduce a technique called OC-FakeDect, which utilizes a one-class Variational Autoencoder (VAE) to exclusively train on authentic facial images. This approach aims to identify non-authentic images, such as deepfakes, by treating them as anomalies. In the initial methodology, the evaluation of the reconstruction score involves the computation of the root mean square error (RMSE) between the original input x and the reconstructed output x' in order to differentiate between genuine and manipulated images. In contrast, the second approach incorporates an additional encoder block subsequent to the decoder. The provided additional component receives the decoder output x' as an input and produces the value of $\mu(x')$. The root mean square error (RMSE) is calculated between the output $\mu(x)$ of the primary encoder and the output $\mu(x')$ of the secondary encoder for the given input image. The hypothesis posits that incorporating an extra encoder block can successfully extract authentic image features from the decoder output. However, this additional encoder block may not possess the capability to extract features from non-authentic images like deepfakes, resulting in a notable increase in the reconstruction score. The two methodologies are referred to as OC-FakeDect1 and OC-FakeDect2. The primary distinction among these suggested methodologies lies in the technique employed for calculating the reconstruction score. In particular, OC-FakeDect-1 algorithm quantifies the reconstruction score by directly comparing the input and output images. On the other hand, OC-FakeDect-2 algorithm evaluates the reconstruction score by comparing the latent representations of the input and output images, enabling a more accurate depiction of their latent features.

Chapter 3

Proposed Approach

3.1 Introduction

In this chapter, we present the methodology developed in this thesis. The methodology encompasses a variety of techniques for classifier fusion, normalization, and the establishment of the objective function.

3.2 Classifier fusion

The ensemble learning approach in terms of classifier fusion has been verified to be able to boost performance compared to single-classifier systems in a variety of different problems. An implicit assumption motivating a classifier fusion paradigm for pattern classification is the independency of the errors incurred by the classifiers enrolled into the ensemble. Generally, a classifier fusion approach may be realised as either a parallel or a sequential ensemble. In the parallel ensemble scheme, once multiple base learners are trained, each classifier outputs a decision (either hard or soft) regarding a test pattern and the ensemble mechanism merges the outputs of all classifiers to produce the final decision. In the

sequential ensemble scheme, on the other hand, the base learners in the ensemble operate in a serial fashion to produce their outputs under the condition that the output of each classifier in the pipeline is constrained by the preceding ones. Each of the aforementioned two fusion structures may operate using either a fixed rule or based on learning. While no learning task is associated with the high-level combination strategy in the fixed-rule category, in the learning-based approaches, the meta-system responsible for the fusion task is learned from the data. Next, an overview of the most common parallel classifier ensemble techniques is provided followed by a brief review of the serial classifier ensemble methods.

3.2.1 Parallel classifier fusion

Let us assume $\mathbf{z}$ is the pattern to be assigned to one of K possible classes with the corresponding measurement vector denoted as $\mathbf{x}$. For a single-classifier system, the decision rule for classification is to assign the pattern $\mathbf{z}$ to the class ω_j which maximises the a posteriori probability of $\mathbf{x}$:

$$\text{assign } \mathbf{z} \longrightarrow \omega_j \text{ if}$$

$$P(\omega_j|\mathbf{x}) = \max_k P(\omega_k|\mathbf{x}) \quad (3.1)$$

Now let us assume there exist R base classifiers each potentially operating on a distinct measurement of the pattern $\mathbf{z}$ denoted as $\mathbf{x}_i$ for the i^{th} classifier. In this case, different rules for the implementation of a parallel ensemble classifier exist, including the fixed-rule schemes, or learning-based methods, discussed next.

3.2.1.1 Fixed-rule parallel fusion

The most widely known mechanisms for a fixed-rule fusion of multiple classifier systems are as follows.

- *The product rule:* the fusion rule in this combination scheme is to merge different classifiers' a posteriori probabilities using multiplication. The product rule operates under a strict requirement of having all experts' outputs relatively high for class assignment of an observation since only one recognition engine can prevent a certain assignment by producing a probability of almost zero [13]. The product rule for classifier fusion is given as:

assign $\mathbf{z} \longrightarrow \omega_j$ if

$$P(\omega_j)^{1-R} \prod_{i=1}^R P(\omega_j|\mathbf{x}_i) = \max_k P(\omega_k)^{1-R} \prod_{i=1}^R P(\omega_k|\mathbf{x}_i) \quad (3.2)$$

where $P(\omega_j)$ is the prior probability of class ω_j and $P(\omega_j|\mathbf{x}_i)$ is the a posteriori probability of class ω_j for the measurement $\mathbf{x}_i$.

- *The sum rule:* in some situations, it may also be useful to make the extra assumption that the classifiers' a posterior probabilities won't differ much from their prior probabilities. This is a pretty strong assumption yet it may hold if the likelihood of the observed data is highly confusing because of excessive noise levels [13]. The sum rule for classifier fusion is:

assign $\mathbf{z} \longrightarrow \omega_j$ if

$$(1-R)P(\omega_j) + \sum_{i=1}^R P(\omega_j|\mathbf{x}_i) = \max_k [(1-R)P(\omega_k) + \sum_{i=1}^R P(\omega_k|\mathbf{x}_i)] \quad (3.3)$$

- *The max rule:* this classifier combination formalism is derived by approximating the sum operation in the sum rule by the largest class a posterior probability. As such, an object is assigned to a class for which the maximum class a posteriori probability is generated by one of the experts, *i.e.*

assign $\mathbf{z} \longrightarrow \omega_j$ if

$$(1-R)P(\omega_j) + R \max_i P(\omega_j|\mathbf{x}_i) = \max_k [(1-R)P(\omega_k) + R \max_i P(\omega_k|\mathbf{x}_i)] \quad (3.4)$$

- *The min rule:* considering the product rule and observing the fact that the product of multiple probabilities is bounded from above by their minimum, the min rule is defined as assigning the object to the class for which the minimum class a posteriori probability is maximised, *i.e.*

assign $\mathbf{z} \longrightarrow \omega_j$ if

$$P(\omega_j)^{1-R} \min_i P(\omega_j|\mathbf{x}_i) = \max_k P(\omega_k)^{1-R} \min_i P(\omega_k|\mathbf{x}_i) \quad (3.5)$$

- *The median rule:* the sum rule may be thought of as calculating the mean a posteriori probability for each class across all classifier outputs. Since the median is less affected by noise and outliers, the median rule may be preferred over the sum rule in some practical applications. The median rule is expressed as:

assign $\mathbf{z} \longrightarrow \omega_j$ if

$$(1-R)P(\omega_j) + \text{med}_i P(\omega_j|\mathbf{x}_i) = \max_k [(1-R)P(\omega_k) + \text{med}_i P(\omega_k|\mathbf{x}_i)] \quad (3.6)$$

- *The majority voting rule:* by hardening the output of every single classifier, the class assignment using the majority voting rule is performed by assigning the test object to the class that receives the majority of votes [13]:

assign $\mathbf{z} \longrightarrow \omega_j$ if

$$\sum_{i=1}^R \Delta_{ji} = \max_k \sum_{i=1}^R \Delta_{ki} \quad (3.7)$$

where Δ_{ki} is defined as

$$\Delta_{ki} = \begin{cases} 1 & \text{if } P(\omega_k|\mathbf{x}_i) < \max_j P(\omega_j|\mathbf{x}_i); \\ 0 & \text{otherwise.} \end{cases} \quad (3.8)$$

3.2.1.2 Learning-based parallel fusion

In addition to the fixed-rule parallel classifier fusion approaches discussed earlier, learning-based ensembles are also considered in different studies. As for the

fixed-rule fusion scheme, a learning-based parallel ensemble approach may be obtained by combining a set of base classifiers which are similar in type (homogeneous), or by combining different types of base classifiers (heterogeneous). A well-known class of learning-based parallel ensemble methods is the stacking approach which enables training an algorithm to combine the predictions of several comparable learning algorithms. In this formalism, once the base learners are trained, an ensemble algorithm is trained to create a final prediction utilising the predictions of the base learners as its inputs. As there is freedom in choosing the meta-learner, a variety of different stacking ensembles may be considered for a problem. The stacking approach has been utilised in different studies, for instance, in object detection [35] where the meta-learner is that of an SVM which is fed with both hard and soft labels produced by base learners that use local and global image characteristics. In another study [36], human activity recognition is addressed by employing a stacking ensemble approach built by a neural network as the meta-learner and multiple MLP (Multi-Layer Perceptron) models as base-learners. Other work [37] presented an ensemble classifier of stacked auto-encoders for identifying emotions to enhance the decision of a deep classification model where the meta-learner is constructed as a deep network. The work in [38] uses a stacking approach for medical image analysis where a stacked CNN model is used to integrate different types of CNN sub-models to analyse radiological images. Another example of the stacking approach corresponds to a weighted averaging of the prediction scores of multiple base learners. For instance, the study in [39] developed a weighted average fusion technique to combine the predictions of multiple face spoofing detectors. A two-stage optimisation strategy was then utilised to determine the parameters of the model. Another study [40] presents an ensemble model based on neural networks for speech emotion recognition that combines the predictions of separate neural networks that can handle both temporal and spatial inputs. A super-learner ensemble is proposed in [41] for the vehicle-type classification where a fully-connected layer with ReLU activation units between the merged outputs of the base learners (each being a CNN) and the output softmax layer is used as the meta-learner.

3.2.2 Sequential classifier fusion

Contrary to the parallel classifier fusion methods, the sequential systems deploy a serial process comprised of a series of base learners in which the predictions of each learner in the pipeline are affected by the outputs of the base learners preceding it in the system.

3.2.2.1 Fixed-rule sequential fusion

In this group of classifier fusion techniques, the learners are operated in a serial fashion but the fusion methodology is typically heuristically chosen and fixed for the problem at hand. For instance, one may consider match scores consecutively until a target degree of confidence is attained or all match scores are exhausted [42]. The work in [43] creates a unique sequential technique and a simple mathematical framework that can anticipate the effectiveness of two serially merged classifiers for the task of biometric recognition. This is accomplished by using a threshold-based categorisation where the identity of the subject is accepted if the first classifier's output score exceeds some upper threshold, and is rejected as an imposter if it is below a certain lower threshold. For any score lying between the two thresholds, a second classifier is invoked for decision-making. A limitation of this study is that a systematic approach for determining the upper and lower thresholds is required which reflects the appropriate levels of confidence. In general, however, the fixed-rule sequential fusion is highly heuristic when making decisions regarding different (hyper-)parameters of the fusion system.

3.2.2.2 Learning-based sequential fusion

Learning-based sequential fusion of classifiers uses training data to optimise some criteria for a serial fusion of multiple learners. The boosting approach [44] may be considered a well-known example of sequential ensembles that integrates multiple classifiers where previously incorrectly labelled data are re-weighted so that

subsequent learners in the ensemble focus more on such difficult examples to improve system performance. Cascade generalization is a specific type of ensemble learning that combines many classifiers using a progressively categorised design where at each level, the original data is extended by inserting additional characteristics obtained from the probability class distribution obtained from base classifiers [45]. One significant issue with this methodology is that it changes the feature space into a new high-dimensional feature space, which often results in a more challenging learning problem.

3.3 Classifier Fusion for One-Class Learning

A special case of classification problems corresponds to that of one class classification (OCC) in which the observed data for training is primarily obtained from a single, typically the positive class. The goal of OCC is to provide a mechanism to detect the genuine/target objects and distinguish them from any other sample that deviates from the supposed normal condition, known as anomalies/outliers. Motivated by the success of a classifier fusion scheme in multi-class classification problems, there have been different studies trying to develop a one-class classifier fusion system. As an example, in [46], an OCC fusion system based on weighted averaging is proposed to deal with the long-tailed non-anomalous data distributions following several types of score normalization of multiple OCC models, with applications to medical, anti-spam, and face spoofing data. The authors of [47] considered conventional classifier combination rules such as the sum rule, the product of the weighted votes, etc. for the cases where the outputs of one-class learners could be standardised. The merits of combining multiple SVDD classifiers are investigated in [48] for image dataset retrieval. In [49] combining OC-SVMs for biometric applications is suggested where different types of score normalisations including z-score, min-max, etc. are examined. In another study [50], genetic algorithms are utilised to optimise an ensemble of multiple OCC fuzzy k-nearest neighbour classifiers. The work in [51] suggests a new score normalisation strategy in the context of one-class classification ensemble to detect face spoofing attacks where a weighted average fusion approach was deployed. In

[52], the authors presented a unique stacking ensemble to combine multiple OCC learners where the ensemble is first pruned using genetic algorithms followed by a re-training procedure to create the final ensemble. The work in [53] presents a medical image classification technique based on a one-class kernel principle component analysis (KPCA) modelling ensemble where the ensemble comprises multiple one-class KPCA-trained models focusing on various image characteristics followed by a fusion algorithm for merging the KPCA model outputs. The one-class random forests (OCRf) approach is proposed in [54] which uses a randomization scheme to generate a diverse pool of tree-based one-class classifiers. As a fingerprint identification system is prone to spoofing attacks, the work in [55] proposed a one-class classification strategy where an ensemble of OC-SVMs fed with a variety of feature sets is used.

3.4 Objective Function

In the context of multiple-classifier fusion, the fixed-rule fusion strategies may be considered as the most widely used fusion techniques in the literature. Nevertheless, deployment of a fixed-rule scheme may suffer from certain limitations as choosing one from among multiple fusion strategies requires a high level of expert knowledge which may not be always accessible. Even in the presence of expert knowledge, a fixed-rule fusion scheme may not perfectly adapt to the characteristics of the data. Despite these limitations, among other fixed-rule fusion strategies, the sum rule has shown to be able to boost classification performance in a wide range of applications and has been widely put into practice. Nevertheless, in the sum rule, each model is considered to be as equally important as others in the ensemble. In practice, however, this simplistic assumption may not always hold. Ideally, in an ensemble approach, the combination scheme is to be learned from the data which may be also useful to reflect the prominence of one or more models in the ensemble. Motivated by these observations, in this work, we consider a weighted fusion of multiple one-class classifiers where fusion weights are learned from data. By virtue of assigning data-dependent weights to different learners, the proposed approach better captures data characteristics. Specifically,

by incorporating a variable vector norm prior to the ensemble, the sparsity of the weight vector is controlled, and consequently, the proposed fusion scheme can better favour one or more learners over others. Furthermore, while the majority of the existing OCC fusion methods lack a probabilistic output score, by incorporating suitable constraints into the proposed learning problem, a probabilistic fused score can be derived in the proposed method. And last but not the least, the learning problem in the proposed approach admits a convex formulation which facilitates effective optimisation.

3.4.1 Multiple One-Class Classifier Fusion Subject to Sparsity Constraints

As noted earlier, in this work, we consider a weighted fusion of multiple one-class classifiers to form the ensemble. Assuming there exist R classifiers to be combined, the final score for decision-making in the proposed combined system is formed as:

$$h = \sum_{i=1}^R \mathbf{s}_i \boldsymbol{\omega}_i = \mathbf{s}^\top \boldsymbol{\omega} \quad (3.9)$$

where the score corresponding to the i^{th} classifier is denoted as $\mathbf{s}_i$ and its weight as $\boldsymbol{\omega}_i$. $\boldsymbol{\omega}$ and $\mathbf{s}$ stand for the vector collections of $\boldsymbol{\omega}_i$'s and $\mathbf{s}_i$'s, respectively. The goal of the proposed approach is to determine $\boldsymbol{\omega}$ that yields a minimum empirical loss. Among other alternatives, the well-known and most used method least square loss may be used to estimate the empirical error. This is by no means the only option, but it is a plausible one. As such, the optimization problem for the proposed approach is:

$$\min_{\boldsymbol{\omega}} \|\mathbf{S}^\top \boldsymbol{\omega} - \mathbf{y}\|_2^2, \quad (3.10)$$

, where $\mathbf{y}$ represents the vector of true labels for all training observations, and $\mathbf{S}$, is the matrix of observation scores where each column vector corresponds to all scores generated for a single observation. The cost function in Eq. 3.10 when applied to one-class based learners for outlier detection, faces several challenges. Notably, it is sensitive to outliers because the squaring operation amplifies the impact of large errors [56]. When the score distribution is highly skewed, as with

imbalanced data, models may be unduly influenced by these outliers, skewing the model to fit these extreme cases rather than the overall trend of the "normal" data [57]. In such cases, the SVM empirical loss can provide better performance, due to its use of hinge loss which creates the largest possible gap between classes [58]. This is beneficial for tasks like outlier detection where the aim is to distinguish extreme cases (outliers) from normal cases. The hinge loss function is defined as

$$\min_{\boldsymbol{\omega}} \frac{1}{N} \sum_{i=1}^N \max(0, \mathbf{S}^T \boldsymbol{\omega} - \mathbf{y}) \quad (3.11)$$

Here, the sum is over all N data points in the dataset. In a pure one-class setting where only normal training objects are available, and considering a label of "1" for normal samples, $\mathbf{y}$ shall be a vector of 1's of a suitable size. Solving the optimisation problem above will then yield a weight vector $\boldsymbol{\omega}$ so that the combined scores in the ensemble are as close as possible to their true labels in a Euclidean sense. Note that the optimisation problem above is convex and can be optimised effectively. Although the weight's optimising the cost in Eq. 3.11 will minimise the distance between true and estimated combined scores in the ensemble, it may suffer from certain limitations. In practice, one or more learners in the ensemble may yield better discriminatory information as compared with others. In such cases, the weight vector would better match the underlying problem if the inherent sparsity/uniformity of the task is taken into account during the learning process. Nevertheless, the optimisation problem in Eq. 3.11 may not be able to directly address such situations. Ideally, a sparsity-inducing mechanism shall be introduced into the learning problem where the degree of sparsity is determined from the data. Such a prior may be also instrumental in reducing possible over-fitting or under-fitting of the model to the data. In practice, sparse regression has been considered in different problems to derive sparse coefficients that fit the data well. In different studies [59, 60] the ℓ_1 -norm prior (a.k.a. lasso) is examined to derive sparse coefficients as a primary sparsity-inducing tool based on regularisation. The ℓ_1 -norm sparsity, however, may not always be desirable as the information carried by some of the sources may be lost as a result of excessive sparsity. As a counter-example of the effectiveness of an ℓ_1 -norm penalty, the work in [61] demonstrated that the ℓ_∞ -norm SVM classifier surpasses the ℓ_1 -norm SVM in different applications. A viable solution in these

cases is a variable p -norm regularisation ($1 \leq p < \infty$) which can offer both sparse (for p 's close to 1) and non-sparse or even uniform solutions (for very large p 's). As such via a proper tuning of parameter p based on data characteristics, the p -norm regularisation has the potential to outperform both the ℓ_1 and ℓ_∞ norms in terms of prediction accuracy. Motivated by the aforementioned observations, and by introducing a variable p -norm regularisation the optimization problem in Eq. 3.11 is reformulated as:

$$\min_{\boldsymbol{\omega}} \frac{1}{N} \sum_{i=1}^N \max(0, \mathbf{S}^\top \boldsymbol{\omega} - \mathbf{y}) + c \|\boldsymbol{\omega}\|_p^p \quad (3.12)$$

where $\max(0, \mathbf{S}^\top \boldsymbol{\omega} - \mathbf{y})$ is the empirical loss, and $\boldsymbol{\omega}$ is the weight vector that must minimize the objective function. The vector p -norm term $\|\boldsymbol{\omega}\|_p^p$ for $1 \leq p < \infty$ controls the sparsity/uniformity of the solution and c is a positive coefficient. For p 's close to 1, the weight vector shall be relatively more sparse while for larger values of p the solution becomes more uniform. In a way, for p values close to 1, the variable p -norm constraint performs a classifier selection by assigning smaller weights to less effective OCC learners while larger weights are assigned to those with a higher discriminatory capability. The optimal p may be determined using validation data for a particular problem optimizing some performance criterion. As the optimization problem in the proposed approach has a convex objective function, the learning problem in Eq. 3.12 is an instance of convex optimization. In this work, we examine the CVX package [62] for the optimization of the convex-constrained problem in Eq. 3.12.

3.4.2 Frank-Wolfe method

Although the CVX package may be deployed for the optimization problem in Eq. 3.12, this method could prove to be relatively slow. Motivated by this observation [63], in this section, using the Frank-Wolfe method, we proposed a method for the optimization of the cost function in Eq. 3.12. The Frank-Wolfe process, an early and straightforward iterative method, has recently become favored again due to its ability to manage unique limitations, like maintaining a low-level or sparse data structure. This technique differs from others, like projected gradient descent and

proximal algorithms, as it doesn't require a projection back to a feasible region with every iteration. Instead, it resolves a linear problem within this space for each cycle, naturally keeping within the feasible area. The technique is often used for problems which require a balance of sparsity, treated as convex optimization within the boundaries of atomic sets, especially domains constrained by a type of mathematical regulation known as the ℓ_p -norm, which makes the Frank-Wolfe algorithm easily applicable. In order to minimize the objective function Eq. 3.12. the Frank-Wolfe algorithm iteratively updates two variables ω^t and s^t at the t_{th} iteration:

$$\omega^{t+1} = (1 - \gamma^t)\omega^t + \gamma^t s^t \quad (3.13)$$

$$s^t = \min_s s^\top \nabla f(\omega) \quad (3.14)$$

Here $\nabla f(\omega^t)$ is the gradient of empirical loss (Hinge loss) Eq.3.12 with respect to ω . Eq. 3.14 can be calculated based on Hölder's inequality as follows:

$$s_i^t = -c \text{sign}(\nabla_i f(\omega^t)) \frac{|\nabla_i f(\omega^t)|^{\frac{1}{p-1}}}{(\sum_{j=1}^M |\nabla_j f(\omega^t)|^{\frac{p}{p-1}})^{\frac{1}{p}}} \quad (3.15)$$

In this context, s_i^t and $\nabla_i f(\omega^t)$ represent the i_{th} entry of s^t and the gradient of $f(\omega^t)$ at position i , respectively. The factor c is a scaling constant. In many applications of the Frank-Wolfe algorithm, this constant is set to 1 for simplification. By setting $c = 1$, the direction s_i^t is normalized according to the ℓ_p -norm. This helps in ensuring that the step size is appropriately scaled, and the solution remains within the feasible set. Additionally, setting $c = 1$ can simplify computations and reduce potential sources of errors or instabilities in the iterative process. Step size γ in Eq. 3.13 can be computed as follows:

$$\gamma^t = \frac{2}{2 + t} \quad (3.16)$$

The convergence of ω^t is directly associated with the convergence of s according to Eq. 3.13, which is a feature intrinsic to the Frank-Wolfe (FW) algorithm. Equation 3.15 in the FW maintains these convergence properties, implying that ω^t will eventually reach a state of convergence. In this study, we established a specific convergence criterion to assess the stability and termination of our iterative processes. This criterion is defined by the maximum absolute difference

Algorithm 1: Training of Frank-Wolfe

```
1 Input: Training scores  $\mathbf{S}_b|_{n=1}^N$ ; labels of training scores  $\mathbf{y}_b|_{n=1}^N$ 
2 Output: new weights  $\omega_{new}$ 
3 Parameters:  $c, \gamma, \omega \in \{\omega_1, \dots, \omega_L\}, s \in \{s_1, \dots, s_L\}, p \in$   

    $\{32/31, 16/15, 8/7, 4/3, 2, 4, 8, 10, 100, 1000\}$ 
4 for  $b \leftarrow 1, \dots, 10$  do // initialization
5    $c \leftarrow 1$ 
6   // #select p hierarchically
7    $p_b \leftarrow p(1, b)$ 
8    $\epsilon \leftarrow 1e-6$ 
9    $\omega_{old} \leftarrow \text{zeros}(4, 1)$ 
10  for  $t \leftarrow 1, \dots, \text{Iteration}$  do // #epochs
11    // #compute gradient of objective function
12     $g \leftarrow \nabla_i f(\omega_{old}^t)$ 
13    // #compute the solution of the linear subproblem s
14     $s_i^t \leftarrow -c \text{sign}(g) \frac{|g|^{\frac{1}{p_b-1}}}{(\sum_{j=1}^M |g|^{\frac{p_b}{p_b-1}})^{\frac{1}{p_b}}}$ 
15    // #updating stepsize  $\gamma$ 
16     $\gamma^t \leftarrow \frac{2}{2+t}$ 
17    // #updating weights
18     $\omega_{new}^t \leftarrow (1 - \gamma^t) \omega_{old}^t + \gamma^t s^t$ 
19    // #check the convergence
20    if  $\max(|\omega_{new}^t - \omega_{old}^t|) \leq \epsilon$  then
21       $\text{break}$ 
22    // #return back
23     $\omega_{old}^t \leftarrow \omega_{new}^t$ 
```

between successive iterations of the weight vector, represented by ω . Specifically, the algorithm is considered to have converged when the difference between the new weight vector ω_{new}^t and the old weight vector ω_{old}^t at iteration t is less than or equal to 10^{-6} . This threshold ensures a high precision in the convergence, indicating that further iterations would result in negligible changes to the weight vector, thereby providing a reliable stopping point for the algorithm. The comprehensive procedure for the training and optimization of the Frank-Wolfe is presented in Algorithm 1.

3.5 Subject dependent/independent modeling

The existing face forgery detection algorithms in the literature attempt to capture the distribution of real and manipulated samples independent of the identity of the test subject. As the training data for such systems are from multiple participants, the underlying distribution is likely to be highly multi-modal. Consequently, an individual classifier utilized on data with such complicated distribution may be less effective than expected. In this context, a subject-specific modelling approach may be a potential alternative to the conventional subject-independent formalism. In a subject-specific learning scheme, instead of attempting to feed the classifiers with the data from all subjects with a highly complicated distribution, a distinct classifier is trained for each person separately. In this case, as the training data from a single subject is expected to be less complicated, the so-trained classifier is more likely to better capture and represent the underlying distribution, leading to an improved classification performance. A subject-specific modelling approach in face image analysis is not new; it has already been studied for face presentation attack detection problem [64], resulting in significant performance improvements. Nonetheless, for the face forgery detection problem, no earlier study has considered a subject-specific modelling approach. Fueled by the success of a subject-specific modelling method in other closely related problems, we advocate this technique to the face manipulation detection problem.

3.6 Normalization

Before deploying the extracted scores into the classifier fusion an important consideration shall be a suitable normalization of classifier output scores to enable an effective fusion system. Score normalization is an important pre-processing step in multiple expert fusion. However, there exist very few studies focusing on the relative effectiveness of different score normalization methods in the context of anomaly detection. There are some contemporary methods of normalization, used frequently like z-score, and min-max normalizations, discussed next.

3.6.0.1 Min-max normalization

In the traditional min-max normalization, every value is converted into a decimal between 0 and 1, with the smallest value of that characteristic becoming a 0 and the greatest value becoming a 1.

$$\tilde{s} = \frac{(S_i - \min(S_i))}{(\max(S_i) - \min(S_i))} \quad (3.17)$$

In the study conducted by Fatemifar et al. [65], a pioneering methodology for score normalization was introduced. The authors presented an innovative two-sided normalization approach, as described in their research [65]. This approach can be expressed mathematically as a quantized form of the min-max equation.

$$\tilde{s} = \frac{(S_i - S_l)}{(S_u - S_l)} \quad (3.18)$$

The variable S_l denotes a normalization point, which serves as a threshold to exclude the lower β percentage of data from a given score distribution. Similarly, the upper tail normalization point, denoted as S_u , is employed to remove the upper β percentage of data.

The variability of the parameter β exhibits a range spanning from 1% to 10%. In order to optimize this parameter, an assessment is conducted on a representative subset comprising 2% (20 samples in the case of the faceforensics++ dataset) of the entire dataset, and subsequently, the most frequently occurring value is identified as the reference point. For the purposes of this study, the value of β has been determined as 5% following the outcome of the aforementioned tuning process.

3.6.0.2 Z-score

Z-score normalization corresponds to the method that converts raw, unstructured data into normalized values or ranges of data utilizing concepts like mean and standard deviation. It can be calculated as:

$$\tilde{s} = \frac{(S_i - \mu)}{\sigma} \quad (3.19)$$

3.6.0.3 Modified Z-score

Utilizing the z-score might cause trouble because of two reasons:

- Potential outliers have a significant impact on the mean and standard deviation.
- When applying the z-score, it is presumed that the data is distributed according to a Gaussian normal distribution. However, rather than being distributed normally, certain data can be approximated by a normal distribution.

To overcome these issues, a modified z-score is defined which is more robust to outliers in the data [18]:

$$\tilde{s} = \frac{Z(S_i - \text{median}(S_i))}{\text{median}\{|S_i - \text{median}(S_i)|\}} \quad (3.20)$$

The median absolute deviation is used in the denominator which is a variation of the average absolute deviation. Variation of the average excludes the calculation on the tails and this property makes the normalization to be less affected by the outliers. As Z is a consistency correction constant, Median Absolute Deviation (MAD) is a reliable estimate of the standard deviation in [18]. value of the Z is determined based on the underlying data distribution, and it is suggested to as 0.6745.

3.6.0.4 Tanh

Hampel et al. [66] proposed a new technique for score normalization called tanh-estimator. the normalization is given below

$$\tilde{s} = 0.5\{\tanh(0.001(\frac{(S_i - \mu)}{\sigma})) + 1\}, \quad (3.21)$$

, in here μ and σ are the mean and standard deviation respective to the genuine score. This robust and efficient method, works very well for noisy training scores.

Chapter 4

Experimental Evaluation

4.1 Introduction

In this chapter, first, we provide a short overview of the datasets used in this study and the initial steps taken to prepare the data, such as extracting different local features and overall feature extraction process. We will also review the base OCC learners that we will deploy in our ensemble approach. Finally, we will discuss the results achieved using these different approaches.

4.2 Datasets

A series of experiments were conducted utilizing a set of fifteen diverse datasets obtained from various domains including healthcare, spam detection, credit card, signal processing, and face manipulated detection to evaluate the effectiveness at the proposal approach. A short description of the datasets employed in this study is provided below.

In this work, we use the FaceForensics++ [67] dataset for face manipulation detection, which includes normal data as well as five different types of deep fake

Table 4.1: datasets of UCI, and Keel

	Dataset	Samples	Normal	Features	Source
d1	<i>Spam</i>	4601	2788	57	UCI
d2	<i>Banknotes</i>	1372	762	5	UCI
d3	<i>Banana</i>	5300	294	3	Keel
d4	<i>Parkinson</i>	1040	520	29	UCI
d5	<i>AlgerianForestFires</i>	244	106	11	UCI
d6	<i>Ionosphere</i>	351	225	35	UCI
d7	<i>Pima</i>	768	500	8	UCI
d8	<i>Vote</i>	435	267	17	UCI
d9	<i>Glass</i>	214	76	9	UCI
d10	<i>Iris</i>	150	50	4	UCI
d11	<i>BreastCancerWisconsin</i>	699	458	10	UCI
d12	<i>Wine</i>	178	71	13	UCI

data in three different quality levels (raw, low, and high). 1,000 YouTube videos have been gathered and show frontal faces without occlusions, allowing automated tampering techniques to create realistic forgeries. FaceForensics++’s creators use these movies to create five various sorts of deepfake images, including FaceSwap, Face2Face, Deepfakes, NeuralTextures, and face shifter.

4.2.1 genuine images

FaceForensics++ provides a ground truth video dataset and uses several facial alteration methods to produce various deepfakes. Using Multitask cascaded Convolutional Neural Networks (MTCNN), we were able to recover actual human face images from these original recordings.

4.2.2 manipulated images

Manipulated facial images may be broadly categorised into two major groups: 1) Reenactment, and 2) Replacement. In face reenactment, the expression and

movements of different part of the face may be altered while preserving the

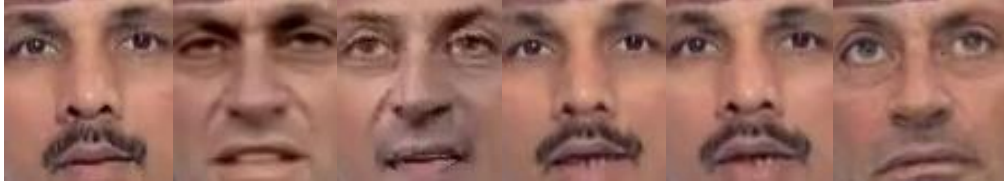


Figure 4.1: from left to right, original, deepfake, faceswap, face2face, neural texture, and face shifter

identity of the subject. In face Replacement, the face of a subject is replaced by the face of a different person, hence, the identity is changed. The FaceForensics++ incorporates both types of manipulations. Two models deployed to perform face reenactment in this dataset are as follows.

- Face2Face: provides real-time face recreation of a target video sequence which preserves the identity of the target person by animating the source person’s facial emotions in a target video and re-rendering the edited output video in a photo-realistic manner [10]. It is based on the manual keyframe selection of two input video streams, which are subsequently utilized for the dense facial reconstruction and the re-synthesis of the face with various manipulations and emotions.
- NeuralTextures: was first trained with an adversarial loss and a photometric reconstruction loss, but in [67], a patch-based GAN-loss implementation is used instead. The tracking module of Face2Face modifies the mouth area to provide neural texture data.

For face replacement, three different approaches are used in the FaceForensics++ dataset, discussed next.

- FaceSwap: is graphics-based method for moving a person’s face from one picture or video to another [68]. For this purpose based on sparse landmark points, the face region is retrieved and then fitted to a 3D template model using blend shapes. The distance between the projected form and the local landmarks is reduced by utilizing the textures of the input picture when this model is subsequently projected back to the target image [67].

- **DeepFakes:** is considered replacing human faces with artificial ones using deep learning methods. The purpose of DF is to swap out a person’s face for a target person’s face in a film. Two Autoencoders, sharing a single encoder trained to reassemble the original and the target person’s picture, are used to create fake video sequences [67].
- **FaceShifter:** FaceShifter is a brand-new, two-stage face swapping algorithm for high-fidelity, occlusion-aware face switching. FaceShifter creates the swapped face with great fidelity by utilizing and combining the target qualities comprehensively and adaptively. In contrast to many other face swapping works that simply use a limited amount of data from the target picture [69].

4.3 OCC models

In order to construct the pool of OCC learners to be combined using the proposed approach, we make use of multiple one-class classifiers applied to multiple sets of features. The one-class classifiers used in the current work are briefly discussed next followed by a description of the feature extraction mechanism followed.

4.3.1 Support Vector Data Description (SVDD)

The goal in the SVDD method [70] is to determine a boundary as a sphere, i.e. the simplest form of boundary, around the data. The outliers are then expected to lie outside the boundary while normal objects are considered to fall inside the spherical boundary. For decision making, given a test sample, $\mathbf{z}$, its squared Euclidean distance from the centre of the sphere, i.e. $\mathbf{a}$, is computed as

$$D^2(\mathbf{z}) = \|\mathbf{z} - \mathbf{a}\|^2 = \langle \mathbf{z}, \mathbf{z} \rangle - 2 \sum_i \alpha_i \langle \mathbf{z}, \mathbf{x}_i \rangle + \sum_{i,j} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad (4.1)$$

where $\langle ., . \rangle$ stands for inner product and $\mathbf{x}_i$'s stand for training data samples while α_i 's are coefficients (Lagrange multipliers) of the model. The resulting distance is compared against the radius of the sphere computed as

$$R^2 = \langle \mathbf{x}_k, \mathbf{x}_k \rangle - 2 \sum_i \alpha_i \langle \mathbf{x}_k, \mathbf{x}_i \rangle + \sum_{i,j} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad (4.2)$$

where $\mathbf{x}$ is a support vector lying on the boundary. Observation $\mathbf{z}$ is then categorized as normal data if $D^2 \leq R^2$, otherwise classified as an outlier. In order to derive flexible descriptions, a non-linear kernel approach may be utilised in the SVDD approach.

4.3.2 Gaussian Process (GP)

The GP method is known as a flexible model of Bayesian estimation that has been widely applied to various learning tasks, including for instance, image reconstruction, recognizing facial expressions, human pose estimation, etc. The GP approach is a generalisation of the multivariate Gaussian distributions to infinite dimensionality where the objective is to reduce the dimension of the high-dimensional data and learn low-dimensional representation [71]. The main assumption in the GP method is that the unknown function mapping the data from the input space to the output space (plus some noise) follows a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{K}$:

$$F \sim \mathcal{GP}(\boldsymbol{\mu}, \mathbf{K}) \quad (4.3)$$

Using a zero-mean GP, the predictive distribution is Gaussian whose moments can be estimated in closed form. For OCC, since the mean $\mu_\star$ decreases for inputs distant from the training data, it may be utilised as a measure for classification:

$$\mu_\star = \mathbf{k}_\star^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{1} \quad (4.4)$$

where $\mathbf{k}_\star$ is the test kernel vector, σ^2 denotes the presumed variance of the noise, and $\mathbf{1}$ stands for a vector of ones of a suitable dimension while $\mathbf{K}$ is the kernel matrix.

4.3.3 Kernel PCA (KPCA)

One of the widely used algorithms for data analysis is the Principal Components Analysis (PCA). PCA is a linear approach which allows the user to reduce the dimension of the data by minimising the information loss. The PCA algorithm shrinks the dimension of the data by discovering an orthogonal linear subspace a.k.a. principal components. When dealing with nonlinear datasets, however, the PCA method may not yield an effective dimensionality reduction. In such cases, the kernel PCA method [72] is proposed to project the dataset to a feature space (high dimensional) where the dataset could be better represented linearly. Assume that we map a d -dimensional point $\mathbf{x}_i$ to a higher N -dimensional space as $\Phi(\mathbf{x}_i)$ where $\Phi(\cdot)$ is a projection function $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^N$. As computations in a higher dimensional space for the KPCA algorithm is quite complex, if not impossible, and since all computations may be described in terms of dot products in the feature space, the kernel trick may be used to perform computations in the new space:

$$\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle = k(\mathbf{x}, \mathbf{y}) \quad (4.5)$$

where $k(\cdot, \cdot)$ is the kernel function. This approach makes it possible to solve for the eigenvectors and eigenvalues of the covariance matrix that couldn't be directly computed in the feature space. The KPCA approach has been utilised in [73] for novelty detection where for the novelty measure, the reconstruction residual of a test sample in the feature space is used for classification.

4.3.4 Gaussian Mixture Model (GMM)

A Gaussian Mixture Model (GMM) refers to a probability density function that includes a weighted sum of M Gaussian densities [74]:

$$p(\mathbf{x}|\theta) = \sum_{i=1}^M \omega_i g(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) \quad (4.6)$$

where $g(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ is

$$g(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\sigma}_i|^{\frac{1}{2}}} \exp \left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)^\top \boldsymbol{\sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right), \quad (4.7)$$

,in which $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ are the mean vector and covariance matrix of the i^{th} component, respectively. In biometrics, GMMs are frequently employed as a mathematical model of the posterior distribution of features in the system. The Expectation-Maximization (EM) algorithm may be employed to determine the parameters of the mixture. For OCC, once a GMM model is fit to the normal samples, the Mahalanobis distances between the test sample and the Gaussian components are calculated. For decision-making, one may use the minimum of such distances.

4.3.4.1 Feature extraction

Motivated by the recent work focusing on the frequency contents of the images to detect forged images/videos, in this work, we extract frequency-domain features for forgery detection. In the frequency domain, an image is represented in terms of its frequency components, which can be obtained using the 2D Fourier transform. The Fourier transform of an image $f(x, y)$ is given as.

$$F(u, v) = \sum_x \sum_y f(x, y) \exp(-2\pi i(ux + vy)), \quad (4.8)$$

,where $F(u, v)$ is the frequency component at position (u, v) , and $f(x, y)$ is the pixel value at position (x, y) in the spatial domain. For feature extraction, we use mid-to-high frequency components in the Fourier spectrum of an image as they have been observed to provide better discriminatory information . In order

to extract mid-to-high frequency components of an image, we use a Gaussian high-pass filter where the standard deviation of the Gaussian filter is set to 20% of the minimum number of rows and columns of the Fourier spectrum.

Once low-frequency components of the Fourier spectrum are removed, regarding the filtered output as an image, we convert the 2D array into a feature vector

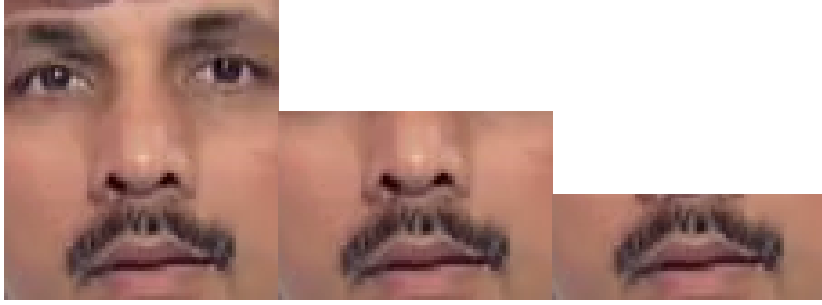


Figure 4.2: from left to right, full face, nose and mouth and mouth regions

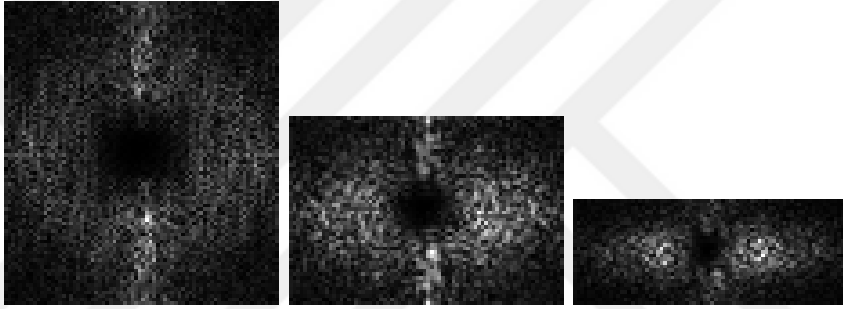


Figure 4.3: from left to right, full face, nose and mouth and mouth regions

by passing it through a pre-trained CNN model and regarding the output from the penultimate layer of the network as the final feature vector. Since local image regions are expected to carry useful discriminatory information for forgery detection, in addition to the whole face image, we apply the procedure described above to two additional local regions the mouth as well as the region surrounding the mouth and the nose region. the regions and related filtered Fourier spectrum are depicted Fig. 4.24.3. The regions used for feature extraction are depicted in Fig. 4.2. In practice, instead of a single CNN model, we use multiple CNN's to extract multiple feature sets. The CNN's used in the current work to extract features from the filtered Fourier spectrum are the Alexnet [75], Inceptionv3 [76], and Darknet19 [77], giving rise to 9 (3 regions $\times$ 3 CNN's) sets of features. In total, we build 36 different one-class learners (4 (classifiers) $\times$ 9 (features)) whose outputs shall be combined using the proposed approach. To implement these Convolutional Neural Networks (CNNs) for the analysis of filtered spectra, it is initially necessary to adjust the dimensions of the filtered Fourier spectra to align with the requisite input size of each specific CNN (e.g., 256x256). This step ensures that the data structure is consistent and compatible with the processing

parameters of each network. This methodology, while effective, requires careful consideration of image processing techniques. In our approach, images are simply re-scaled before the 2D-Fourier transformation. Re-scaling is a standard procedure in image processing that is frequently used to adjust the resolution or harmonize the size of images across datasets. Notably, the re-scaling of images has no effect on the decision-making process because our method relies on frequency-based features. This highlights the durability of our strategy, which ensures consistent performance regardless of changes in image scale.

4.4 Metrics

In this research, we have chosen to employ three key performance metrics to evaluate the effectiveness of our model: Area Under the Receiver Operating Characteristic Curve (AUC), Accuracy (ACC), and Equal Error Rate (EER). AUC provides a comprehensive evaluation of the model’s performance across different classification thresholds, while ACC represents the overall proportion of correctly classified instances out of the total number of instances, offering a straightforward and intuitive assessment of the model’s success. Lastly, EER measures the point where false acceptance and false rejection rates are equal, making it a useful metric for balanced performance. By utilizing these three complementary metrics, we aim to provide a robust and comprehensive analysis of our model’s performance.

4.5 Implementation Details

This section focuses on the implementation details, including the train/validation/test split ratios and parametric specifics. The generic machine learning datasets presented in Table 4.1 have been partitioned into distinct subsets, namely, train, validation, and test, using specific ratios. The allocation of samples within these subsets adheres to predetermined guidelines known as the fixed rule and pure method. Specifically, within the normal sample category, 70% of the samples are

assigned to the train set, while the remaining 20% and 10% are designated for the validation and test sets, respectively. As for the anomaly samples, the first

50

The face manipulation detection dataset used in this study is Faceforensics++, and different splitting ratios were employed based on the chosen methods for a fair comparison with the existing methods in the literature. For the fixed-rule and pure learning approaches, 72% of the genuine data was allocated for training. The validation set consisted of 14% of both genuine and fake samples, selected from the middle range of the dataset while the remaining 14% of genuine and fake samples were used as the test set. In the case of the non-pure learning method, which incorporates unseen/unknown fake samples, all of the training fake samples were included in the training set.

The parametric details can be discussed in two stages. Firstly, the parametric details specific to the employed models, namely Kernel Principal Component Analysis (KPCA), Support Vector Data Description (SVDD), Gaussian Process Regression (GPR), and Gaussian Mixture Model (GMM), are addressed. These details encompass the specific parameters, hyperparameter tuning, and model-specific considerations for each of the mentioned models. Secondly, the implementation details pertaining to the fusion of one-class classifiers subject to sparsity constraints are examined. These implementation details encompass the specific techniques, algorithms, and methodologies employed to achieve the fusion of classifiers while incorporating sparsity constraints. The discussion includes techniques for weight determination, optimization approaches, and any additional constraints or considerations necessary for achieving sparsity in the fusion process.

4.5.1 Model's details

In the upcoming experiments, the Kernel Principal Component Analysis (KernelPCA) involves two crucial parameters: γ_{KPCA} and n . The γ_{KPCA} parameter controls the non-linear mapping of the input data into a higher-dimensional

feature space, affecting the behavior and performance of Kernel PCA in other words, it is the width of the Gaussian kernel function. The n parameter determines the number of principal components to be retained or extracted from the kernel PCA analysis, defining the dimensionality of the reduced feature space. In our experiments, we consider the number of components from the set $n \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We set the γ_{KPCA} parameter to 0.25 based on experimental findings, as it does not significantly impact the performance.

The second model under consideration is the Support Vector Data Description (SVDD) algorithm, which involves two key parameters: γ_{SVDD} and C . The γ_{SVDD} parameter determines the width or scale of the kernel function in SVDD, influencing the similarity or distance computation between data points in the feature space. Higher γ_{SVDD} values result in a more localized kernel function, concentrating the influence of a training sample within a smaller region. Conversely, lower γ_{SVDD} values lead to a more spread-out kernel function. The second parameter, C , often represented as the *Cost* parameter, controls the balance between the size of the sphere (or hypersphere) used to encapsulate the target class and the allowance for training data to be considered as outliers. Selected hyperparameters were fine-tuned and validated using a dedicated validation set to ensure optimal performance and generalization on unseen data. The parameter ranges for γ_{SVDD} and C are defined according to the table provided below.

Table 4.2: The hyper-parameters for SVDD

<i>SVDD – Prameters</i>	1	2	3	4
γ_{SVDD}	0.01	0.1	1	10
C	0.1	1	10	100

In the subsequent model, Gaussian Process Regression (GPR), the selection of the kernel function and its associated hyperparameters plays a crucial role in defining the shape and characteristics of the Gaussian process model. There exist various kernel functions that are commonly employed, including the squared exponential (RBF) kernel, Matérn kernel, periodic kernel, and others. Each kernel function comes with its own set of hyperparameters that govern specific aspects of the model, such as its smoothness, length scales, and periodicity. For the purpose of this research, the (RBF) kernel function was utilized, thereby implying that

the smoothness and covariance between data points were modeled based on radial basis functions.

The Gaussian Mixture Model (GMM) is a probabilistic model that characterizes a dataset as a combination of K Gaussian distributions, where K represents the number of components. Each component within the GMM corresponds to a distinct cluster or mode of the data. In this study, the number of components for the GMM has been set as 3, implying that the dataset is represented by a mixture of three Gaussian distributions.

4.5.2 one-class classifier fusion details

There are two pivotal parameters in the fusion of One-class Classifiers, namely, c and the parameter of the ℓ_p -norm denoted as p (Equation 3.12). The parameter p represents the vector norm for $1 \leq p < \infty$, influencing the sparsity or uniformity of the solution. On the other hand, c is a positive constant coefficient. When p approaches unity, the weight vector tends to exhibit greater sparsity, whereas larger values of p lead to a more uniform solution. In essence, for p values close to unity, the p -norm constraint aids in the selection of classifiers by assigning lower weights to less effective one-class classifiers, while assigning higher weights to those with superior discriminatory capabilities. The optimal value of p can be determined through the use of validation data, optimizing specific performance criteria. Additionally, the parameter c governs the influence of the ℓ_p -norm on the objective function, ensuring the attainment of optimal results by minimizing said objective function. The considered values for these parameters is as follows:

Table 4.3: The hyper-parameters in the proposal approach

OCC fusion parameters	1	2	3	4	5	6	7
p	32/31	16/15	8/7	4/3	2	4	100
c	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	100

4.6 Results

In this section, the results acquired through the proposal method are analyzed in the context of three distinct configurations: (1)Results of each models, at each facial regions and CNN models for faceforensics++ dataset; (2)The pure approach, which is grounded in one-class classification (OCC) fusion in which outcomes are using uniform weights; (3)The pure approach, which is grounded in one-class classification (OCC) fusion in which outcomes are derived by learning weights (4) The non-pure approach, characterized by the application of the unseen/unknown method through cross-manipulation. This evaluation aims to provide a comprehensive understanding of the performance and implications of each setting within the context of the current study.

4.6.1 Evaluation on the Faceforensics++ dataset using single models

In the subsequent section, the Area Under the Curve (AUC) percentages for various models—namely, KPCA, SVDD, GPR, and GMM—are presented across distinct regions. These regions encompass the cropped face (R1), the combined area of the mouth and nose (R2), and solely the mouth (R3). The performance of these models is evaluated based on different Convolutional Neural Network (CNN) architectures, including AlexNet, Darknet19, and Inceptionv3. The data is drawn from Table 4.4 through Table 4.8, with the top-performing results highlighted.

Table 4.4: Model’s AUC(%) in DeepFakes High quality frame based

Models	R1_Alexnet	R1_Darknet19	R1_Inceptionv3	R2_Alexnet	R2_Darknet19	R2_Inceptionv3	R3_Alexnet	R3_Darknet19	R3_Inceptionv3
KPCA	95.06	89.68	89.62	92.71	89.38	87.75	88.22	88.13	84.50
SVDD	95.58	92.49	92.26	95.04	92.50	91.85	91.91	89.57	88.73
GPR	97.30	95.15	93.94	96.48	94.21	92.81	94.26	92.15	89.89
GMM	94.27	89.01	88.64	92.34	87.00	87.44	87.80	86.21	85.18

Table 4.5: Model’s AUC(%) in FaceShifter High quality frame based

Models	R1_Alexnet	R1_Darknet19	R1_Inceptionv3	R2_Alexnet	R2_Darknet19	R2_Inceptionv3	R3_Alexnet	R3_Darknet19	R3_Inceptionv3
KPCA	87.61	81.92	79.43	83.45	78.90	76.41	79.31	75.75	71.54
SVDD	89.30	84.31	82.65	87.04	82.54	81.29	83.05	78.26	76.10
GPR	92.76	88.79	86.21	90.85	87.14	83.62	87.32	83.74	78.66
GMM	86.97	81.16	80.59	82.98	78.29	77.13	79.42	75.90	72.57

Table 4.6: Model’s AUC(%) in FaceSwap High quality frame based

Models	R1_Alexnet	R1_Darknet19	R1_Inceptionv3	R2_Alexnet	R2_Darknet19	R2_Inceptionv3	R3_Alexnet	R3_Darknet19	R3_Inceptionv3
KPCA	91.37	83.63	83.32	86.78	82.92	82.00	84.62	81.36	80.57
SVDD	94.21	88.50	89.04	91.61	87.74	86.57	90.27	86.95	84.69
GPR	94.49	90.65	91.78	93.85	90.85	90.49	93.03	89.22	89.36
GMM	86.97	84.00	86.96	84.82	84.16	85.19	83.88	81.68	83.70

Table 4.7: Model’s AUC(%) in Face2Face High quality frame based

Models	R1_Alexnet	R1_Darknet19	R1_Inceptionv3	R2_Alexnet	R2_Darknet19	R2_Inceptionv3	R3_Alexnet	R3_Darknet19	R3_Inceptionv3
KPCA	71.53	68.21	69.11	71.64	70.75	71.00	63.84	67.20	64.49
SVDD	72.62	70.85	72.05	74.70	73.46	74.39	65.76	66.51	65.85
GPR	77.71	75.09	75.14	79.44	78.75	77.55	68.14	69.29	67.81
GMM	71.14	67.78	68.14	70.93	70.00	70.84	63.14	65.38	64.43

Table 4.8: Model’s AUC(%) in Neurltextures High quality frame based

Models	R1_Alexnet	R1_Darknet19	R1_Inceptionv3	R2_Alexnet	R2_Darknet19	R2_Inceptionv3	R3_Alexnet	R3_Darknet19	R3_Inceptionv3
KPCA	73.84	70.27	68.15	78.65	74.36	73.00	72.01	74.07	69.82
SVDD	70.43	70.09	67.76	78.20	75.54	75.48	72.94	72.08	72.06
GPR	75.38	73.19	70.00	79.03	76.89	74.62	73.42	73.07	71.01
GMM	72.93	68.17	66.18	76.25	70.47	71.66	70.09	70.26	69.60

4.6.2 uniform weights

In this subsection, the fusion stage for the extracted scores of one-class classification (OCC) models does not incorporate a learning method. Consequently, all weight components are assigned equal values, and thus all features are evaluated with equal importance. The following outcomes, derived on the datasets provided in Table 4.1, are presented at Table 4.9, delineating diverse methods of feature normalization and score normalization, specifically encompassing Z-score, Tanh, modified Z-score, and min-max approaches.

Regarding feature normalization, this research explores six distinct methodologies, namely the Z-score, Z-score with norm2, traditional min-max, traditional min-max with norm2, min-max 3.18, and min-max 3.18 with norm2 techniques. Through an analysis of the results, it becomes evident that the majority of datasets exhibit improved efficiency when the Z-score approach is employed as the preferred method for feature normalization.

Table 4.9: AUC's(mean $\pm$ std%) for the datasets in Table 4.1 for a uniform weight fusion scheme

Datasets	<i>Z - score</i>	<i>ModifiedZ - score</i>	<i>Tanh</i>	<i>Min - Max</i>
d1	86.24 $\pm$ 0.1	77.38 $\pm$ 0.3	86.24 $\pm$ 0.2	83.16 $\pm$ 0.3
d2	98.33 $\pm$ 0.1	94.90 $\pm$ 0.0	98.33 $\pm$ 0.1	98.33 $\pm$ 0.1
d3	61.76 $\pm$ 0.7	60.13 $\pm$ 0.5	61.76 $\pm$ 1.3	61.3 $\pm$ 1.3
d4	89.88 $\pm$ 1.0	88.09 $\pm$ 1.5	89.84 $\pm$ 0.6	89.88 $\pm$ 0.7
d5	91.69 $\pm$ 0.9	88.46 $\pm$ 1.2	91.69 $\pm$ 0.8	91.69 $\pm$ 1.2
d6	90.69 $\pm$ 0.6	92.85 $\pm$ 0.7	90.69 $\pm$ 1.0	90.69 $\pm$ 1.3
d7	67.76 $\pm$ 4.1	71.25 $\pm$ 4.4	67.76 $\pm$ 4.3	67.76 $\pm$ 4.4
d8	98.05 $\pm$ 0.6	98.76 $\pm$ 1.0	98.09 $\pm$ 1.4	98.05 $\pm$ 1.0
d9	89.61 $\pm$ 3.8	89.78 $\pm$ 4.3	89.61 $\pm$ 3.8	89.61 $\pm$ 3.6
d10	96.0 $\pm$ 0.3	96.0 $\pm$ 0.1	96.0 $\pm$ 0.3	96.0 $\pm$ 0.0
d11	96.79 $\pm$ 0.6	96.96 $\pm$ 1.1	96.87 $\pm$ 0.7	96.81 $\pm$ 1.3
d12	88.67 $\pm$ 6.2	91.64 $\pm$ 5.6	90.29 $\pm$ 6.2	88.67 $\pm$ 5.6

The results obtained from the FaceForensics++ dataset, considering varying levels of quality (high and low). The tables encompass four different normalization techniques: Z-score, Tanh, Modified Z-score, and Min-Max. The results

are evaluated based on video-based, which are computed by applying parallel fixed-based rules to the output of $\mathbf{S}^\top \boldsymbol{\omega}$, including median, minimum, maximum, and mean. Additionally, frame-based (FB) results are provided for comparison between Table 4.10 and Table 4.24.

Table 4.10: AUC(%) on the DeepFakes dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	99.4	98.8	99.4	99.5	98.22	98.1	90.7	97.4	98.4	92.12
Modified Z-score	99.5	99	99.4	94.1	97.42	95.65	93.4	97.9	92.5	89.73
Tanh	99.5	98.8	99.4	99.4	97.87	96.13	90.20	96.90	96.90	90.27
Min-Max	99.4	98.8	99.5	99.5	98.21	98.2	90.7	97.4	98.4	92.12

Table 4.11: AUC(%) on the FaceShifter dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	98.3	91	98.4	98.3	93.88	97.6	87.5	97.4	97.5	91.03
Modified Z-score	95.8	97	97.7	90.6	90.41	92.7	91.23	96.09	88.3	86.96
Tanh	97.9	88.4	98.4	98.6	92.95	96.61	86.72	97.4	96.88	89.27
Min-Max	98.3	91	98.5	98.3	93.89	97.5	87.5	97.4	97.5	91.04

4.6.2.1 Evaluation on the Faceforensics++ dataset using uniform weights fusion scheme

To assess the findings, we initially focus on the mean Area Under the Curve (AUC) values at the frame-based (FB) level for both high and low quality categories. This process aids in determining which normalization method exhibits superior performance, as illustrated in the Table 4.25:

As shown in Table 4.25, the $Z - score$ outcome surpasses other normalization approaches in terms of both high and low quality categories. The subsequent phase involves examining the mean AUC in the chosen normalization technique,

Table 4.12: AUC(%) on the FaceSwap dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	99.6	98.3	99.4	99.6	97.13	95.28	84.51	96.33	94.49	86.3
Modified Z-score	99.4	97.2	99.5	94.7	95.02	92.52	88.19	93.96	87.93	81.91
Tanh	99.6	98	99.4	99.6	96.49	95.01	81.36	96.85	95.54	84.71
Min-Max	99.6	98.2	99.4	99.6	97.13	95.01	84.25	96.33	94.49	86.3

Table 4.13: AUC(%) on the Face2Face dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	95.3	89.9	92.5	95.2	84.07	70.3	60.8	68.2	69.4	60.46
Modified Z-score	92.9	82.8	93.7	89.7	80.69	69.3	62.7	70.3	66.3	60.14
Tanh	95.1	87.8	92.6	95.7	82.64	70.3	60.7	69.2	71.5	60.12
Min-Max	95.1	89.6	92.3	95.2	84.05	70.5	60.9	68.1	69.2	60.29

Table 4.14: AUC(%) on the Neuraltxtures dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	96.4	83.6	96.9	95.6	83.95	70.5	58.2	75.5	70.6	59.99
Modified Z-score	94.9	93	96.4	91.3	83.66	71.5	73.2	76.4	69.2	61.24
Tanh	96.1	84.4	97.1	96.7	83.45	68.9	58.5	75.8	71.2	59.75
Min-Max	96.4	83.4	96.9	95.6	83.95	70.3	58.3	75	70.8	59.92

Table 4.15: ACC(%) on the DeepFakes dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	98.35	94.4	98.8	98.8	94.78	95.05	84.28	89.05	94.9	86.7
Modified Z-score	91.65	59.95	97.85	81.4	90.5	89.3	50.26	86.6	80.8	82.55
Tanh	85.6	92.15	98.7	96.8	88.01	81.96	79.64	88.79	93.04	82.23
Min-Max	98.4	94.35	98.85	98.8	94.76	95.05	84.41	89.05	94.95	86.7

Table 4.16: ACC(%) on the FaceShifter dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	96.1	84	91.9	95.4	89.74	94.2	81.77	87.76	93.75	86.15
Modified Z-score	87.9	50.4	89.35	78.15	83.87	86.35	50.39	83.85	77.25	80.43
Tanh	82.75	81.05	91.85	94.6	84.43	81.12	79.43	87.24	91.93	81.56
Min-Max	96.25	84.1	91.95	95.55	89.76	94.1	81.77	87.63	93.75	86.15

Table 4.17: ACC(%) on the FaceSwap dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	97.85	96.4	96.25	97.95	92.22	90	74.54	79.92	89.1	79.64
Modified Z-score	92.15	50.15	95.4	82.2	86.61	82.1	50.39	75.07	75.95	76.74
Tanh	85.7	94.75	96.3	96.55	85.87	77.56	70.47	79.66	90.16	72
Min-Max	97.85	96.4	96.3	97.9	92.22	90.2	74.67	80.05	89.25	79.66

Table 4.18: ACC(%) on the Face2Face dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	88	76.2	69.35	87.5	77.23	63.85	57.65	52.3	63.55	57.42
Modified Z-score	82.5	50.85	73	75	73.47	62.1	51.55	54.7	60.35	57.22
Tanh	75.4	70.5	69	88.05	75.01	58.7	54.9	52.4	65	57.21
Min-Max	87.95	76.2	69.55	87.55	77.22	63.85	57.7	52.3	63.45	57.43

Table 4.19: ACC(%) on the Neurltextures dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	88.6	68.7	72.7	87	76.44	62.7	55.95	53.35	61.9	56.78
Modified Z-score	85	50.25	78.75	76.7	75.67	63.7	50.75	57.3	59.95	57.86
Tanh	76.45	64.1	72.85	87.95	74.5	58.2	53.95	53.15	62.3	56.45
Min-Max	88.55	68.7	72.75	87	76.43	62.65	56.1	53.55	61.5	56.77

Table 4.20: EER(%) on the DeepFakes dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	0.3	0.9	0.3	0.2	2.55	1.8	9.2	2.5	1.5	11.28
Modified Z-score	0.2	0.7	0.3	5.6	3.54	4.25	6.5	2	7.4	14.1
Tanh	0.2	0.9	0.3	0.3	3.34	3.61	9.54	2.84	2.84	14.08
Min-Max	0.3	0.9	0.2	0.2	2.57	1.7	9.2	2.5	1.5	11.35

Table 4.21: EER(%) on the FaceShifter dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	1.7	9	1.6	1.7	7.94	2.3	12.24	2.34	2.4	11.97
Modified Z-score	4.2	97	2.3	9.4	11.26	7.2	8.51	3.65	11.6	15.94
Tanh	2.1	11.6	1.6	1.4	9.19	3.13	13.02	2.34	2.86	13.51
Min-Max	1.7	9	1.5	1.7	7.93	2.4	12.24	2.34	2.4	11.95

Table 4.22: EER(%) on the FaceSwap dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	0.4	1.7	0.6	0.4	4.8	4.46	15.22	3.41	5.25	16.92
Modified Z-score	0.6	2.8	0.5	5.3	7.23	7.17	11.55	5.77	11.81	20.92
Tanh	0.4	2	0.6	0.4	5.89	4.72	18.37	2.89	4.2	18.87
Min-Max	0.4	1.8	0.6	0.4	4.85	4.72	15.49	3.41	5.25	16.88

Table 4.23: EER(%) on the Face2Face dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	4.5	9.9	7.3	4.6	20.74	29.6	39.1	31.7	30.5	41.78
Modified Z-score	6.9	17	6.1	10.1	23.74	30.6	37.2	29.6	33.6	41.85
Tanh	4.7	12	7.2	4.1	22.36	29.6	39.2	30.7	28.4	41.92
Min-Max	4.7	10.2	7.5	4.6	20.81	29.4	39	31.8	30.7	41.74

Table 4.24: EER(%) on the Neuraltextures dataset using an uniform weight fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	3.5	16.3	3	4.3	21.08	29.4	41.7	24.4	29.3	42.09
Modified Z-score	5	6.9	3.5	8.6	21.53	28.4	26.7	23.5	30.7	40.86
Tanh	3.8	15.5	2.8	3.2	21.72	31	41.4	24.1	28.7	42.07
Min-Max	3.5	16.5	3	4.3	21.07	29.6	41.6	24.9	29.1	41.98

Table 4.25: Average AUC(%) on the Faceforensics++ dataset through frame based results in the uniform weights

Normalizations	High quality FB	Low quality FB
Z-score	91.45	77.98
Modified Z-score	89.42	76
Tanh	90.68	76.83
Min-Max	91.44	77.93

with the aim to ascertain the video-based method demonstrating the most optimal performance.

Table 4.26: Average AUC(%) on the Faceforensics++ dataset through $Z - score$ results in the uniform weights

Normalizations	High quality VB	Low quality VB
Median	97.80	86.36
Min	92.32	76.34
Max	97.32	86.36
Mean	97.64	86.08

It is evident from the analysis that the optimal strategy for a video-based approach involves employing the Median technique for both high and low-quality datasets. A comparative evaluation of the frame-based outcomes, as presented in Table 4.10 and Table 4.24, against the results of singular models delineated from Table 4.4 to Table 4.8, underscores a significant observation. Specifically,

the amalgamation of diverse features consistently outperforms the efficacy of individual models, especially when assessing varied manipulated samples derived from the FaceForensics++ dataset.

4.6.3 Pure approach based on one-class classification

The results of pure learning-based approach is given from Table 4.45 to Table 4.42.

Table 4.27: AUC's(mean $\pm$ std%) for the datasets in Table 4.1 for a pure approach learning based fusion scheme

Datasets	<i>Z - score</i>	<i>ModifiedZ - score</i>	<i>Tanh</i>	<i>Min - Max</i>
d1	89.00 $\pm$ 0.7	87.84 $\pm$ 0.9	89.00 $\pm$ 0.7	88.32 $\pm$ 0.7
d2	99.97 $\pm$ 0.0	99.92 $\pm$ 0.0	99.97 $\pm$ 0.0	99.97 $\pm$ 0.0
d3	89.13 $\pm$ 0.9	88.86 $\pm$ 1.1	89.17 $\pm$ 0.9	88.59 $\pm$ 1.3
d4	94.29 $\pm$ 1.6	94.29 $\pm$ 1.6	94.29 $\pm$ 1.6	94.29 $\pm$ 1.6
d5	97.23 $\pm$ 1.5	97.47 $\pm$ 2.0	97.23 $\pm$ 1.6	97.23 $\pm$ 1.5
d6	96.99 $\pm$ 1.0	96.65 $\pm$ 1.0	96.99 $\pm$ 1.0	96.98 $\pm$ 1.0
d7	71.20 $\pm$ 4.2	71.20 $\pm$ 4.2	71.20 $\pm$ 4.2	71.20 $\pm$ 4.2
d8	98.12 $\pm$ 1.2	97.69 $\pm$ 1.2	98.15 $\pm$ 1.2	98.12 $\pm$ 1.2
d9	92.37 $\pm$ 4.4	92.58 $\pm$ 4.4	92.39 $\pm$ 4.4	92.39 $\pm$ 4.4
d10	99.95 $\pm$ 0.0	99.95 $\pm$ 0.0	99.95 $\pm$ 0.0	99.95 $\pm$ 0.0
d11	98.18 $\pm$ 1.2	98.34 $\pm$ 1.0	98.18 $\pm$ 1.2	98.18 $\pm$ 1.2
d12	92.83 $\pm$ 6.3	92.53 $\pm$ 6.4	92.83 $\pm$ 6.3	92.83 $\pm$ 6.3

Table 4.28: AUC(%) on the DeepFakes dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	99.9	98.8	99.8	99.9	98.41	98.5	92.3	97.9	98.5	92.61
Modified Z-score	99.3	99.1	99.4	95.5	97.4	95.75	95.8	96.8	93	89.36
Tanh	99.6	98.3	99.4	99.5	97.84	97.4	90.4	97.9	98.1	91.23
Min-Max	99.6	98.5	99.5	99.6	98.31	98.5	92.2	97.8	98.4	92.52

Table 4.29: AUC(%) on the FaceShifter dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	98.3	92	98.6	98.5	94.39	97.6	89.8	97.8	97.7	91.54
Modified Z-score	96.1	96.4	97.9	92.7	90.48	94	94	96.4	88.6	87.27
Tanh	98.1	89.1	98.7	98.7	93.29	96.5	86.7	98.2	97.9	89.99
Min-Max	98.4	92.1	98.7	98.5	94.38	97.4	89.7	97.7	97.6	91.43

Table 4.30: AUC(%) on the FaceSwap dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	99.8	98.6	99.3	99.6	97.2	95.8	84.9	96.3	95.4	86.6
Modified Z-score	99.1	1.7	99.6	95	95.62	95.75	91.5	96.8	93	81.64
Tanh	99.6	98.3	99.3	99.7	96.45	93.9	80.3	96.7	95.4	84.79
Min-Max	99.8	98.6	99.3	99.6	97.11	90.65	8.4	93.3	86.7	86.48

Table 4.31: AUC(%) on the Face2Face dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	95.5	89.9	92.9	95.8	84.57	72.2	62.5	67.7	71.5	61.01
Modified Z-score	85.05	51.4	86.6	87.8	64.92	69.7	63.9	72	68.5	59.85
Tanh	95.1	87.6	92.6	95.7	82.76	71.8	60.3	68	73.4	60.62
Min-Max	94.9	89.5	92.9	95	84.21	72.2	62.5	67.6	71.6	60.94

Table 4.32: AUC(%) on the Neuraltxtures dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	97.7	88.2	97.4	97.3	84.1	72	59.2	76.9	72	60.78
Modified Z-score	94.7	91.1	96.2	91.3	82.32	71	70.3	74.6	71.1	60.16
Tanh	96.5	83.8	95.8	96.6	83.43	71.1	58.9	76.2	72.6	60.33
Min-Max	96.6	84	96.2	96.1	84.03	71.7	59	74.1	71.8	60.58

Table 4.33: ACC(%) on the DeepFakes dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	98.55	95.1	98.8	98.9	94.58	96.05	85.4	91.15	95.65	87.21
Modified Z-score	88.4	59.95	98.4	83.15	89.29	86.6	50.5	90.55	81.25	81.71
Tanh	90.15	93.3	98.8	97.7	88.72	84.65	81.5	91.25	94.9	83.14
Min-Max	98.55	95.15	98.85	98.85	94.57	95.9	85.5	91.2	95.45	87.12

Table 4.34: ACC(%) on the FaceShifter dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	97	86.25	93.4	96.4	89.92	95.25	82.95	90.1	94.8	86.5
Modified Z-score	86.45	50.5	90.95	79.65	83.19	84	50.35	87.25	78	80.09
Tanh	85.9	82.25	93.05	95.65	84.95	83.95	79.1	90.25	93.8	82.4
Min-Max	96.95	86.25	93.5	96.3	89.88	95	83	89.85	94.7	86.42

Table 4.35: ACC(%) on the FaceSwap dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	98.05	96.5	97.05	98.15	92.03	90.85	75.45	82.25	89.8	80.12
Modified Z-score	88.4	50.1	97.2	82.4	86.52	81.6	50.95	79.4	76.6	74.36
Tanh	88.95	95.5	96.8	97.2	86.35	79.45	70.75	82	90.05	76.75
Min-Max	98.05	96.45	97	98.15	91.99	90.55	75.4	82.15	89.7	80.05

Table 4.36: ACC(%) on the Face2Face dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	88.9	77.85	72.1	88.05	77.63	65.45	60.25	56.2	65.2	57.98
Modified Z-score	71.7	70.55	91.05	77.25	61.27	62.25	52.55	59.3	60.25	56.89
Tanh	75.4	70.65	69	87.9	75.04	62.35	57.85	55.75	65.05	57.65
Min-Max	88.7	78.25	73.75	88.25	77.43	65.3	60.4	56.2	65	57.96

Table 4.37: ACC(%) on the Neuraltextures dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	89.25	70	75.25	87.9	76.76	64.25	58.4	57.55	64.15	57.47
Modified Z-score	83.8	51.4	81.2	78.85	74.69	62.05	52.25	61.9	60.9	57.09
Tanh	77.5	65.65	74.95	88.55	74.63	59.8	55.85	57.4	63.45	57.06
Min-Max	89.35	70.4	75.2	87.85	76.71	64.25	58.3	57.65	64.1	57.44

Table 4.38: EER(%) on the DeepFakes dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	0.1	1.2	0.2	0.1	2.67	1.4	7.6	2	1.4	11.21
Modified Z-score	0.4	0.6	0.3	4.2	3.87	4.15	4.1	3.1	6.9	14.87
Tanh	0.1	1.4	0.3	0.2	3.64	2.5	9.5	2	1.8	12.89
Min-Max	0.1	1.2	0.2	0.1	2.7	1.4	7.7	2.1	1.5	11.19

Table 4.39: EER(%) on the FaceShifter dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	1.7	8	1.4	1.5	7.57	2.3	10.1	2.1	2.2	11.89
Modified Z-score	3.9	3.6	2.1	7.3	11.25	5.9	5.9	3.5	11.3	16.03
Tanh	1.9	10.9	1.3	1.3	9.05	3.4	13.2	1.7	2	13.63
Min-Max	1.6	7.9	1.3	1.5	7.55	2.5	10.2	2.2	2.3	11.88

Table 4.40: EER(%) on the FaceSwap dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	0.2	1.4	0.7	0.4	5.28	4.1	15	3.6	4.5	17.17
Modified Z-score	0.9	98.3	0.4	5	6.98	9.23	4.1	6.6	13.2	21.64
Tanh	0.4	1.7	0.7	0.3	6.34	6	19.6	3.2	4.5	19.19
Min-Max	0.2	1.4	0.7	0.4	5.32	4.3	15	3.7	4.8	17.32

Table 4.41: EER(%) on the Face2Face dataset using pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	4.3	9.9	6.9	4	20.59	27.7	37.4	32.2	28.4	41.26
Modified Z-score	14.77	48.4	13.2	12	36.05	30.2	36	27.9	31.4	42.12
Tanh	4.7	12.2	7.2	4.1	22.38	28.1	39.6	31.9	26.5	41.69
Min-Max	4.9	10.3	6.9	4.8	20.84	27.7	37.4	32.3	28.3	41.19

Table 4.42: EER(%) on the Neuraltxtures dataset using pure approach learning based based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	3	15.6	3.9	3.5	20.78	27.9	40.7	23	27.9	41.4
Modified Z-score	5.2	8.8	3.7	8.6	22.55	28.9	29.6	25.3	28.8	41.66
Tanh	3.4	16.1	4.1	3.3	21.69	28.8	41	24.7	27.3	41.71
Min-Max	3.3	15.9	3.7	3.8	20.84	28.2	40.9	25.8	28.1	41.48

4.6.3.1 Evaluation on the Faceforensics++ dataset using Pure approach learning based fusion scheme

To conduct a similar evaluation as that performed for uniform weights, we initially calculate the mean Area Under the Curve (AUC) on the frame-based (FB) level for both high and low quality data sets. This step assists us in identifying the normalization method that surpasses others in performance, as displayed in the table below:

Table 4.43: Average AUC(%) of Faceforensics++ dataset through frame based results in Pure approach one-class classification

Normalizations	High quality FB	Low quality FB
Z-score	91.73	78.50
Modified Z-score	86.14	75.65
Tanh	90.75	77.39
Min-Max	91.60	78.39

Table 4.43 reveals that the $Z - score$ outcome outperforms other normalization methods for both high and low quality categories. The second step entails evaluating the mean AUC in the selected normalization technique, aiming to identify the video-based method that has achieved the highest performance

Table 4.44: Average AUC(%) of Faceforensics++ dataset through $Z - score$ results in Pure approach one-class classification

Normalizations	High quality VB	Low quality VB
Median	98.24	87.22
Min	93.5	77.74
Max	97.6	87.32
Mean	98.22	87.02

The observation suggests that employing the Median technique stands out as the most effective method for the video-based approach in both high and low quality datasets. Comparing the results of Table 4.26 relating to uniform weight and Table 4.44 relating to pure approach learning based, it can be observed that the best performance in pure learning based approach with 98.24 % and 87.32 % in both high and low qualities outperform the uniform weight fusion scheme with 97.80%, 86.36% on high and low quality respectively. It can be concluded that the pure approach in learning based is superior to uniform weight fusion scheme.

4.6.4 Non-pure approach of unseen/unknown method

It's been mentioned that the non-pure approach utilizes both positive and negative samples for predictions, with two subcategories: seen/known and unseen/unknown classifications. Addressing the challenge of identifying unseen/unknown manipulated fake faces involves using cross-manipulation and cross-dataset methods to improve a model's generalization capabilities. Cross-manipulation tests a model's robustness by applying image manipulations to a dataset not used during training.

Table 4.45: AUC's(mean $\pm$ std%) for the datasets in Table 4.1 for a non-pure approach learning based fusion scheme

Datasets	$Z - score$	$Modified Z - score$	$Tanh$	$Min - Max$
d1	90.22 $\pm$ 0.1	89.13 $\pm$ 0.4	90.47 $\pm$ 0.0	89.31 $\pm$ 0.3
d2	99.99 $\pm$ 0.0	99.98 $\pm$ 0.0	99.99 $\pm$ 0.0	99.99 $\pm$ 0.0
d3	89.73 $\pm$ 0.2	90.05 $\pm$ 0.6	90.32 $\pm$ 0.4	89.21 $\pm$ 1.4
d4	95.36 $\pm$ 1.1	96.77 $\pm$ 1.3	95.7 $\pm$ 1.4	96.29 $\pm$ 1.0
d5	99.68 $\pm$ 0.3	98.99 $\pm$ 0.4	98.36 $\pm$ 0.6	98.02 $\pm$ 1.0
d6	99.36 $\pm$ 0.5	98.02 $\pm$ 0.8	97.52 $\pm$ 1.0	98.16 $\pm$ 0.7
d7	73.53 $\pm$ 3.7	73.1 $\pm$ 4.2	72.04 $\pm$ 3.7	73.56 $\pm$ 3.7
d8	98.79 $\pm$ 1.0	99.51 $\pm$ 0.4	98.92 $\pm$ 1.0	99.99 $\pm$ 0.0
d9	93.80 $\pm$ 3.6	94.74 $\pm$ 3.5	94.22 $\pm$ 3.7	94.35 $\pm$ 3.8
d10	99.99 $\pm$ 0.0	99.98 $\pm$ 0.0	99.99 $\pm$ 0.0	99.99 $\pm$ 0.0
d11	98.40 $\pm$ 0.9	98.41 $\pm$ 0.7	98.40 $\pm$ 1.0	98.40 $\pm$ 1.3
d12	93.68 $\pm$ 5.5	93.65 $\pm$ 6.1	93.86 $\pm$ 5.9	93.38 $\pm$ 5.9

Table 4.46: AUC(%) on the DeepFakes dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	99.95	99.25	99.85	99.95	98.88	99.2	96.7	99.4	99.3	97.48
Modified Z-score	99.8	55.6	86.5	98.6	98.32	98	72.5	94.7	97.5	94.35
Tanh	99.85	99.25	99.85	99.85	98.55	99.4	97	99.5	99.5	97.42
Min-Max	99.75	99.15	99.75	99.75	98.69	99.2	96.7	99.3	99.2	97.48

Table 4.47: AUC(%) on the FaceShifter dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	99.4	98.3	99.5	99.5	98.68	97.7	89.2	98.2	97.9	91.72
Modified Z-score	98.4	51.4	75.8	96.2	96.19	97.6	68.6	92	96.5	91.07
Tanh	99.7	98.5	99.7	99.8	98.91	97.5	89.3	98.3	98	91.7
Min-Max	99.5	98.3	99.5	99.6	98.64	97.7	89.1	98.2	97.9	91.68

Table 4.48: AUC(%) on the FaceSwap dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	99.9	99.5	99.6	99.8	99.19	97.2	91.3	97.3	97.2	92.27
Modified Z-score	99	53.6	67	96.1	98.47	93.6	57.9	68.5	91.5	90.21
Tanh	99.8	99.6	99.6	99.8	99.29	97.3	91.6	97.8	97.3	92.05
Min-Max	99.8	99.5	99.6	99.8	99.16	97.3	91.3	97.4	97.1	92.27

Table 4.49: AUC(%) on the Face2Face dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	96.22	90.13	92.02	95.48	85.32	73.8	60.8	75.8	72.4	62
Modified Z-score	94.01	59.24	71.01	91.6	82.47	76.8	65.8	75.8	75.4	62.59
Tanh	96.22	89.29	92.02	96.43	84.84	74.1	61.3	76.6	73.7	62.19
Min-Max	96.22	90.13	92.23	95.48	85.31	73.8	60.9	75.8	72.4	61.93

Table 4.50: AUC(%) on the Neuraltxtures dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	97.93	87.79	96.73	97.49	85.48	74.2	56.1	79.3	73.4	61.63
Modified Z-score	96.92	55.62	75.11	94.38	85.63	78.1	64.4	77.9	77.6	62.69
Tanh	97.14	86.01	96.81	96.81	85.13	74.2	57.3	79.3	74.1	61.88
Min-Max	97.25	86.89	96.04	96.92	85.46	73.9	56.3	79.2	73	61.55

Table 4.51: ACC(%) on the DeepFakes dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	98.9	95.8	98.85	99.05	95.57	96.55	85.7	97.9	96.15	90.39
Modified Z-score	97.1	68.85	88.7	95.3	91.96	92.35	58.7	91.8	89.85	85.4
Tanh	96.9	94.05	98.85	98.9	94.14	95.1	84.95	97.4	96.45	89.72
Min-Max	98.9	95.8	98.9	99.05	95.68	96.5	85.8	97.95	96.2	90.38

Table 4.52: ACC(%) on the FaceShifter dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	96.85	85.55	98.95	96.35	92.43	94.5	82.6	91.45	94.35	86.34
Modified Z-score	93.95	69.35	80.35	90.75	87.8	91.4	59.8	89.25	89.75	83.65
Tanh	94.85	84.85	99.05	96.8	91.45	94.05	82.9	91.55	94.45	86.28
Min-Max	96.8	85.75	99	96.4	92.42	94.6	82.85	91.55	94.35	86.34

Table 4.53: ACC(%) on the FaceSwap dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	98.9	97.65	99.55	99	95.75	91.85	75.05	92.7	91.05	83.14
Modified Z-score	95.25	74.2	77.9	91.85	91.47	87.35	66.65	72.3	85.15	80.75
Tanh	97	96.7	99.4	98.9	93.52	91.1	75.3	92.1	91.4	82.82
Min-Max	98.95	97.65	99.55	98.95	95.74	91.8	75.4	92.8	91.05	83.14

Table 4.54: ACC(%) on the Face2Face dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	90.28	77.84	67.33	89.65	77.36	68.4	58.6	57.35	66.8	58.61
Modified Z-score	85.87	63.45	64.08	82.88	75.14	69.05	53.95	63.75	65.85	59.2
Tanh	86.03	73.27	66.96	89.81	76.96	67.7	58.75	57.5	67.2	58.73
Min-Max	90.28	78.2	67.49	89.65	77.35	68	58.75	57.7	66.8	58.53

Table 4.55: ACC(%) on the Neuraltextures dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	90.53	72.08	71.26	89.54	78.35	65.15	56.7	59.2	64.3	58.24
Modified Z-score	88.82	56.88	64.48	85.3	78.47	67.2	53.5	65.2	65.35	58.97
Tanh	86.62	67.46	71.04	89.92	77.23	65.2	56.8	59.4	64.6	58.48
Min-Max	90.47	72.03	71.42	89.37	78.34	65.1	56.8	59.45	64.25	58.22

Table 4.56: EER(%) on the DeepFakes dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	0.05	0.75	0.05	0.05	1.86	0.8	3.3	0.6	0.7	3.25
Modified Z-score	0.2	44.4	13.5	1.4	2.44	2	27.5	5.3	2.5	6.52
Tanh	0.25	0.85	0.25	0.25	2.05	0.6	3	0.5	0.5	3.43
Min-Max	0.25	0.85	0.25	0.25	1.86	0.8	3.3	0.7	0.8	3.21

Table 4.57: EER(%) on the FaceShifter dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	0.6	1.7	0.5	0.5	1.47	2.3	10.8	1.8	2.1	10.99
Modified Z-score	1.6	48.6	24.2	3.8	4.11	2.4	31.4	8	3.5	11.71
Tanh	0.3	1.5	0.3	0.2	1.19	2.5	10.7	1.7	2	11.01
Min-Max	0.5	1.7	0.5	0.4	1.49	2.3	10.9	1.8	2.1	10.88

Table 4.58: EER(%) on the FaceSwap dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	0.6	1.7	0.5	0.5	1.47	2.3	10.8	1.8	2.1	10.99
Modified Z-score	1.6	48.6	24.2	3.8	4.11	2.4	31.4	8	3.5	11.71
Tanh	0.3	1.5	0.3	0.2	1.19	2.5	10.7	1.7	2	11.01
Min-Max	0.5	1.7	0.5	0.4	1.49	2.3	10.9	1.8	2.1	10.88

Table 4.59: EER(%) on the Face2Face dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	3.05	7.33	6.92	4.07	19.21	26.2	39.2	24.2	27.6	40.58
Modified Z-score	3.46	43.18	26.27	5.7	20.33	23.2	34.2	24.2	24.6	40.33
Tanh	2.85	8.35	6.31	3.05	19.74	25.9	38.7	23.4	26.3	40.33
Min-Max	3.05	7.33	6.72	4.48	19.02	26.2	39.1	24.2	27.6	40.64

Table 4.60: EER(%) on the Neuralttextures dataset using non-pure approach learning based fusion scheme

Normalizations	High-Quality					Low-Quality				
	Median	Min	Max	Mean	FB	Median	Min	Max	Mean	FB
Z-score	2.53	12.78	3.74	2.97	19.37	25.7	43.8	20.6	26.5	40.63
Modified Z-score	2.86	44.16	24.67	5.4	19.46	21.8	35.5	22	22.3	39.9
Tanh	2.64	13.77	2.97	2.97	19.84	25.7	42.6	20.6	25.8	40.43
Min-Max	2.53	12.89	3.74	2.86	19.41	26	43.6	20.7	26.9	40.72

4.6.4.1 Evaluation results on the Faceforensics++ dataset for Non-pure approach cross-manipulation

In this section, mirroring previous methodologies, our initial step involves calculating the average Area Under the Curve (AUC) on a frame-based (FB) level for both high and low-quality datasets. This procedure is intended to determine the superior normalization method, as depicted in the given Table 4.61:

Table 4.61: Average AUC(%) of Faceforensics++ dataset through frame based results in Non-pure approach cross-manipulation

Normalizations	High quality FB	Low quality FB
Z-score	93.50	81.01
Modified Z-score	92.21	80.18
Tanh	93.34	81.04
Min-Max	93.45	80.98

Per Table 4.61, the *Z - score* and *Tanh* results rank higher than other normalization methods for high and low quality categories, respectively. The next step involves evaluating the mean AUC in the chosen normalization methods to pinpoint the video-based approach that showcases superior performance

We can see that in Table 4.62 using the Median technique for both high and low-quality datasets is the best method for the video-based approach. So, *Z - score* normalization and the Median approach work better than other normalization and video-based methods for all three methods, respectively. Based

Table 4.62: Average AUC(%) of video-base Faceforensics++ dataset through $Z - score$ results in Non-pure approach cross-manipulation

Normalizations	High quality VB	Low quality VB
Median	98.67	88.56
Min	94.99	69.84
Max	97.53	82.9
Mean	98.44	87.76

on this conclusion, it makes sense to compare the results of the $Z - score$ and median methods with those from the studies we mentioned, including both one-class classification and cross-manipulation results. It can be observed that the results in Table 4.62 with 98.67% and 88.56 % in both high and low qualities outperform the results in Table 4.44 relating to pure learning based fusion approach with 98.24% in high quality and 87.32% in low quality. According to Table 4.26, Table 4.44, and Table 4.62, non-pure approach superior to pure learning based, and the pure approach superior to the uniform weight fusion scheme.

4.6.5 Evaluation results on the datasets of Table 4.1 based for Frank-Wolfe method

Upon evaluating the results derived from the Frank-Wolfe algorithm, it is observed that they closely mirror those obtained using the CVX method. This similarity underscores the efficiency and accuracy of the Frank-Wolfe approach in the context of the problem at hand. The difference in weights between the two methods is shown at Table 4.63, calculated based on the $norm(\omega_{cvx} - \omega_{fw})$. Here, ω_{cvx} and ω_{fw} represent the weights associated with the convex method and the Frank-Wolfe, respectively.

However, when the weight differences are not significant, the outcomes from both the CVX method and the Frank-Wolfe are comparable. Importantly, if the Frank-Wolfe algorithm is allowed to iterate sufficiently, the differences in weights can be minimized to a very low value, further attesting to its capability. The

Table 4.63: norm difference of weights

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12
norm	0.0072	0.0197	0.3377	0.0081	0.3377	0.1018	0.2201	0.3378	0.0063	0.0003	0.0591	0.0070

following outcomes, derived from the data provided in Table 4.1, are presented herein, delineating diverse methods of feature normalization and score normalization (SN) techniques, specifically encompassing Z-score (SN1), modified Z-score (SN2), Tanh (SN3), and min-max Eq.3.18 (SN4) approaches. Regarding feature normalisation, this research explores six distinct methodologies, namely the Z-score, Z-score with norm2, traditional min-max, min-max with norm2, min-max Eq.3.18, and min-max Eq.3.18 with norm2 techniques. Through an analysis of the results, it becomes evident that the majority of datasets exhibit improved efficiency when the min-max with norm2 approach is employed as the preferred method for feature normalisation. And the evaluations of datasets in Table 4.1 are done by utilising friedmen’s test [78].

Table 4.64: Average ranking of different approaches based on Freidman’s test for the dataset of Table 4.1 ($p - value = 0.0005$) according to uniform weights, pure (CVX package), and non-pure (CVX package)

	Pure uniform weights				Pure(CVX package)				Non-pure (CVX package)			
	SN1	SN2	SN3	SN4	SN1	SN2	SN3	SN4	SN1	SN2	SN3	SN4
Ranking	10.5000	9.7083	10.2083	10.6667	6.5833	7.2083	6.2500	6.8750	2.5417	2.5000	2.5417	2.4167

Based on the outcomes of Friedman’s test and the data in Table 4.64, it is apparent that the min-max 3.18 (SN4) yield superior results for the non-pure learning-based method, with average ranks of 2.4167, compared to other techniques. This is further corroborated by the corresponding p-values of $p - value = 1.2029e - 19$, indicating statistical significance. Thus, it can be concluded that the learning-based method outperforms the fixed rule approach. In addition, in pure learning based approach min-max score normalization is outperformed the other normalization techniques in pure method. As previously mentioned, conditional gradient descent becomes a key factor when the processing time is significant, such as in the use of CVX package. The following Table 4.65 demonstrates the average

elapsed time in seconds for both CVX package and Frank-Wolfe algorithm.

Table 4.65: Elapsed time (seconds)

Dataset	Pure CVX package	Pure Frank-Wolfe
d1	109.51	10.14
d2	29.75	15.00
d3	182.78	27.26
d4	114.88	15.00
d5	90.15	11.00
d6	83.13	27.00
d7	173.48	15.00
d8	128.85	43.58
d9	20.05	6.03
d10	140.57	21.15
d11	100.29	91.45
d12	141.22	42.04
Avg.	109.55	27.05

As indicated in Table 4.65, the Frank-Wolfe demonstrates superior time efficiency compared to the CVX package, being on average 4 times faster. This underscores the significant potential for time savings when employing the Frank-Wolfe in this context.

4.7 Summary

In this section, to facilitate a comparison with the literature for FaceForensics++ and the dataset presented in Table 4.1, we divide the discussion into two categories: firstly, we compare the pure method (one-class classification) with its corresponding references; secondly, we contrast the non-pure method (cross-manipulation) approach with its relevant literature.

The pure learning-based fusion technique we proposed, which utilizes the Frank-Wolf algorithm combined with suitable score normalization (specifically,

min-max) as deduced from Table 4.64, can be benchmarked against existing research, such as Krawczyk et al. [79] propose dynamic classifier selection to create efficient one-class ensembles. Their proposal computes all classifier competencies for a validation example. The Gaussian potential function will estimate classifier capabilities across the decision space using these areas of expertise, and Fatemifar et al.[65] proposes a novel fitness function allows OCC ensemble design using only non-anomalous samples. Moreover, a new two-sided normalization method effectively handles long-tail non-anomalous data. And lastly, using a particle swarm optimization algorithm for determining weights results in the most efficient OCC fusion system. It is detailed in the subsequent table.

Table 4.66: Comparison of the mean AUC(%) for datasets referenced in Table 4.1 using the pure learning-based approach(One-Class Classification) with Tanh score normalization in our methodology.

Dataset	Wang et al.[23]	Krawczyk et al.[24]	Wang et al.[25]	Fatemifar et al.[26]	proposed pure learning based fusion
d1	-	83.40	-	84.10	89.00
d2	95.37	-	-	99.27	99.97
d3	-	-	93.6	75.99	89.17
d4	-	-	-	96.54	94.29
d5	-	-	-	-	97.23
d6	84.53	80.93	94.90	90.02	96.99
d7	72.36	65.29	67.10	69.54	71.20
d8	90.76	91.59	-	92.57	98.15
d9	83.00	86.89	62.10	-	92.39
d10	91.65	-	95.00	-	99.95
d11	90.77	84.55	-	97.43	98.18
d12	-	-	90.80	-	92.83

As indicated in Table 4.66, it is evident that the our proposed method exhibits superior performance in majority of the datasets compared to the existing research relating to one-class classification approach. Note that, some of the one-class methods didn't use some of the datasets so "-" is considered as the mean AUC(%) instead.

Relating to Faceforensics++ dataset, based on the evidence derived from Table 4.43 and further corroborated by Table 4.44, it has been established that

Z-score normalization and median-based evaluations for video data outperform their respective counterparts. This finding is critical when it comes to comparing various normalization methods and video level approaches. As a step forward in enhancing the comparability between existing literature and our current methodology, we will place an emphasis on the utilization of Z-score normalization and median results. This decision is predicated on their superior performance in our evaluations, thus providing a more robust and reliable analysis. In prior discourse, it was highlighted that One-Class Classification (OCC) can be defined as a pure methodology wherein only single-class data is utilized throughout the training phase. In the following table, we present the best performance metrics for each face manipulation technique utilized within the FaceForensics++ framework. Performance is evaluated based on the Area Under the Curve (AUC) metrics, as determined through our chosen normalization and video level methods. This summary provides a comprehensive comparison and benchmark of various techniques within the context of this particular framework.

Table 4.67: AUC (%) of Faceforensics++ dataset through video based (VB) results in Pure approach one-class classification

Data	High quality VB	Low quality VB
Deepfakes	99.9	98.5
FaceShifter	98.3	97.6
FaceSwap	99.8	95.8
Face2Face	95.5	72.2
Neuraltextures	97.7	72
Avg.	0.9824	0.8722

Referring to Table 4.68, a comparative analysis is provided between our results and those from the Oc-Fake model [34] based on performance. Notably, the latter stands as the sole study employing the One-Class Classification approach within the context of the FaceForensics++ dataset.

In reference to Table 4.68, we have highlighted the superior outcomes using bold text. A careful analysis suggests that our proposed methodology surpasses

Table 4.68: Comparing performance in One-Class Classification method on Faceforensics++ dataset in Low quality

<i>Methods</i>	<i>Deepfake</i>	<i>FaceSwap</i>	<i>FaceShifter</i>	<i>Face2Face</i>	<i>Neuraltextures</i>
OC-FakeDect1[34]	86.20	84.80	-	70.70	95.30
OC-FakeDect2[34]	88.40	86.10	-	71.20	97.50
Ours	98.5	95.8	97.6	72.2	72

the performance of the method proposed by Khalid et al.[34] for most face manipulation techniques, the exception being Neuraltextures.

4.7.1 Non-pure method (cross-manipulation) comparisons

Similar to the evaluation process of the pure method, we refer to Table 4.61 and subsequently, Table 4.62, for assessing the non-pure method. As inferred from these tables, Z-score normalization and median-based video level have been identified as the most efficacious techniques. Consequently, these two methods are selected to facilitate a comprehensive comparison with various cross-manipulation studies. This selection is intended to ensure our analysis aligns with the most robust and effective approaches, thereby enhancing the validity of our research outcomes. The subsequent table delineates the highest performance outcomes associated with each face manipulation technique as implemented within the FaceForensics++ framework. The comparative metric of performance is the Area Under the Curve (AUC), which has been ascertained in accordance with the employed normalization and video level methodologies.

Drawing from the data in Table 4.69, we are equipped to conduct a comparison with the non-pure cross-manipulation studies over faceforensics++ dataset. The succeeding tables facilitate this comparative analysis, thereby illuminating the performance of different techniques within the context of non-pure cross-manipulation:

Table 4.69: AUC (%) of Faceforensics++ dataset through video based (VB) results in Non-pure method (cross-manipulation)

Data	High quality VB	Low quality VB
Deepfakes	99.95	99.2
FaceShifter	99.4	99.2
FaceSwap	99.9	97.2
Face2Face	96.21	73.8
Neuraltextures	97.92	74.2
Avg.	98.67	88.72

Table 4.70: Comparing AUC (%) in cross-manipulation method on Faceforensics++ dataset in Low quality

<i>Methods</i>	<i>Deepfake</i>	<i>FaceSwap</i>	<i>FaceShifter</i>	<i>Face2Face</i>	<i>Neuraltextures</i>
Miao et al.[21]	86.80	-	-	73.01	-
Sun et al.[22]	75.6	68.1	-	72.4	60.8
Miao et al($F^2Trans - S$).[23]	86.92	-	-	76.52	-
Miao et al($F^2Trans - B$).[23]	88.77	-	-	77.73	-
Tran et al.[24]	74.7	-	-	72.7	-
Dong et al.[25]	81.8	69.9	-	72.5	62.6
Wang et al.[26]	-	-	-	-	-
Zhu et al.[27]	-	-	-	-	-
Haliassos et al.[28]	-	-	-	-	-
Ours	99.2	97.2	99.2	73.8	74.2

Table 4.71: Comparing AUC (%) in cross-manipulation method on Faceforensics++ dataset in High quality

<i>Methods</i>	<i>Deepfake</i>	<i>FaceSwap</i>	<i>FaceShifter</i>	<i>Face2Face</i>	<i>Neuraltextures</i>
Miao et al.[21]	-	-	-	-	-
Sun et al.[22]	92.7	64	-	80.2	77.3
Miao et al($F^2Trans - S$).[23]	97.47	-	-	90.55	-
Miao et al($F^2Trans - B$).[23]	98.92	-	-	94.08	
Tran et al.[24]	90.2	-	-	80.1	
Dong et al.[25]	94.1	65.6	-	81.4	79.2
Wang et al.[26]	92.9	70.5	-	90.1	91.9
Zhu et al.[27]	99.45	83.67	-	94.64	79.22
Haliassos et al.[28]	99.7	90.1	-	99.7	99.1
Ours	99.95	99.9	99.4	0.96.21	97.92

The findings of our study, as illustrated in Table 4.70, clearly demonstrate a superior performance in cross-manipulation under low-quality evaluations compared to the majority of the existing literature. However, it should be noted that Miao et al.[23] showed a marginal advantage, improving the performance by approximately 3.5%, but this was specifically confined to the Face2Face manipulation type. Moreover, according to Table 4.71, our results exceed those of most other studies in high quality, with the exception of Haliassos et al.[28], who obtained slightly better results in Face2Face and Neural textures manipulation, surpassing our performance by nearly 2.7% and 1.4%, respectively. The majority of papers in the existing literature do not employ FaceShifter, although a few consider it under a cross-dataset approach. However, these findings are not directly comparable to our own due to our unique contribution—specifically, the preservation of personal identity (subject-dependent) during the evaluation of our method.

Chapter 5

Conclusion

Most of the previous work was centered on differentiating real from fake images or videos, but these techniques had their own limitations. Building traditional binary classification technique which were time-consuming, required a vast array of different Deepfake examples, and occasionally overlooked unique Deepfakes. In addition, Utilizing single classifiers causes to lose best performance chances. Last not at least in subject independent approaches, discerning patterns from varied data without considering the test subject, which can lead to individual tools being less effective due to the complexity of the patterns.

In light of this, our research brought forward several key findings and strategies:

- **Approaches:** We considered two main methods for Deepfake detection.
 - **First Strategy:** Used only real data for training and tested on both real and fake data.
 - **Second Strategy:** Used manipulated images or videos for training, which were different from the ones used in the testing phase.
 - **Third Strategy:** considering subject dependent strategy in these context.
- **Ensemble System:** Our system combined various models to detect more

Deepfakes.

- It employed an ℓ_p -norm regularization, giving both sparse and non-sparse results.

- **Dataset Findings:** Our tests on the UCI and Keel datasets showed:
 - Pure uniform weight (Sum rule) fusion performed better than single models.
 - Pure learning-based fusion was better than uniform weight fusion.
 - Non-pure learning-based fusion showed better results compared to the pure approach.
- **Optimal Techniques:** The best score normalization was Z-score, and the median fixed rule looked promising for video evaluations.
- **Speed:** The Frank-Wolfe gradient descent algorithm was about 4 times faster than the CVX package.
- **Comparison with Existing Research:** This study showed improvements in both pure learning-based fusion and non-pure learning-based fusion techniques compared to earlier research.
- **Efficiency:** Our new methods were good at spotting various fake face types. They performed best in evaluations using the DeepFakes, FaceSwap, and Neuraltextures samples of the FaceForensics++ dataset.

Future Work: We’re looking into using CNN models trained only on manipulated face samples, not using pre-trained ones. This could help spot fake faces even in difficult cases such as reenactment manipulated samples (Face2Face and Neuraltextures).

Bibliography

- [1] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “Deepfakes and beyond: A survey of face manipulation and fake detection,” *Information Fusion*, vol. 64, pp. 131–148, 2020.
- [2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [3] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.
- [4] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” *arXiv preprint arXiv:1710.10196*, 2017.
- [5] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, “Synthesizing obama: learning lip sync from audio,” *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.
- [6] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” *arXiv preprint arXiv:1611.01236*, 2016.
- [7] E. Gonzalez-Sosa, J. Fierrez, R. Vera-Rodriguez, and F. Alonso-Fernandez, “Facial soft biometrics for recognition in the wild: Recent works, annotation, and cots evaluation,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2001–2014, 2018.

- [8] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8789–8797, 2018.
- [9] Y. Liu, J. L. Bailey, and J. G. Davidsen, “Social-cultural ecosystem services of sea trout recreational fishing in norway,” *Frontiers in marine science*, vol. 6, p. 178, 2019.
- [10] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, “Face2face: Real-time face capture and reenactment of rgb videos,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2387–2395, 2016.
- [11] J. Thies, M. Zollhöfer, and M. Nießner, “Deferred neural rendering: Image synthesis using neural textures,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.
- [12] D. H. Wolpert, “Stacked generalization,” *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [13] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, “On combining classifiers,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 3, pp. 226–239, 1998.
- [14] D.-C. Stanciu and B. Ionescu, “Deepfake video detection with facial features and long-short term memory deep networks,” in *2021 International Symposium on Signals, Circuits and Systems (ISSCS)*, pp. 1–4, IEEE, 2021.
- [15] T.-A. To, H.-C. Luong, N.-T. Nguyen, T.-T. Nguyen, M.-T. Tran, and T.-L. Do, “Deepfake detection using efficientnet: Working towards dense sampling and frames selection,” in *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pp. 612–617, IEEE, 2022.
- [16] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.

- [17] Z. Ma, P. Xu, X. Mei, and J. Shen, “A novel facial manipulation detection method based on contrastive learning,” in *2022 IEEE 5th International Conference on Electronics Technology (ICET)*, pp. 1163–1167, IEEE, 2022.
- [18] X. Fu, S. Li, Y. Yuan, B. Li, and X. Li, “Forgery face detection via adaptive learning from multiple experts,” *Neurocomputing*, 2023.
- [19] Y. Luo, F. Ye, B. Weng, S. Du, and T. Huang, “A novel defensive strategy for facial manipulation detection combining bilateral filtering and joint adversarial training,” *Security and Communication Networks*, vol. 2021, pp. 1–10, 2021.
- [20] S. Chen, T. Yao, Y. Chen, S. Ding, J. Li, and R. Ji, “Local relation learning for face forgery detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1081–1088, 2021.
- [21] C. Miao, Z. Tan, Q. Chu, N. Yu, and G. Guo, “Hierarchical frequency-assisted interactive networks for face manipulation detection,” *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3008–3021, 2022.
- [22] K. Sun, H. Liu, Q. Ye, Y. Gao, J. Liu, L. Shao, and R. Ji, “Domain general face forgery detection by learning to weight,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 2638–2646, 2021.
- [23] C. Miao, Z. Tan, Q. Chu, H. Liu, H. Hu, and N. Yu, “F 2 trans: High-frequency fine-grained transformer for face forgery detection,” *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1039–1051, 2023.
- [24] V.-N. Tran, S.-H. Lee, H.-S. Le, B.-S. Kim, and K.-R. Kwon, “Learning face forgery detection in unseen domain with generalization deepfake detector,” in *2023 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 01–06, IEEE, 2023.
- [25] F. Dong, X. Zou, J. Wang, and X. Liu, “Contrastive learning-based general deepfake detection with multi-scale rgb frequency clues,” *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 4, pp. 90–99, 2023.

- [26] H. Wang, Z. Liu, and S. Wang, “Exploiting complementary dynamic incoherence for deepfake video detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
- [27] X. Zhu, H. Fei, B. Zhang, T. Zhang, X. Zhang, S. Z. Li, and Z. Lei, “Face forgery detection by 3d decomposition and composition search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [28] A. Haliassos, K. Vougioukas, S. Petridis, and M. Pantic, “Lips don’t lie: A generalisable and robust approach to face forgery detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5039–5049, 2021.
- [29] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.
- [30] S. S. Khan and M. G. Madden, “A survey of recent trends in one class classification,” in *Artificial Intelligence and Cognitive Science: 20th Irish Conference, AICS 2009, Dublin, Ireland, August 19-21, 2009, Revised Selected Papers 20*, pp. 188–197, Springer, 2010.
- [31] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, “Protecting world leaders against deep fakes,” in *CVPR workshops*, vol. 1, p. 38, 2019.
- [32] P. Ekman and W. V. Friesen, “Measuring facial movement,” *Environmental psychology and nonverbal behavior*, vol. 1, pp. 56–75, 1976.
- [33] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.
- [34] H. Khalid and S. S. Woo, “Oc-fakedect: Classifying deepfakes using one-class variational autoencoder,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 656–657, 2020.

- [35] D. A. Lisin, M. A. Mattar, M. B. Blaschko, E. G. Learned-Miller, and M. C. Benfield, “Combining local and global image features for object class recognition,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)-Workshops*, pp. 47–47, IEEE, 2005.
- [36] F. Rustam, A. A. Reshi, I. Ashraf, A. Mehmood, S. Ullah, D. M. Khan, and G. S. Choi, “Sensor-based human activity recognition using deep stacked multilayered perceptron model,” *IEEE Access*, vol. 8, pp. 218898–218910, 2020.
- [37] Z. Yin, M. Zhao, Y. Wang, J. Yang, and J. Zhang, “Recognition of emotions using multimodal physiological signals and an ensemble deep learning model,” *Computer methods and programs in biomedicine*, vol. 140, pp. 93–110, 2017.
- [38] M. Gour and S. Jain, “Automated covid-19 detection from x-ray and ct images with stacked ensemble convolutional neural network,” *Biocybernetics and Biomedical Engineering*, vol. 42, no. 1, pp. 27–41, 2022.
- [39] S. Fatemifar, M. Awais, A. Akbari, and J. Kittler, “Particle swarm and pattern search optimisation of an ensemble of face anomaly detectors,” in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 3622–3626, IEEE, 2021.
- [40] B. Yalamanchili, S. K. Samayamantula, and K. R. Anne, “Neural network-based blended ensemble learning for speech emotion recognition,” *Multidimensional Systems and Signal Processing*, vol. 33, no. 4, pp. 1323–1348, 2022.
- [41] M. A. Hedeya, A. H. Eid, and R. F. Abdel-Kader, “A super-learner ensemble of deep networks for vehicle-type classification,” *IEEE Access*, vol. 8, pp. 98266–98280, 2020.
- [42] N. Poh, T. Bourlai, J. Kittler, L. Allano, F. Alonso-Fernandez, O. Ambekar, J. Baker, B. Dorizzi, O. Fatukasi, J. Fierrez, *et al.*, “Benchmarking quality-dependent and cost-sensitive score-level multimodal biometric fusion

- algorithms,” *IEEE transactions on information forensics and security*, vol. 4, no. 4, pp. 849–866, 2009.
- [43] G. L. Marcialis, F. Roli, and L. Didaci, “Personal identity verification by serial fusion of fingerprint and face matchers,” *Pattern Recognition*, vol. 42, no. 11, pp. 2807–2817, 2009.
- [44] R. E. Schapire, “The boosting approach to machine learning: An overview,” *Nonlinear estimation and classification*, pp. 149–171, 2003.
- [45] J. Gama and P. Brazdil, “Cascade generalization,” *Machine learning*, vol. 41, no. 3, pp. 315–343, 2000.
- [46] S. Fatemifar, M. Awais, A. Akbari, and J. Kittler, “Developing a generic framework for anomaly detection,” *Pattern Recognition*, vol. 124, p. 108500, 2022.
- [47] D. M. Tax and R. P. Duin, “Combining one-class classifiers,” in *International Workshop on Multiple Classifier Systems*, pp. 299–308, Springer, 2001.
- [48] C. Lai, D. M. J. Tax, R. P. W. Duin, E. Pekalska, and P. Paclik, “On combining one-class classifiers for image database retrieval,” in *Multiple Classifier Systems* (F. Roli and J. Kittler, eds.), (Berlin, Heidelberg), Springer Berlin Heidelberg, 2002.
- [49] C. Bergamini, L. S. Oliveira, A. L. Koerich, and R. Sabourin, “Combining different biometric traits with one-class classification,” *Signal processing*, vol. 89, no. 11, pp. 2117–2127, 2009.
- [50] V. Di Gesù and G. Lo Bosco, “Combining one class fuzzy knn’s,” in *International Workshop on Fuzzy Logic and Applications*, pp. 152–160, Springer, 2007.
- [51] S. Fatemifar, M. Awais, S. R. Arashloo, and J. Kittler, “Combining multiple one-class classifiers for anomaly based face spoofing attack detection,” in *2019 International Conference on Biometrics (ICB)*, pp. 1–7, IEEE, 2019.

- [52] S. Fatemifar, M. Awais, A. Akbari, and J. Kittler, “A stacking ensemble for anomaly based client-specific face spoofing detection,” in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 1371–1375, IEEE, 2020.
- [53] Y. Zhang, B. Zhang, F. Coenen, J. Xiao, and W. Lu, “One-class kernel subspace ensemble for medical image classification,” *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–13, 2014.
- [54] C. Desir, S. Bernard, C. Petitjean, and L. Heutte, “One class random forests,” *Pattern Recognition*, vol. 46, no. 12, pp. 3490–3506, 2013.
- [55] Y. Ding and A. Ross, “An ensemble of one-class svms for fingerprint spoof detection across different fabrication materials,” in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, 2016.
- [56] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005.
- [57] C. C. Aggarwal and S. Sathe, “Theoretical foundations and algorithms for outlier ensembles,” *Acm sigkdd explorations newsletter*, vol. 17, no. 1, pp. 24–47, 2015.
- [58] S. Maldonado, R. Weber, and F. Famili, “Feature selection for high-dimensional class-imbalanced data sets using support vector machines,” *Information sciences*, vol. 286, pp. 228–246, 2014.
- [59] Z. Xu, R. Jin, I. King, and M. Lyu, “An extended level method for efficient multiple kernel learning,” *Advances in neural information processing systems*, vol. 21, 2008.
- [60] C. Cortes, A. Gretton, G. Lanckriet, M. Mohri, and A. Rostamizadeh, “Kernel learning: Automatic selection of optimal kernels,” in *NIPS 2008 Workshop: Kernel Learning: Automatic Selection of Optimal Kernels*, p. 94, 2008.
- [61] H. Zou and M. Yuan, “The ∞ norm support vector machine,” *Statistica Sinica*, pp. 379–398, 2008.

- [62] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” Mar. 2014.
- [63] D. Liu, K. Sun, Z. Wang, R. Liu, and Z.-J. Zha, “Frank-wolfe network: An interpretable deep structure for non-sparse coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 3068–3080, 2019.
- [64] S. Fatemifar, S. R. Arashloo, M. Awais, and J. Kittler, “Client-specific anomaly detection for face presentation attack detection,” *Pattern Recognition*, vol. 112, p. 107696, 2021.
- [65] S. Fatemifar, M. Awais, A. Akbari, and J. Kittler, “Developing a generic framework for anomaly detection,” *Pattern Recognition*, vol. 124, p. 108500, 2022.
- [66] F. R. Hampel, E. M. Ronchetti, P. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*. Wiley-Interscience; New York, 1986.
- [67] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “Faceforensics++: Learning to detect manipulated facial images,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1–11, 2019.
- [68] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, “Two-stream neural networks for tampered face detection,” in *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 1831–1839, IEEE, 2017.
- [69] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, “Advancing high fidelity identity swapping for forgery detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5074–5083, 2020.
- [70] D. M. Tax and R. P. Duin, “Support vector data description,” *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

- [71] M. Kemmler, E. Rodner, E.-S. Wacker, and J. Denzler, “One-class classification with gaussian processes,” *Pattern recognition*, vol. 46, no. 12, pp. 3507–3518, 2013.
- [72] B. Schölkopf, A. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [73] H. Hoffmann, “Kernel pca for novelty detection,” *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.
- [74] A. Bánhalmi, A. Kocsor, and R. Busa-Fekete, “Counter-example generation-based one-class classification,” in *European Conference on Machine Learning*, pp. 543–550, Springer, 2007.
- [75] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.
- [76] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.
- [77] J. Redmon, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [78] M. Friedman, “The use of ranks to avoid the assumption of normality implicit in the analysis of variance,” *Journal of the american statistical association*, vol. 32, no. 200, pp. 675–701, 1937.
- [79] B. Krawczyk and M. Woźniak, “Dynamic classifier selection for one-class classification,” *Knowledge-Based Systems*, vol. 107, pp. 43–53, 2016.