

3) *Bimodal information-based video features*: The deep extracted features from visual video frames and audio modalities using the above-mentioned unimodality-based feature extraction methods are mid-fused at a concatenate layer. This produces a feature vector representation for the whole video, which is based on bimodal information.

4) *Temporal information extraction-based attention mechanism*: Most deepfake videos are generated based on synthesizing faces frame-by-frame, cloning voices, and synchronizing lips. They suffer from flickering and discontinuity of the face frames and lack of normal emotions, breathing, pauses, and the pace at which the target subject speaks among audio segments. As a result, the GRU-based attention mechanism is applied to the three levels of the extracted features independently; visual video frames, audio, and the whole video. This aims to capture the instructive temporal information that helps to differentiate real videos from fake ones.

The GRU architecture is composed of two gates; update (upd) and reset (res), that modulate the information flow from the previous time step to the current step. At each time step t , the update gate decides the amount of previous information that should be retained, and the reset gate determines the amount of information that needs to be forgotten [53]. The GRU hidden state h at the time t is defined by the following formulae [54]:

$$\text{upd}_t = S(W_{\text{upd}}x_t + U_{\text{upd}}h_{t-1}) \quad (3)$$

$$\text{res}_t = S(W_{\text{res}}x_t + U_{\text{res}}h_{t-1}) \quad (4)$$

$$\hat{h}_t = \tanh(W_h x_t + \text{res}_t \circ U_h h_{t-1}) \quad (5)$$

$$h_t = (1 - \text{upd}_t) \circ \hat{h}_t + \text{upd}_t \circ h_{t-1} \quad (6)$$

where x refers to the input, and W and U represent the weight matrices. The symbol $S(\cdot)$ represents the sigmoid function, $\tanh(\cdot)$ represents the Hyperbolic Tangent, \circ denotes the Hadamard product, and \hat{h}_t denotes the candidate hidden state. As can be seen in Fig. 4, a single GRU is applied to the above-mentioned feature representations on the three levels. It produced a matrix of hidden state vectors at each time step t , which represents the learned temporal information per visual video, audio, or the whole video. The hidden state vector is defined as follows:

$$H = [h_1, h_2, \dots, h_t] \quad (7)$$

The attention mechanism uses the weights to concentrate on the important features from the input sequence H . It is defined by the following equations [17, 55]:

$$u_t = \tanh(W_h x_t + b) \quad (8)$$

$$\alpha_t = \text{softmax}(u_t) \quad (9)$$

$$c_t = \alpha_t h_t \quad (10)$$

$$v = \sum_t c_t \quad (11)$$

where u_t is a result of feeding a hidden vector h_t into a single-layer Multi-Layer Perceptron (MLP) with the tanh

activation function. W represents the weight matrix, and b refers to the bias term. The symbol α_t represents the normalized attention weights that are produced by applying the softmax layer to u_t . v is a video representation that is formed by summing hidden vectors h_t weighted by attention weights α_t .

C. Classification

After the instructive temporal features are produced from the GRU-based attention mechanism, a fully connected layer is used as an output layer with two classes. Softmax function is used to decide deepfake videos from real ones. The Softmax formula is defined as follows:

$$\text{Softmax}(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (12)$$

where y_i denotes the values resulting from the output layer neurons.

D. Dataset

The proposed method has been evaluated on the FakeAVCeleb multimodal videos dataset. This dataset consisted of 490 celebrity genuine videos that were selected from the VoxCeleb2 dataset based on various ethnic groups, gender, and age. Its genuine videos are face-centered and cropped. The fake videos of the FakeAVCeleb dataset were generated using DeepFaceLab, Faceswap, and FSGAN, while fake audios were generated using a real-time voice cloning tool (SV2TTS). Additionally, the Wav2Lip was applied to the deepfake videos to re-enact these videos based on the cloned audios. Thus, the FakeAVCeleb dataset had more realistic deepfakes. The FakeAVCeleb was divided into four groups; genuine visual videos with genuine audios, genuine visual videos with deepfake audios, deepfake visual videos with genuine audios, and deepfake visual videos with deepfake audios [4].

To evaluate the proposed method, 1215 genuine and deepfake videos of the FakeAVCeleb dataset are employed. These videos are divided into three subsets: training, validation, and testing.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed deepfake video detection method is evaluated by the FakeAVCeleb dataset. Its performance is assessed using the following evaluation metrics [56]:

$$\text{precision} = \frac{\text{True_Positives}}{\text{True_Positives} + \text{False_Positives}} \quad (13)$$

$$\text{sensitivity} = \text{recall} = \frac{\text{True_Positives}}{\text{True_Positives} + \text{False_Negatives}} \quad (14)$$

$$F_1 - \text{score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (15)$$

$$\text{accuracy} = \frac{\text{True_Positives} + \text{True_Negatives}}{\text{True_Positives} + \text{True_Negatives} + \text{False_Negatives} + \text{False_Positives}} \quad (16)$$

$$\text{specificity} = \frac{\text{True_Negatives}}{\text{True_Negatives} + \text{False_Positives}} \quad (17)$$

$$\text{AUROC} = \int_0^1 \text{sensitivity}((1 - \text{Specificity})^{-1}(x)) dx = p(x_2 > x_1) \quad (18)$$