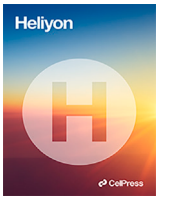
Heliyon 10 (2024) e25933

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

Deepfake detection using deep feature stacking and meta-learning

Gourab Naskar [a], Sk Mohiuddin [b], Samir Malakar [c], Erik Cuevas [d,*], Ram Sarkar [a]

[a] Department of Computer Science and Engineering, Jadavpur University, Kolkata, 700032, India
[b] Department of Computer Science, Asutosh College, Kolkata, 700026, India
[c] Department of Computer Science, UiT The Arctic University of Norway, Tromsø, 9019, Norway
[d] Departamento de Electrónica, Universidad de Guadalajara, C.P. 44430, Mexico

ARTICLE INFO

Keywords:
Deepfake
Stacking based ensemble
Deep learning
Feature selection
Meta-learning

ABSTRACT

Deepfake is a type of face manipulation technique using deep learning that allows for the replacement of faces in videos in a very realistic way. While this technology has many practical uses, if used maliciously, it can have a significant number of bad impacts on society, such as spreading fake news or cyberbullying. Therefore, the ability to detect deepfake has become a pressing need. This paper aims to address the problem of deepfake detection by identifying deepfake forgeries in video sequences. In this paper, a solution to the said problem is presented, which at first uses a stacking based ensemble approach, where features obtained from two popular deep learning models, namely Xception and EfficientNet-B7, are combined. Then by selecting a near-optimal subset of features using a ranking based approach, the final classification is performed to classify real and fake videos using a meta-learner, called multi-layer perceptron. In our experimentation, we have achieved an accuracy of 96.33% on Celeb-DF (V2) dataset and 98.00% on the FaceForensics++ dataset using the meta-learning model both of which are higher than the individual base models. Various types of experiments have been conducted to validate the robustness of the current method.

1. Introduction

Technology has made significant advancements in this digital era. Deepfake is one of the special features where a fake image can be constructed or someone's face can be imposed on some real images. It is a fact that deepfakes have some advantages in the digital world. However, the problem arises when deepfakes are used with the wrong intentions. Deepfakes can be easily misused in a malicious way when deepfake images or videos are spread for the wrong reasons. There are many reasons for making and spreading deepfakes, such as spreading propaganda, creating controversies, defaming celebrities or public figures, advancing a political agenda, or just for fun [30]. Deepfakes can be constructed in many ways, such as visual, textual, audio, or multimodal. However, in this research work, the focus is exclusively on videos and images of individuals' faces. In Deepfakes, the lips, eyes, and facial movements of an individual are captured and superimposed on another environment that generates a realistic image of the person in the simulated fake environment. As the world becomes more connected through social media, deepfakes news is spreading too easily. They are used to generate synthetic data on public figures, thereby spreading fake news and information over the internet.

* Corresponding author.
E-mail addresses: gourabn2000@gmail.com (G. Naskar), myselfmohiuddin@gmail.com (S. Mohiuddin), malakarsamir@gmail.com (S. Malakar), erik.cuevas@cucei.udg.mx (E. Cuevas), raamsarkar@gmail.com (R. Sarkar).

https://doi.org/10.1016/j.heliyon.2024.e25933
Received 23 August 2023; Received in revised form 25 January 2024; Accepted 5 February 2024

Available online 15 February 2024

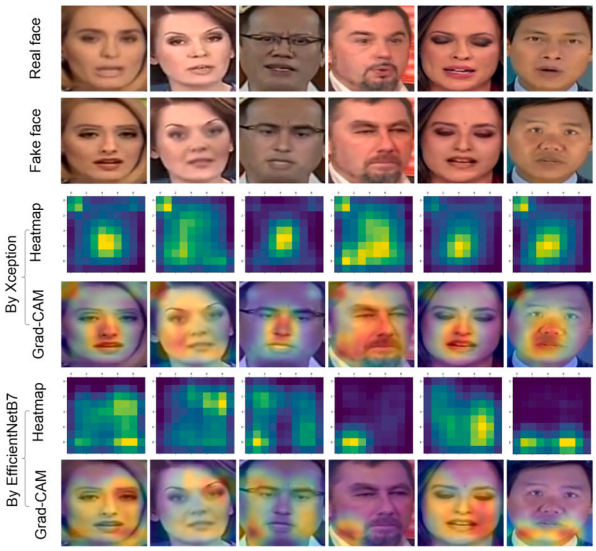


Fig. 1. The fine-tuned Xception and EfficientNet-B7 model was used to generate several counterfeit faces, and their visualizations were produced using heatmap and Grad-CAM techniques [38].

Models like Generative Adversarial Networks (GANs) [8], are very effective in generating Deepfakes. GANs have made it easy to generate synthetic images, videos, and audio content as real content. Deepfakes generated by GANs are so powerful that it becomes very tough to identify the difference between real and fake with traditional methods. The first deepfakes came into the limelight in 2017 when popular celebrities were targeted and fake videos of them spread over the internet. Deepfakes can affect the world and threaten world peace with controversial statements from politicians. Deepfakes can be generated by using synthesis and superimposing the real image. A GAN model contains two networks: a generator network (generates synthetic images out of the noise vector) and a discriminator network (classifies the produced synthetic images as real or fake). Both networks interact with each other, and the discriminator's feedback is given to the generator network. With time, the generator network learns to create synthetic content that looks very real and fools the discriminator. Thus, deepfakes created by the generator-discriminator network raise concerns about the authenticity of news and content on social platforms.

AttGAN [20], DeepFaceLab [33], Faceswap,[1] IcGAN [32], STGAN [26], StarGAN [5] and many others are some examples that create deepfakes. In DeepFaceLab, a person's face can be swapped with others, the age of a person can be changed, and the lip and eye movements of a person can be synchronized in a video. Deepfakes are used to spread propaganda. For example, a video of 44^{th} US President Barack Obama published on BuzzFeed showed that he was defaming former US president, Donald Trump, but the entire video was created by GAN technology. It went viral, but the whole content was fake. Thus, the authenticity of news is a matter of huge concern in this era of synthetic content. To identify the highly accurate synthetic products created by GANs as fake, a good deepfake detection tool is needed. Most of the approaches published in the literature point to designing a powerful deepfake detection model that lacks effectiveness and robustness in training deepfake detection. The robustness of a model means it is able to detect both high and low quality fake images and videos. The performance of the system would not vary with changing the resolution of the content. Recent research works [12,43,7,42] suggest that convolutional neural networks (CNN) based models are very effective at detecting deepfakes. Hence, in the present work, we aim to develop a CNN based robust system for the said purpose.

Gradient-weighted Class Activation Mapping (Grad-CAM) provides a visualization of the areas in an image that are crucial for a deep learning model's prediction. The heatmap image helps understand the model's decision-making process and provides insights into its attention mechanism. Fig. 1 shows such information and shows that deep learning models do not always focus on the desired location. Hence, in the current paper, we have followed a stacking based ensemble approach, which stacks features obtained from two CNN models. After that, we have concentrated on learning the valuable features by removing the unnecessary ones.

In summary, the **key points** of our work can be listed as follows:

- We have proposed a stacking based ensemble method, where features generated by dual CNN models are stacked followed by the selection of optimal features and elimination of inconsistent features.
- Feature selection is performed by averaging the feature importance scores obtained from two machine learning models, sorting them in descending order, and then choosing the top-k% of the entire stacked features.
- Our method has been tested and validated using two widely used and difficult benchmark datasets, namely FaceForensics++ and Celeb-DF. The results demonstrate that our system surpasses the performance of numerous state-of-the-art methods.

[1] https://github.com/MarekKowalski/FaceSwap.

- We have conducted a thorough evaluation of our model's performance in comparison to state-of-the-art methods, and our results are highly promising. Additionally, our robustness assessment shows satisfactory performance when subjected to a brightness test.

The subsequent sections of this article are structured as follows: Section 2 provides an overview of prior research followed by motivation in Section 3 Our proposed method is detailed in Section 4, and Section 5 is dedicated to the presentation and discussion of obtained results. Later, strengths, limitations, and possible improvements of the proposed method are suggested in Section 6. Finally, Section 7 concludes the paper.

2. Related work

As deepfakes have become a serious issue, there has been a significant amount of research in the field of deepfake detection techniques. Deep learning based techniques are the most advanced methods to detect deepfakes. Researchers have already adopted several ways to generate an efficient deepfake detection system. Though there are several approaches, the underlying principles remain consistent in most of them. Below are some past methods discussed that dealt with the considered problem.

Implementing deepfake detection using customized features is a straightforward process that demands minimal time and computational resources to identify synthetic images effectively. Hence, numerous researchers in the past have endeavored to identify fake images with customized features and machine learning. Koopman et al. [22] combine several techniques to identify deepfake videos and distinguish them from authentic videos. The approach involves analyzing various visual cues, such as facial inconsistencies, unnatural movements, and temporal artifacts, which are commonly present in deepfake videos. Durall et al. [11] employ the Discrete Fourier Transform (DFT) to identify abnormal attributes within counterfeit videos. Subsequently, these features are transformed into a 1D power spectrum in the spatial domain. The resulting spectrum is then input into logistic regression and support vector machine (SVM) classifiers for classification purposes. Yang et al. [44] introduce a novel technique that estimates head poses using 68 facial landmarks extracted from the central regions of the face. Their study demonstrates that when images are manipulated, the positions of these landmarks undergo shifts.

Guarner et al. [16] present a comparable approach that applies the expectation maximization (EM) algorithm to each channel of the input frame. This algorithm extracts the correlation between pixels and subsequently employs various classifiers to classify the image as either genuine or counterfeit. The authors, Li et al. [23] proposed a novel approach that exploits the inconsistencies and irregularities present in the color channels of Deepfake images. By examining the variations between the color components, they develop a detection method capable of distinguishing between genuine and artificially generated images. While these methods have demonstrated significant success in detecting deepfake generated in earlier periods, their performance has proven inadequate in light of the recent advancements in deepfake production techniques.

Recently, due to the use of GANs in deepfake creation, the quality of synthetic images has improved so much that many traditional methods fail to detect fake images. So traditional ways are gradually replaced by deep learning-based methods as their auto-learning of features produces better results than their hand-crafted counterparts. A hybrid method proposed by Guera et al. [17] based on CNN and long short-term memory (LSTM) networks. A CNN model extracts deep learning features from multiple sequential frames, and then those are transferred to the LSTM after concatenation of the features to find fake faces. In their study, Afchar et al. [2] developed a model that consists of two distinct CNN architectures. Firstly, they create Meso-4, which is a CNN network comprising four convolutional layers followed by a fully connected layer. Secondly, they introduce MesoInception-4, which replaces the first two layers of Meso-4 with a modified version of the Inception module. A two-stream network is proposed by Zhou et al. [45] for detecting face manipulations in video. In the first stream, a face classification network is trained based on CNN for capturing tampering artifact evidence, and in the second stream, a steganalysis based triplet network is trained to control the local noise residual evidence capturing functions. Li et al. [24] detect warping artifacts in deepfakes by training four different CNNs on real and fake images.

From the literature survey, we can state that stacking based ensemble methods provide a robust framework that effectively combines multiple models to enhance predictive accuracy, dynamically fuse predictions, and investigate feature interrelationships in a flexible and adaptable way [41,9,1,36]. Several domains have successfully utilized stack-based ensembles and achieved notable performance improvements. A random forest classifier-based stacking technique to integrate decision tree base learners was used for cancer detection by Wang et al. [41] and an XGBoost stacking based ensemble learning method was used for image classification by Aboneh et al. [1]. A prediction method based on stacking ensemble learning was used for earthquake casualty prediction by Cui et al. [9] whereas Mienye et al. [28] presented a concise overview of ensemble learning covering three main ensemble methods: bagging, boosting, and stacking.

Feature selection is an important concept used in the domain of machine learning [27,10,37]. Many research articles found in the literature have followed filter based feature selection. Sen et al. [39] presented a bi-stage feature selection approach in which Mutual Information (MI) and Relief-F were used in the first stage and the Dragonfly algorithm (DA) was used in the second stage. Guha et al. [18] used a score-based filter feature selection approach in which two popular statistical dependence measures, namely MI and Pearson Correlation Coefficient (PCC) were combined. The authors in [21,14,4] provided an extensive study on popularly used filter ranking methods. Ghosh et al. [15] proposed a 2-stage model for feature selection. In the first stage, an ensemble of filter methods was developed by considering the union and intersection of the top-n features of ReliefF, Chi-square, and Symmetrical Uncertainty. In the next stage, the Genetic Algorithm (GA) was used on the union and intersection to get fine-tuned results, and the union performed

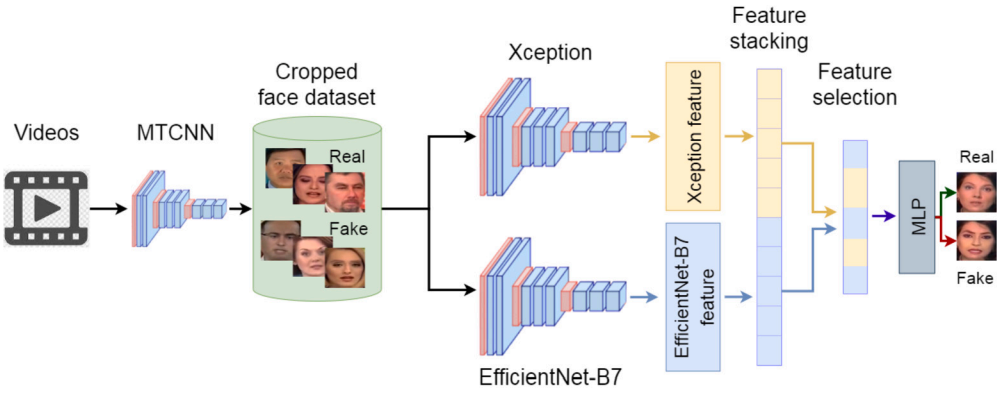


Fig. 2. Overall workflow diagram of the proposed deepfake detection method.

better than the latter. From this discussion, it is understood that feature selection plays a vital role in improving the performance of a learning model, while the dimension of the feature vector used is reduced significantly.

Keeping the above facts in mind, in the present method, we have applied a stacking base ensemble approach, where features from two deep learning models are stacked, followed by a feature selection method for better detection of deepfakes. Our experiments ensure that modern deepfakes are more difficult to detect than older ones. Our approach to encountering this problem is to stack the features of two CNNs and feed only those features to a meta-learner based on their importance in classifying the frames of the videos.

3. Motivation

Deepfake technology can be misused to create explicit or compromising videos, leading to privacy violations and harassment. Deepfake videos have the potential to spread false information, manipulate public opinion, and deceive individuals. It can be used for malicious purposes, such as creating fake news stories, political propaganda, or defaming individuals. With the rise of deepfake videos, there is a growing concern about the trustworthiness of visual media.

As a result, detecting deepfake videos helps us combat the spread of misinformation and promote the dissemination of accurate information. By developing effective detection methods, individuals can better protect their privacy and prevent their personal information from being exploited or manipulated without their consent. Detecting and exposing deepfake videos also helps maintain the integrity and authenticity of visual content. Researchers mainly concentrated on finding subtle artifacts without putting much emphasis on computational cost. In our method, we have emphasized both factors by filtering the essential features from the combined CNN outputs and truncating the inconsistent feature elements. Following the method, the proposed architecture achieves better results compared to many state-of-the-art methods, as reported in subsection 5.3.4.

4. Methodology

In this work, we have proposed a deepfake detection technique based on the selection of near optimal features obtained from deep features. Before that, deep features are extracted from two CNN architectures, followed by the stacking of those feature vectors. An overview of the proposed method is given in Fig. 2.

4.1. Data analysis and pre-processing

Before experimenting with the datasets, we have employed some pre-processing on the raw data. The methods for preparing the datasets are as follows:

- The entire video sequence is not considered. Rather, some consecutive frames with faces are extracted using Multi-Task Cascaded CNN (MTCNN), a popular face detection model.
- The video dataset contains both real and synthetic videos. Frames were extracted and saved in a folder with a proper naming convention. The naming of fake images started with fake and the real ones started with real.
- The cropped faces were then resized to 224×224 for convenient use.

4.2. Base models

In this section, we have described the two base CNN models that are used for feature extraction from the input data.

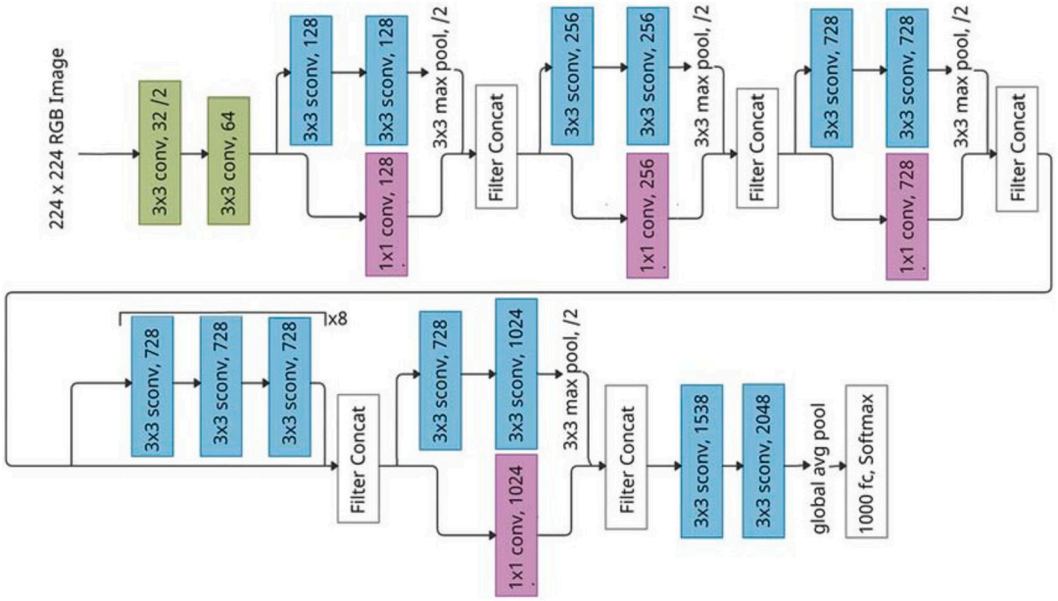


Fig. 3. Architecture of the Xception model [40].

4.2.1. *Xception*

Extreme Inception (Xception) is a deep CNN model introduced by Google researchers in 2016. The model is based on the Inception architecture and was designed to improve its efficiency and accuracy by replacing the traditional Inception module with a novel module called a depthwise separable convolution. The depthwise separable convolution in Xception consists of two operations: a depthwise convolution, which applies a single filter to each input channel separately, and a pointwise convolution, which applies a 1×1 convolution to combine the outputs of the depthwise convolution. This approach reduces the computational complexity of the network while maintaining accuracy.

In addition to the depthwise separable convolution, Xception also uses a modified residual module called “entry flow” and “exit flow” to improve the accuracy of the network. The entry flow consists of a series of convolutional layers that increase the number of feature maps, while the exit flow consists of a global average pooling layer followed by a softmax layer to produce the final classification output. Xception has achieved state-of-the-art performance on several image classification benchmarks, including the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and the CIFAR-10 dataset. The model has also been shown to be effective in transfer learning, where the pre-trained network is fine-tuned on a different dataset for a specific task. In summary, Xception is a deep CNN model based on the Inception architecture that uses a depthwise separable convolution and modified residual modules to improve its efficiency and accuracy. The model has achieved state-of-the-art performance on several image classification benchmarks and is effective in transfer learning. Xception architecture is shown in Fig. 3.

4.2.2. *EfficientNet-B7*

EfficientNet-B7 is a CNN model that is part of the EfficientNet family of models introduced by Google researchers in 2019. It is the largest and most complex model in the EfficientNet series, with over 66 million parameters. EfficientNet-B7 is designed using a combination of techniques, including compound scaling, improved building blocks, and neural architecture search, to achieve high accuracy and efficiency. The compound scaling technique involves scaling the depth, width, and resolution of the network in a balanced way to achieve optimal performance. This technique allows EfficientNet-B7 to achieve high accuracy while using fewer computational resources than other state-of-the-art models.

The model uses a series of convolutional layers and residual modules to extract hierarchical features from the input image. The residual modules use a modified version of the Inception module called the “MBConv block”, which consists of depthwise separable convolutions and a squeeze-and-excitation layer to improve the efficiency and accuracy of the network. EfficientNet-B7 has achieved state-of-the-art performance on several image classification benchmarks, including the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and the Common Objects in Context (COCO) dataset. The model is also suitable for transfer learning, where the pre-trained network is fine-tuned on a different dataset for a specific task. The EfficientNet-B7 architecture is shown in Fig. 4.

4.3. *Feature stacking*

An ensemble approach is a powerful technique used in machine learning to improve the predictive accuracy of a system of models. It involves combining the predictions of multiple base models to make a more accurate prediction as used in [31,12,29]. The ensemble is based on the principle that a diverse set of models will make different errors, and by combining their predictions, the overall error will be reduced. To follow this logic, for each base model, predictions have been made for each sample, and the output

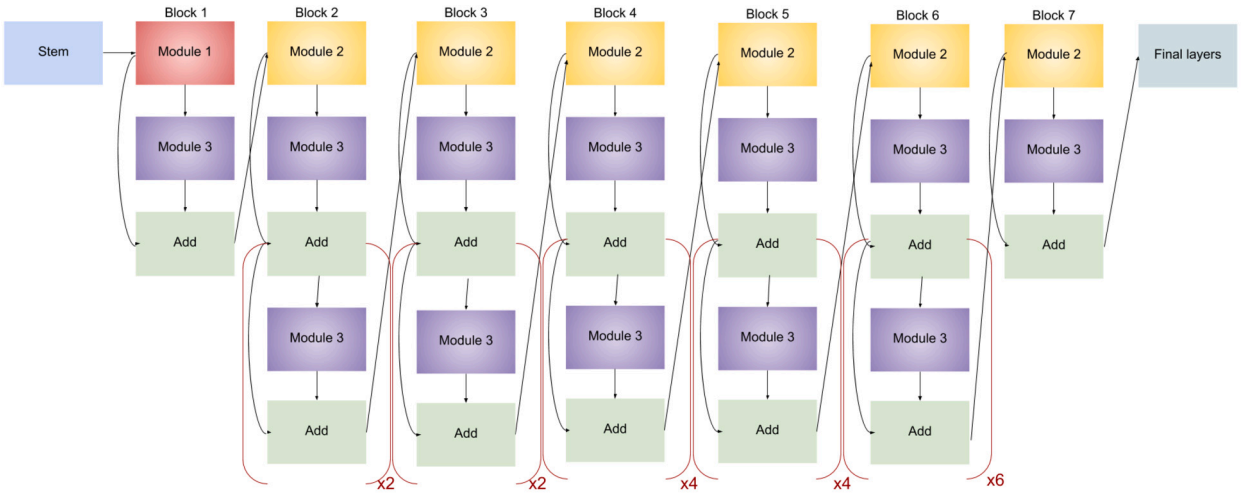


Fig. 4. Architecture of the EfficientNet-B7 model [3].

of the final layer, excluding the dense layers, is extracted as the predicted feature vector for each sample. Then, the stacking of those feature vectors has been done.

Therefore, for each sample, Xception predicts a feature vector of length 2048, and EfficientNet-B7 predicts a feature vector of length 2560. Hence, for each sample, after stacking feature vectors obtained from the base models, we get a final feature vector of length 4608. After repeating this with all samples, we fit the final feature vector to a meta-learning based model for training after completing the feature selection process. We have applied it to models that have low predictive accuracy on their own but can be combined to produce a more accurate result.

4.4. Feature selection

Feature selection based on feature importance is a technique that leverages the intrinsic feature ranking capabilities of certain models to identify the most relevant features for a given task. Some machine learning models provide a measure of feature importance, which indicates how much each feature contributes to the model's predictive performance. Considering this fact, we have used the XG Boost Regressor and Random Forest classifier to assign feature weights for each feature vector. XG Boost determines feature importance by assessing the extent to which each feature enhances the model's performance. It accomplishes this by first dividing the data into two sets: one set includes the feature under consideration, while the other does not. Subsequently, the model is trained on each of these subsets, and the disparity in accuracy between them is computed. The feature that exhibits the most substantial disparity in accuracy is designated as the most pivotal feature. Random forest computes feature importance through the utilization of out-of-bag (OOB) error estimation. Initially, a model is constructed, and its OOB error is determined. Subsequently, a feature is systematically permuted, and the OOB error is recalculated. The measure of feature importance is then ascertained by quantifying the reduction in node impurity, taking into account the probability of reaching that particular node. This probability is established by dividing the number of samples that reach the node by the overall number of samples. A higher value signifies the feature's greater significance. Then we have created a final feature importance vector by taking the average of the values from the two feature importance vectors. After that, we have used weight sorting to sort the feature vector as per decreasing feature importance. We have then decided to pick the top 50% (as per experiments done in Section 5.3.1) to select the best-performing features, which are then fed to a meta-learning model to get the final results. It is to be mentioned that feature selection helps reduce the dimensionality of the dataset, leading to less computationally expensive and potentially more efficient models. The feature selection process is shown in Fig. 5.

Our feature selection process is given in Algorithm 1. This algorithm takes the input feature matrix X, the target variable y, and two machine learning models, *model1* and *model2*, as input. It trains both models on the input data and calculates their respective feature importance. Then, it computes the average feature importance by averaging the values obtained from both models. The algorithm sorts the average feature importances in descending order and selects the top k features with the highest importance. Finally, it returns the indices or names of the selected features.

4.5. Meta-learning model

Once the base models are ready, we then use a meta-learning model. So, after stacking the predictions of the individual base models, we get a feature vector list, and after selecting the best features, we get a new feature vector list of half the size as the new training data for the meta model. Now we fit the new training data into our meta model. After that, we repeat the same steps to get the new testing data to evaluate our model.

Algorithm 1 Process of feature selection.

Input:
X: Input feature matrix of shape $(n_{\text{samples}}, n_{features})$
y: Target variable of shape $(n_{samples})$
$M1$: First machine learning model (XG Boost Regressor)
$M2$: Second machine learning model (Random Forest)
Output: Sorted k% features based on their importance.
Procedure:

Step 1: Train $M1$ on the input data X and target variable y
Step 2: Train $M2$ on the same input data X and target variable y
Step 3: Get the feature importance scores from both models
Step 4: F_importance1 = $M1.F\_importance$
Step 5: F_importance2 = $M2.F\_importance$
Step 6: Calculate the average feature importance
Step 7: $average\_F\_importance = \frac{F\_importance1 + F\_importance2}{2}$
Step 8: Sort the average feature importance scores in descending order
Step 9: Select top-k% features based on average importance scores

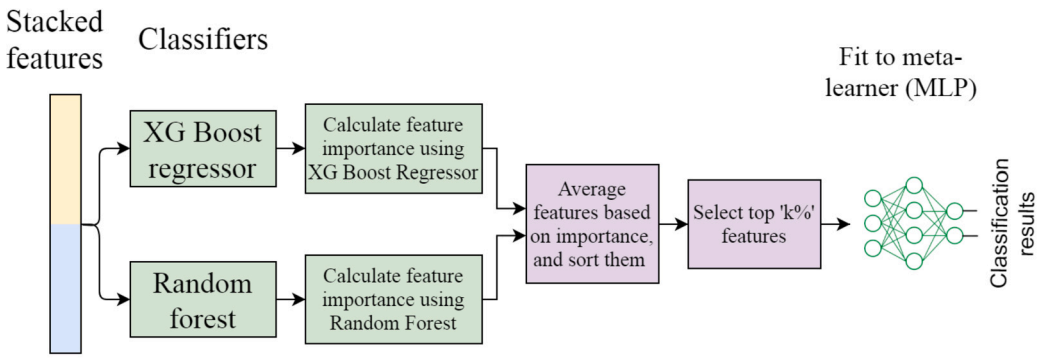


Fig. 5. Feature selection process used in the present work.

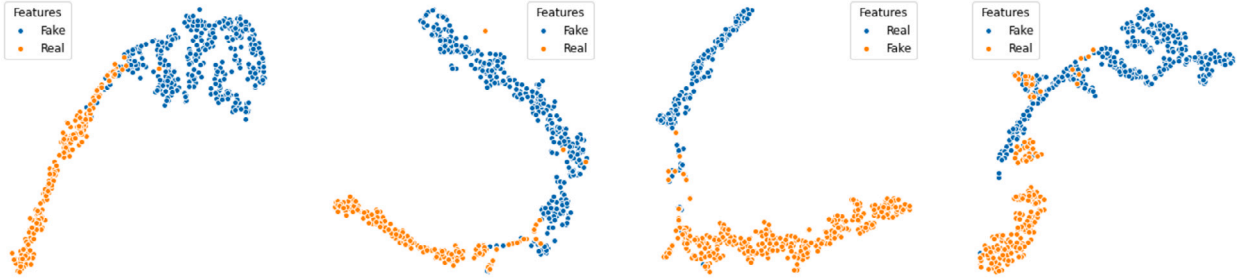


Fig. 6. Views of two-dimensional projections of feature maps with different numbers of layers in the meta-learning model. As the number of layers increases, the model exhibits an improved capacity to discern distinctive features that differentiate between different classes. However, when the number of layers is 5, the confusion between the classes "fake" and "real" gets saturated.

Meta-learning, often termed "learning to learn", is a captivating facet of machine learning. It focuses on empowering models to efficiently adapt and learn new tasks, rather than training them for specific tasks. This involves instilling models with the ability to generalize knowledge from past tasks and apply it to novel challenges. Multilayer Perceptrons (MLPs), with their capacity to capture intricate data relationships, are instrumental in this process. MLPs are seamlessly incorporated into the architecture of meta-learners, forming a critical component that enables models to easily adapt and develop new competencies, representing a significant stride forward in the realm of machine learning. Hence, in the case of a meta-learning model, we have decided to use a simple MLP based classifier. At its core, our MLP consists of an input layer, five hidden layers, and an output layer. The input layer is of size 2304, and the five hidden layers consist of 2048, 1024, 512, 256, and 128 neurons, respectively. The final output layer consists of two neurons. The selection of these layers has been determined through experimental analysis. We have generated t-distributed stochastic neighbor embedding (t-SNE) plots of the final feature maps obtained from the preceding layer of our meta-learning model. Fig. 6a-6d illustrates that the addition of the fifth layer significantly enhances the separation of the feature maps in comparison to the addition of the third or fourth layer. Beyond the fifth layer, as more layers are introduced, the feature maps begin to exhibit overlap.

Table 1
Class distribution of the two datasets used here for experiments. Here "Re" and "Fa" represent Real and Fake respectively.

	#Video						#image					
	Train		Validation		Test		Train		Validation		Test	
Dataset	Re	Fa	Re	Fa	Re	Fa	Re	Fa	Re	Fa	Re	Fa
Celeb-DF	612	4399	100	900	178	340	1130	8022	100	900	178	340
FF++	700	700	200	200	100	100	2930	2946	200	200	100	100

The model uses the cross-entropy as the loss function, the ReLU activation function for all the hidden layers, and the Softmax activation function for the final layer. The model has been trained for 50 epochs, and the batch size was 32. During the training phase for both datasets, training loss, training accuracy, validation loss, and validation accuracy have been measured, and whenever a new best validation accuracy is achieved, the model is saved with its current weight values so that, later, it is considered the best performing version of every model that will be used as the meta-learning model.

The primary strength of an MLP as a meta-learning model lies in its ability to learn nonlinear relationships and capture intricate patterns within the face images. By stacking multiple hidden layers, an MLP can create complex representations of the input faces, enabling it to identify a wide range of complex facial features.

5. Experimental results and analysis

The following subsections provide details of our experiment and its outcomes.

5.1. Dataset description

For doing the experimentation and evaluating our proposed technique, the Celebrity video dataset (Celeb-DF) [25] and FaceForensics++ [35] dataset have been selected here. We have used MTCNN to detect faces in video frames and then cropped them. From each video, 3 or 4 samples have been collected based on their length, and the datasets have been prepared.

- **Celeb-DF** is a large scale challenging dataset for deepfake forensics. It includes 590 original videos collected from YouTube with subjects of different ages, ethnic groups, and genders, and 5639 corresponding DeepFake videos. Frames from different videos are extracted, and the dataset is created. The images exclusively contain the faces of different celebrities from all over the world. After the extraction of frames, the training set contains 1130 real images and 8022 fake images; the validation set contains 100 real images and 900 fake images; and the test set contains 178 real images and 340 fake images.
- **FaceForensics**++ (FF++) is a forensics dataset consisting of 1000 original video sequences that have been manipulated with four automated face manipulation methods: Deepfake, Face2Face, FaceSwap, and NeuralTextures. The data has been sourced from 977 YouTube videos, and all videos contain a trackable, mostly frontal face without occlusions, enabling automated tampering methods to generate realistic forgeries. In our experiment, we have considered deepfake and the original set. Frames from different videos are extracted, and the dataset is created. After the extraction of frames, the training set contains 2930 real images and 2946 fake images; the validation set includes 198 real images and 197 fake images; and the test set contains 100 real images and 100 fake images.

Table 1 shows the detailed distribution of video and image levels used in our research.

5.2. Evaluation metrics

Let, TP be the number of fake images that are correctly detected as fake images; TN be the number of real images that are correctly detected as real images; FP be the number of real images that are erroneously detected as fake images; and FN be the number of fake images that are detected as real images.

Accuracy refers to the fraction of predictions made by a model that is correct and measured using Equation (1).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

Precision (Pr) refers to the ability to accurately predict the outcome of a given event or process. It is typically measured as the percentage of correct predictions made by a model or an algorithm. It is calculated as shown in Equation (2).

$$Pr = \frac{(TP)}{(TP + FP)} \tag{2}$$

Recall (Re) in machine learning refers to the ability of a model to correctly identify all instances of a particular class in a dataset. It is typically measured as a ratio of the number of TP predictions (correctly identified instances of the class) to the total number of instances of the class in the dataset. It is calculated as shown in Equation (3).

Table 2
Performance of Xception and EfficientNet-B7 models on the test set of Celeb-DF and FF++ datasets. All values are in %.

	Celeb-DF		FF++	
Unit	Xception	EfficientNet-B7	Xception	EfficientNet-B7
Accuracy	94.20	93.44	95.50	97.00
Precision	93.30	91.13	94.23	97.98
Recall	98.24	99.70	97.03	96.04
F1 score	95.70	95.22	95.61	97.00
AUC	98.16	98.25	99.14	99.61

$$Re = \frac{(TP)}{(TP + FN)} \tag{3}$$

The **F1 score** is a metric used to evaluate the performance of a machine learning model. It is calculated by taking the harmonic mean of precision and recall. A higher F1 score indicates a better performing model, with a score of 1 being the optimal performance. It is calculated as shown in Equation (4).

$$F1\ Score = \frac{(2 * Pr * Re)}{(Pr + Re)} \tag{4}$$

Area Under the Curve (AUC) is a metric used to evaluate the performance of a binary classification model. It measures the ability of the model to distinguish between positive and negative samples across all possible threshold values.

5.2.1. Training of CNN models

Two models built on different CNN architectures and with varying configurations are trained individually on the training dataset. In addition to the base model's configurations and values for training settings, we have replaced the standard CNN architecture's fully connected layers with two dense layers, one with 512 neurons and the final layer with 2 neurons. Both models used the cross-entropy as a loss function, the rectified linear unit (ReLU) activation function for the first dense layer, and the SoftMax activation function for the final dense layer.

Celeb-DF dataset: Due to the large size of the training dataset, we could not fit all the training samples at once due to limited GPU RAM. Each of the base models is trained for 50 epochs, and for each epoch, 40 batches of random training samples have been used for training. The batch size used is 32. So, a total of 40* 32 = 1280 random samples for each epoch have been used for training. All the two base models have been trained on the same training set, and validated on the same validation set.

FaceForensics++ dataset: In the case of the FaceForensics++ dataset, each of the foundational models underwent training for 50 epochs. Throughout each epoch, the training involved utilizing 40 batches of randomly selected training samples. The batch size employed was 16, resulting in a cumulative usage of 640 random samples for each epoch. Both base models have been trained on identical training data and evaluated using the same validation set for each of the two datasets.

During the training phase for both datasets, training loss, training accuracy, validation loss, and validation accuracy have been measured, and whenever a new best validation accuracy is achieved, the model is saved with its current weight values so that, later, it is considered as the best performing version of every model that we will use as ensemble members.

5.3. Results and analysis

5.3.1. Experiments with varying k in feature selection

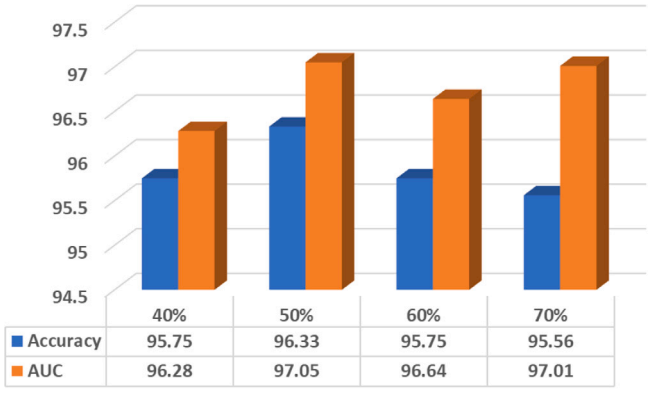
We have done experiments to select the right percentage of features while doing feature selection. We have checked the accuracy and AUC values while selecting the top 40%, top 50%, top 60%, and top 70% features while doing feature selection on Celeb-Df dataset and the FF++ dataset. The results are given in Fig. 7a-7b. For the celeb-df dataset, we can observe that the highest accuracy and highest AUC are achieved in the case of selecting the top 50% features. For the FF++ dataset, we can observe that we got the highest accuracy and AUC at top 50% feature selection, and after that, accuracy got saturated. Therefore, after careful evaluation of the results from this experiment, we decided to go ahead with selecting the top 50% features in our proposed method.

5.3.2. Base models

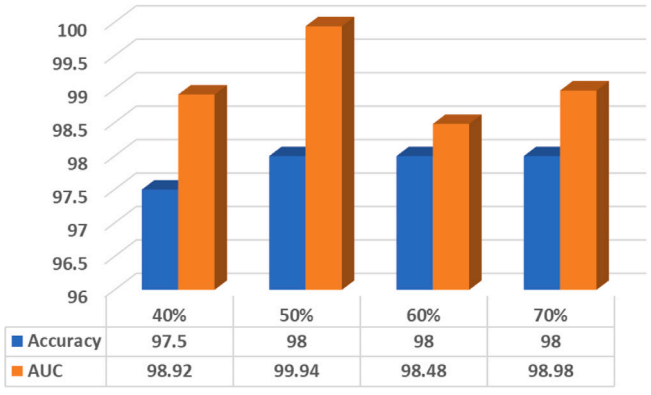
Each base model is individually trained on the training set, i.e., the weights of the base-learners are updated during the training since their trainable property is set to true (i.e., trainable) while defining them. Moreover, the accuracy, precision, recall, F1 score, and AUC scores for the two base models (i.e., Xception and EfficientNet-B7) on the two datasets used are given in Table 2.

5.3.3. Meta-learning based model

The meta-learner combines the predictions of the base models and is trained on the predictions made by the base models. Accuracy, Precision, Recall, F1 score, and AUC values of the meta-learner on the two datasets are given in Table 3. This also makes it evident that the combined feature selection model, achieved through averaging, outperforms each of the individual feature selection models. Also, the confusion matrices of the meta-learner on the two datasets with intra-dataset experimental setups are given in Fig. 8a-8b.

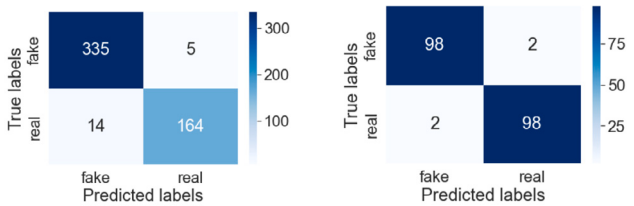


(a)



(b)

Fig. 7. Performance comparison with varying K ranging from 40% to 70% with step size 10 during feature selection on (a) Celeb-DF dataset and (b) FF++ dataset.



(a) (b)

Fig. 8. Confusion matrices of the intra-dataset experimental setup, i.e., our model is trained, validated, and tested on (a) Celeb-DF dataset and (b) FF++ dataset.

Table 3
Performance of the individual feature selection method with the combined (average/proposed) method on intra-dataset experimental setups for Celeb-DF and FF++ datasets. All values are in %.

Dataset	Method	Accuracy	Precision	Recall	F1 score	AUC
FF++	XG Boost	97.00	97.00	97.00	97.00	98.97
	Random forest	97.50	97.50	97.50	97.50	97.99
	Proposed	98.00	98.00	98.00	98.00	99.94
Celeb-DF	XG Boost	95.56	96.27	93.94	95.09	96.82
	Random forest	95.76	96.07	94.49	95.27	96.95
	Proposed	96.33	95.99	98.53	97.24	97.05

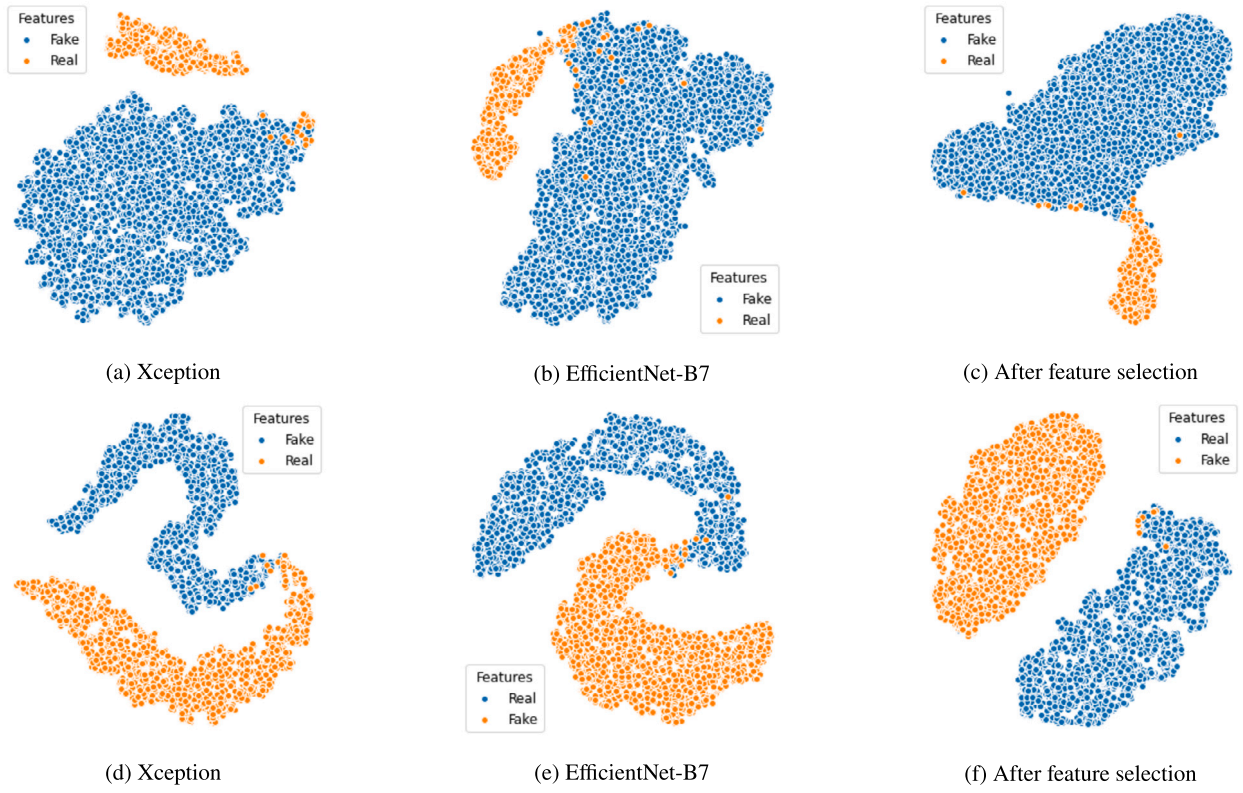


Fig. 9. Visualization of features using t-SNE tool obtained from Xception, EfficientNet-B7, and after selection of specific features. The first and second rows represent Celeb-DF and FF++ datasets, respectively.

To analyze the selected optimal feature set compared to the feature set provided by two base models, we have used the t-SNE data visualization tool. This tool provides insights and reveals patterns, clusters, or relationships within the features by converting them into 2D space that may not be easily discernible in the original high-dimensional space. Fig. 9a-9f is used to visualize the extracted features from individual base models after the selection of optimal features for both datasets. This also relates to the fact that some of the misclassified features are properly aligned after the selection of optimal features that Xception and EfficientNet-B7 could not do individually.

It is observed that our proposed technique increases the overall accuracy. We can observe around 2.13% on the Celeb-DF dataset, and 1% on the FF++ dataset increase in overall accuracy as compared to the corresponding best base models.

The probable reasons behind such improvements are as follows:

- The stacking of features allows for the use of a diverse range of models, each with its strengths and weaknesses. By combining the feature vectors of these models, stacking can take advantage of the strengths of each model and produce more accurate predictions overall. Also, using multiple models and stacking their predictions can help reduce the impact of overfitting and improve the overall accuracy of the prediction.
- Feature selection reduces the dimensionality of the feature set by selecting a subset of features. This reduction in the number of features leads to faster training and inference times as the model has fewer computations to perform. Also, irrelevant or inconsistent features can introduce noise or bias into the learning algorithm, leading to suboptimal model performance. Feature selection helps to eliminate these irrelevant or inconsistent features, allowing the model to focus on the most discriminative and informative features, which in turn can improve the model's ability to capture the underlying patterns in the data and make more accurate predictions.
- MLP can automatically learn and extract high-level representations from the input data through its hidden layers. It can discover relevant features or combinations of features that are most informative for the given task and can learn complex non-linear relationships within data. It consists of multiple layers of interconnected neurons, allowing it to capture intricate patterns. On the other hand, traditional machine learning based classifiers tend to have a simpler structure, may struggle with capturing complex patterns, and often require feature engineering, where domain knowledge and manual feature selection or extraction are needed to provide the most relevant input to the classifier.

Overall improvement in performance by using the meta-learning based model as compared to the best base model for the two datasets is given in Fig. 10a-10b.

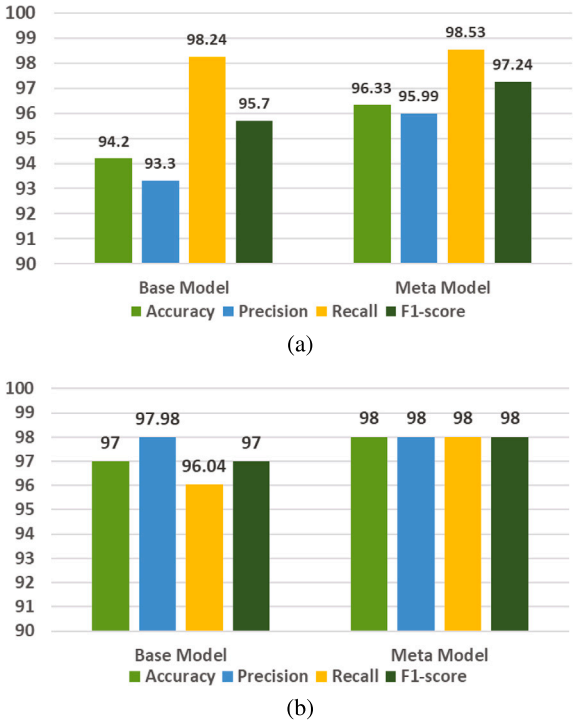


Fig. 10. Performance comparison of the best base model and meta-learner tested on (a) Celeb-DF dataset and (b) FF++ dataset.

Table 4
The test accuracy and AUC of the proposed model are compared with several state-of-the-art models that have been evaluated using our experimental configurations.

	Celeb-DF		FF++	
Method	Accuracy (%)	AUC (%)	Accuracy (%)	AUC (%)
Afchar et al. [2] (Meso-4)	65.64	65.33	67.00	66.81
Afchar et al. [2] (Meso-Inception-4)	65.83	64.80	65.00	66.93
Chollet et al. [6]	89.38	95.59	95.50	98.79
Coccomini et al. [7]	87.06	81.84	95.50	96.52
Ganguly et al. [13]	68.33	78.04	78.00	87.62
Guo et al. [19]	65.64	61.07	86.00	96.41
Mohiuddin et al. [29]	84.56	78.46	86.00	85.92
Qian et al. [34]	81.32	80.50	84.00	83.04
Wodajo et al. [42]	90.35	96.63	96.00	99.06
Ganguly et al. [12]	94.40	98.54	97.00	98.57
Proposed	**96.33**	**97.05**	**98.00**	**99.94**

5.3.4. Comparison with past methods

In this section, we compare the performance of the proposed model with some state-of-the-art deepfake detection methods proposed by Afchar et al. [2] (Meso-4), Afchar et al. [2] (MesoInception-4), Chollet et al. [6], Coccomini et al. [7], Ganguly et al. [13], Guo et al. [19], Mohiuddin et al. [29], Qian et al. [34], Wodajo et al. [42] and Ganguly et al. [12]. To ensure a fair comparison, we have assessed these methods according to our established experimental protocols, as previously described in 5.1. This approach enables a thorough evaluation of different methods based on their robustness and generalizability. In Table 4, we present the performance results, including test accuracy and AUC score, of these deepfake detection methods on both the Celeb-DF and FF++ image datasets. Notably, our proposed method demonstrates superior performance compared to all the state-of-the-art methods employed for comparison within the intra-dataset experimental setup. Furthermore, when evaluating the model's generalizability through inter-dataset experiments, our method consistently outperforms other approaches, achieving state-of-the-art outcomes in the majority of the experiments.

5.3.5. Robustness of the proposed method

To test the robustness of our proposed method, we have tested our model on inter-dataset experimental setups. In the first setup, we used the Celeb-DF dataset for training and validation purposes, and the FF++ dataset for testing purposes. Using this setup, we have achieved a test accuracy of 84.50%. In the second setup, we have used the FF++ dataset for both training and validation

Table 5
Performance comparisons of the meta-learning model on the inter-dataset experimental setup with state-of-the-art methods. Here, X Y in the first column indicates that our method is trained and validated on dataset X and evaluated on dataset Y. All values are in %. Here, Ex. Setup and Acc represent Experimental Setup and Accuracy, respectively.

Ex. Setup	Method	Acc	AUC
Celeb-DF FF++	Li et al. [25]	64.50	75.19
	Ganguly et al. [13]	65.00	63.80
	Mohiuddin et al. [29]	60.00	59.93
	Mohiuddin et al. [31]	66.50	76.72
	Proposed	84.50	91.88
FF++ Celeb-DF	Li et al. [25]	58.06	55.60
	Ganguly et al. [13]	68.04	66.12
	Mohiuddin et al. [29]	63.71	56.43
	Mohiuddin et al. [31]	79.34	87.58
	Proposed	65.83	70.57

Table 6
Performance of the meta-learner on brighter and darker versions of the test sets. All values are in %. Here, Acc, Pre, Rec, and F1 represent Accuracy, Precision, Recall, and F1 score respectively.

Dataset	Acc	Pre	Rec	F1	AUC
Celeb-DF Brighter	96.14	98.78	95.29	97.00	97.41
Celeb-DF Brighter	94.02	91.64	100.00	95.64	94.36
FF++ Brighter	99.00	100.00	98.00	98.99	99.92
FF++ Darker	97.00	95.19	99.00	97.06	98.64

purposes, and the Celeb-DF dataset for testing purposes. Using this setup, we have achieved a test accuracy of 65.83%. The reason behind the higher accuracy in the first setup and the lower accuracy in the second setup is that FF++ is a first generation dataset and the Celeb-DF dataset is a second generation dataset. Also, the GAN models used to generate the Celeb-DF dataset are superior to the ones used to generate the FF++ dataset. So the training of the model in the first setup has been far better as compared to the second setup, and hence the results. Table 5 displays the outcomes of the inter-dataset experiment for the proposed method and state-of-the-art (SOTA) methods.

To evaluate the robustness of our model, we have conducted additional tests using modified versions of the test sets with varying brightness levels. We have increased the brightness factor by 1.3 for both test sets and also decreased it by 0.7 to create darker versions of the Celeb-DF and FF++ test sets. These tests have been performed on the previously trained model and evaluated using unknown samples. The results of these tests reveal that our model achieves an accuracy of approximately 96.14% and 94.02% on the brighter and darker versions of the Celeb-DF test set. Similarly, for the brighter and darker versions of the FF++ dataset, our model achieves an accuracy of approximately 99.00% and 97.00% respectively.

It is worth noting that the brighter images produce slightly better results compared to the darker ones. This observation suggests that brighter images generally tend to have fewer deepfake artifacts due to several reasons. Firstly, bright images tend to have a higher signal-to-noise ratio, which means that the desired information (the actual content of the image) is more distinguishable from the unwanted noise or artifacts. Secondly, brighter images often contain more texture and detail, making it harder for deepfake algorithms to convincingly generate realistic-looking details in manipulated areas. This makes it easier for deepfake detection algorithms to identify and differentiate between manipulated and authentic regions. The results of these robustness tests are given in Table 6.

6. Strengths, limitations and future scope

This section discusses the strengths, limitations, and possible improvements of the proposed deepfake detection method.

6.1. Advantages

- Increased detection accuracy: Stacking based ensemble of two CNN models leverages their strengths, thereby improving accuracy in identifying deepfake videos.
- Robust feature selection: feature selection techniques help identify the most relevant attributes for differentiating between real and manipulated videos, reducing noise, and improving the overall detection performance.
- Reducing false positives and false negatives: By combining the outputs of multiple models, the ensemble helps minimize both false positives (misclassifying real videos as deepfakes) and false negatives (failing to identify actual deepfakes).
- Enhanced robustness: Ensemble models with different architectures and varied principles enhance the robustness of the deepfake detection system, making it more resilient to adversarial attacks and novel manipulation techniques.

- Versatility: A combination of ensemble and feature selection techniques is useful across different types of deepfake detection methods, making them adaptable and applicable to various detection approaches.

6.2. Limitations

- Increased computational complexity: When we form an ensemble of multiple models and perform feature selection, it can introduce higher computational requirements, which may limit real-time implementation or pose challenges for resource-constrained environments.
- Interpretability challenges: Ensemble based frameworks may reduce the interpretability of the overall detection system, making it difficult to understand the reasoning behind the decision-making process.
- Overfitting risk: If the ensemble is not properly designed or trained, there is a risk of overfitting the training data, which can negatively impact the generalization performance on unseen data.

6.3. Future scope

- Hybrid approaches: Exploring hybrid approaches that combine various techniques, such as machine learning, deep learning, and adversarial training, can further improve deepfake detection accuracy and interpretability.
- Adapting to evolving deepfake techniques: Continual research is required to keep up with the advancements in deepfake generation techniques, ensuring that methods can effectively detect new and sophisticated manipulations.
- Robustness against adversarial attacks: Developing techniques to enhance the robustness of ensemble-based detection models against adversarial attacks and evasion strategies is crucial for real-world deployment.
- Real-time implementation: Investigating optimization methods to reduce the computational complexity of ensemble and feature selection techniques can enable real-time deepfake detection in practical applications.
- Large-scale datasets: Collecting diverse and comprehensive datasets that include various deepfake variations, in addition to real-world videos, is essential for training robust and generalizable ensemble models.

7. Conclusion

In this paper, the use of stacking techniques and feature selection in deepfake detection has shown promising results in addressing the growing threat of manipulated media. By combining multiple detection models and selecting relevant features, the proposed approach has demonstrated improved accuracy and robustness in identifying deepfake videos. The stacking approach helps mitigate the limitations of individual models, enhancing the overall detection performance. Whereas, feature selection allows for the identification of discriminative attributes that effectively differentiate between real and manipulated videos. Furthermore, a robustness checking technique has been employed to assess the system's resilience against adversarial attacks and attempts to evade detection. By evaluating the system's performance under various challenging scenarios, including different types of deepfake techniques and perturbations, its effectiveness is ensured. While there are still challenges to overcome in this field, such as the emergence of more sophisticated deepfake techniques, the stacking, and feature selection methods provide a solid foundation for developing more advanced and reliable deepfake detection systems in the future.

CRediT authorship contribution statement

Gourab Naskar: Methodology, Investigation, Formal analysis. **Sk Mohiuddin:** Validation, Methodology, Investigation. **Samir Malakar:** Writing – original draft, Visualization, Investigation. **Erik Cuevas:** Methodology, Conceptualization. **Ram Sarkar:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Erik Cuevas holds an editor position at Heliyon If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and materials

Not applicable in our case.

References

[1] Tagel Aboneh, Abebe Rorissa, Ramasamy Srinivasagan, Stacking-based ensemble learning method for multi-spectral image classification, Technologies 10 (1) (2022) 17.
[2] Darius Afchar, Vincent Nozick, Junichi Yamagishi, Isao Echizen, Mesonet: a compact facial video forgery detection network, in: 2018 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2018, pp. 1–7.
[3] Vardan Agarwal, Complete Architectural Details of all EfficientNet Models.

[4] Andrea Bommert, Xudong Sun, Bernd Bischl, Jörg Rahnenführer, Michel Lang, Benchmark for filter methods for feature selection in high-dimensional classification data, Comput. Stat. Data Anal. 143 (2020) 106839.
[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul 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, 2018, pp. 8789–8797.
[6] François Chollet, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.
[7] Davide Alessandro Coccomini, Nicola Messina, Claudio Gennaro, Fabrizio Falchi, Combining EfficientNet and vision transformers for video deepfake detection, in: International Conference on Image Analysis and Processing, Springer, 2022, pp. 219–229.
[8] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, Anil A. Bharath, Generative adversarial networks: an overview, IEEE Signal Process. Mag. 35 (1) (2018) 53–65.
[9] Shaoze Cui, Yunqiang Yin, Dujuan Wang, Zhiwu Li, Yanzhang Wang, A stacking-based ensemble learning method for earthquake casualty prediction, Appl. Soft Comput. 101 (2021) 107038.
[10] Susovan Das, Akash Chatterjee, Samiran Dey, Shilpa Saha, Samir Malakar, Breast cancer detection from histology images using deep feature selection, in: Proceedings of International Conference on Frontiers in Computing and Systems: COMSYS 2021, Springer, 2022, pp. 323–330.
[11] Ricard Durall, Margret Keuper, Franz-Josef Pfreundt, Janis Keuper, Unmasking deepfakes with simple features, arXiv preprint arXiv:1911.00686, 2019.
[12] Shreyan Ganguly, Aditya Ganguly, Sk Mohiuddin, Samir Malakar, Ram Sarkar, ViXNet: vision transformer with xception network for deepfakes based video and image forgery detection, Expert Syst. Appl. 210 (2022) 118423.
[13] Shreyan Ganguly, Sk Mohiuddin, Samir Malakar, Erik Cuevas, Ram Sarkar, Visual attention-based deepfake video forgery detection, Pattern Anal. Appl. (2022) 1–12.
[14] Kushal Kanti Ghosh, Shemim Begum, Aritra Sardar, Sukdev Adhikary, Manosij Ghosh, Munish Kumar, Ram Sarkar, Theoretical and empirical analysis of filter ranking methods: experimental study on benchmark DNA microarray data, Expert Syst. Appl. 169 (2021) 114485.
[15] Manosij Ghosh, Sukdev Adhikary, Kushal Kanti Ghosh, Aritra Sardar, Shemim Begum, Ram Sarkar, Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods, Med. Biol. Eng. Comput. 57 (2019) 159–176.
[16] Luca Guarnera, Oliver Giudice, Sebastiano Battiato, Deepfake detection by analyzing convolutional traces, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 666–667.
[17] David Güera, Edward J. Delp, Deepfake video detection using recurrent neural networks, in: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2018, pp. 1–6.
[18] Ritam Guha, Kushal Kanti Ghosh, Showmik Bhowmik, Ram Sarkar, Mutually informed correlation coefficient (MICC)-a new filter based feature selection method, in: 2020 IEEE Calcutta Conference (CALCON), IEEE, 2020, pp. 54–58.
[19] Zhiqing Guo, Gaobo Yang, Jiyou Chen, Xingming Sun, Fake face detection via adaptive manipulation traces extraction network, Comput. Vis. Image Underst. 204 (2021) 103170.
[20] Z. He, W. Zuo, M. Kan, S. Shan, X. Chen, AttGAN: facial attribute editing by only changing what you want, IEEE Trans. Image Process. 28 (11) (Nov 2019) 5464–5478.
[21] Sydney M. Kasongo, Yanxia Sun, Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset, J. Big Data 7 (2020) 1–20.
[22] Marissa Koopman, Andrea Macarulla Rodriguez, Zeno Geradts, Detection of deepfake video manipulation, in: The 20th Irish Machine Vision and Image Processing Conference (IMVIP), 2018, pp. 133–136.
[23] Haodong Li, Bin Li, Shunquan Tan, Jiwu Huang, Identification of deep network generated images using disparities in color components, Signal Process. 174 (2020) 107616.
[24] Yuezun Li, Siwei Lyu, Exposing deepfake videos by detecting face warping artifacts, arXiv preprint arXiv:1811.00656, 2018.
[25] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, Siwei Lyu, Celeb-df: a large-scale challenging dataset for deepfake forensics, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3207–3216.
[26] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, Shilei Wen, STGAN: a unified selective transfer network for arbitrary image attribute editing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3673–3682.
[27] Samir Malakar, Swaraj Sen, Sergei Romanov, Dmitrii Kaplun, Ram Sarkar, Role of transfer functions in PSO to select diagnostic attributes for chronic disease prediction: an experimental study, J. King Saud Univ, Comput. Inf. Sci. 35 (9) (2023) 101757.
[28] Ibomoiye Domor Mienye, Yanxia Sun, A survey of ensemble learning: concepts, algorithms, applications, and prospects, IEEE Access 10 (2022) 99129–99149.
[29] Sk Mohiuddin, Shreyan Ganguly, Samir Malakar, Dmitrii Kaplun, Ram Sarkar, A feature fusion based deep learning model for deepfake video detection, in: International Conference on Mathematics and Its Applications in New Computer Systems, Springer, 2022, pp. 197–206.
[30] Sk Mohiuddin, Samir Malakar, Munish Kumar, Ram Sarkar, A comprehensive survey on state-of-the-art video forgery detection techniques, Multimed. Tools Appl. (2023) 1–41.
[31] Sk Mohiuddin, Khalid Hassan Sheikh, Samir Malakar, Juan D. Velásquez, Ram Sarkar, A hierarchical feature selection strategy for deepfake video detection, Neural Comput. Appl. 35 (13) (2023) 9363–9380.
[32] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, Jose M. Álvarez, Invertible conditional GANs for image editing, in: NIPS Workshop on Adversarial Training, 2016, pp. 1–9.
[33] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr Dpfks, Carl Shift Facenheim, R.P. Luis, Jian Jiang, et al., DeepFaceLab: integrated, flexible and extensible face-swapping framework, arXiv preprint arXiv:2005.05535, 2020.
[34] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, Jing Shao, Thinking in frequency: face forgery detection by mining frequency-aware clues, in: European Conference on Computer Vision, Springer, 2020, pp. 86–103.
[35] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, Matthias Nießner, FaceForensics++: learning to detect manipulated facial images, in: International Conference on Computer Vision (ICCV), 2019, pp. 1–11.
[36] Soham Roy, Archan Bhattacharya, Navonil Sarkar, Samir Malakar, Ram Sarkar, Offline hand-drawn circuit component recognition using texture and shape-based features, Multimed. Tools Appl. 79 (2020) 31353–31373.
[37] Suryadipto Sarkar, Manosij Ghosh, Agneet Chatterjee, Samir Malakar, Ram Sarkar, An advanced particle swarm optimization based feature selection method for tri-script handwritten digit recognition, in: Computational Intelligence, Communications, and Business Analytics: Second International Conference, CICBA 2018, Kalyani, India, July 27–28, 2018, Revised Selected Papers, Part I 2, Springer, 2019, pp. 82–94.
[38] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-cam: visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.
[39] Shibaprasad Sen, Soumyajit Saha, Somnath Chatterjee, Seyedali Mirjalili, Ram Sarkar, A bi-stage feature selection approach for Covid-19 prediction using chest ct images, Appl. Intell. 51 (2021) 8985–9000.
[40] Kathiravan Srinivasan, Lalit Garg, Debajit Datta, Abdulellah Alaboudi, Noor Jhanjhi, Rishav Agarwal, Anmol Thomas, Performance comparison of deep CNN models for detecting driver's distraction, Comput. Mater. Continua 68 (05 2021) 4109–4124.
[41] Yuyan Wang, Dujuan Wang, Na Geng, Yanzhang Wang, Yunqiang Yin, Yaochu Jin, Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection, Appl. Soft Comput. 77 (2019) 188–204.

[42] Wodajo Deressa, Solomon Atnafu, Deepfake video detection using convolutional vision transformer, arXiv preprint arXiv:2102.11126, 2021.
[43] Yuanlu Wu, Yan Wo, Caiyu Li, Guoqiang Han, Learning domain-invariant representation for generalizing face forgery detection, Comput. Secur. 130 (2023) 103280.
[44] Xin Yang, Yuezun Li, Siwei Lyu, Exposing deep fakes using inconsistent head poses, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 8261–8265.
[45] Peng Zhou, Xintong Han, Vlad I. Morariu, Larry S. Davis, Two-stream neural networks for tampered face detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2017, pp. 1831–1839.