International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Deepfake Detection Using Mamba (Selective State Space Model)

Chanakya Gattu[1], A P Ramya[2], P Sai Harsha[3], Likhith G[4], Dr. Sai Madhavi[5]

[1,2,3,4]CSE(AIML), Rao Bahadur Y. Mahabaleswarappa Engineering College
[5]HoD, CSE(AIML), Rao Bahadur Y. Mahabaleswarappa Engineering College

Abstract

The rapid proliferation of deepfake technology, driven by advanced generative models like GANs and Diffusion Models, has enabled the creation of hyper-realistic manipulated media that threatens information integrity, personal reputation, and national security. As these tools become more accessible, the need for automated, robust, and efficient detection systems is critical.

Existing detection methodologies largely rely on Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or Vision Transformers (ViTs). However, each approach faces significant limitations. CNN-based methods suffer from "temporal blindness," failing to detect inconsistencies across video frames. Hybrid CNN-RNN models address this but are hindered by slow training speeds and limited capacity for modeling long-range dependencies. While Transformer-based models excel at capturing global context, their quadratic computational complexity ($O(N^2)$) makes them computationally expensive and memory-intensive, rendering them inefficient for processing long, high-resolution video sequences. Furthermore, many current models struggle to generalize effectively to unseen datasets and novel manipulation techniques.

To overcome these challenges, this paper proposes a novel hybrid deepfake detection framework that integrates an EfficientNet backbone with the Mamba Selective State Space Model (SSM). This architecture leverages EfficientNet for robust spatial feature extraction and utilizes Mamba for temporal sequence modeling. Mamba introduces a content-aware selection mechanism and operates with linear-time complexity ($O(N^2)$), allowing it to efficiently capture subtle long-range temporal inconsistencies that traditional models miss, without the heavy computational cost of Transformers. Experimental results demonstrate that this proposed system offers a scalable, efficient solution with superior generalization capabilities, making it highly suitable for real-time deepfake detection in diverse and complex video environments.

Keywords: Deepfake Detection, GAN, Selective State Space Models (SSM), Mamba model, Vision Mamba (Vim), Temporal Sequence Modeling, EfficientNet Backbone, machine learning, deep learning, convolutional neural networks, transformers, attention mechanisms, benchmarking systems, datasets, Synthetic Media Detection, Linear-Time Complexity, Video Anomaly Detection, Computational Efficiency, Hybrid CNN-SSM, Facial Forgery Detection, Long-Range Dependency Modeling.

1. Introduction

Deepfake technology, once solely a tool for entertainment, has evolved into a mainstream method for facial forgery, posing serious threats to personal reputation, social stability, and national security. Advances in Generative Adversarial Networks (GANs) and Diffusion Models (DMs) now enable even non-technical users to produce hyper-realistic fake content, making detection a critical challenge. This has sparked an "arms race" where research in deepfake detection now surpasses generation, emphasising the urgent need for effective detection methods. While early approaches using Convolutional Neural Networks (CNNs) showed promise, they often fail to generalise to new forgery techniques and struggle with the compressed videos prevalent on social media.

This paper offers a comprehensive overview of the current landscape. We categorise detection methods, critically analyse benchmark datasets, and evaluate state-of-the-art models, noting a common weakness in cross-dataset generalisation. By identifying key limitations such as the quadratic computational complexity of transformers and their inefficiency for long video sequences we introduce the Mamba model as a novel solution. As a selective State Space Model (SSM), Mamba's linear-time complexity and ability to handle long-range dependencies present a promising and efficient alternative for the next generation of deepfake detection tools

2. A Comparative Analysis of Benchmark Datasets

The development of robust deepfake detection models is intrinsically linked to the datasets used for their training and evaluation. The characteristics of these datasets, such as their scale, diversity, and the quality of manipulations, define the challenges that models must overcome. This section provides an in-depth analysis of three seminal datasets in the field: FaceForensics++, the Deepfake Detection Challenge (DFDC) dataset, and Celeb-DF (v2).

2.1. FaceForensics++ (FF++)

FaceForensics++ is a pioneering large-scale dataset designed to provide a controlled environment for benchmarking face manipulation detection algorithms.

- **Creation and Composition:** The dataset originates from 1000 real videos downloaded from YouTube. These source videos were then used to generate 4000 manipulated videos, resulting in a total of 5000 videos. The dataset maintains a balanced gender distribution of 60% female and 40% male subjects. Manipulations were performed using five primary methods: DeepFakes, Face2Face, FaceSwap, FaceShifter, and NeuralTextures.
- **Key Features:**
  FaceForensics++ (FF++) is a benchmark dataset featuring controlled manipulations generated by specific forgery methods, making it ideal for artifact analysis. It simulates real-world conditions by offering videos in three quality levels (to test robustness against compression) and multiple resolutions (480p, 720p, 1080p).

2.2 Deepfake Detection Challenge (DFDC)

The DFDC dataset is currently the largest public facial forgery dataset, created for a global competition to accelerate the development of deepfake detection technologies capable of operating "in the wild".

- **Creation and Composition:** The dataset contains 119,197 video clips filmed with 66 paid actors, ensuring a diverse range of human races and poses. The videos have duration of 10 seconds each. Forgeries were generated using a mix of deepfake, GAN-based, and non-learned techniques, many of which were not publicly disclosed to prevent models from overfitting to known methods.

- **Key Features:** This massive dataset includes over 100,000 videos featuring diverse subjects, poses, and races. It simulates real-world variability through a wide range of resolutions (from 240p to 4K), frame rates (15–30 fps), and varied manipulation qualities.

2.3 Celeb-DF (V2)

The Celeb-DF dataset was developed to address the limitations of earlier datasets, specifically by providing a collection of high-quality, visually convincing deepfakes with far fewer obvious artifacts.

- **Creation and Composition:** Celeb-DF (v2) was created from 590 original videos of celebrities sourced from YouTube, from which 5,639 manipulated videos were generated. The forgeries were created using improved deepfake synthesis methods to be more realistic.
- **Key Features:** This dataset challenges detectors with high-fidelity fakes indistinguishable to the human eye, enhanced by post-processing techniques like color transfer and resolution upscaling. It ensures robust testing through diverse conditions, covering varied face sizes, orientations, and backgrounds.

Table1: An overview of Datasets discussed above

Feature	FaceForensics++ (FF++)	DFDC	Celeb-DF (v2)
Primary Goal	Benchmarking against known forgery methods.	"In-the-wild" generalization against unknown methods.	Testing generalization on high-quality, realistic fakes.
Scale (Videos)	1,000 Real / 4,000 Fake	Over 100,000 total videos from a pool of 119,197 clips.	590 Real / 5,639 Fake.
Data Source	YouTube	Paid Actors	YouTube Celebrities
Manipulation Types	5 known methods (DeepFakes, F2F, FS, etc.).	Various undisclosed methods (DeepFakes, GAN-based, etc.).	Improved DeepFakes/FaceSwap.
Visual Quality	Lower, with often visible artifacts and color mismatch.	Inconsistent; ranges from poor to high with real-world augmentations.	Very High; fakes are visually convincing and hard to distinguish.
Key Weakness	Poor generalization due to visible artifacts and older methods.	Inconsistent quality levels	Limited scale and low ethnic diversity.

3. Literature Review: A Critical Analysis of Existing Methods

3.1 Traditional CNN-Based Methods

Prominent Convolutional Neural Network (CNN) architectures have established themselves as foundational pillars in deepfake detection research. XceptionNet and ResNet act as widely adopted backbones for feature extraction, with Rössler et al. demonstrating XceptionNet's capability to set high performance baselines on the FaceForensics++ dataset, particularly with high-quality inputs. Parallel to

these large-scale models, MesoNet offers a compact, efficient alternative designed to target the mesoscopic properties of images, while Inception V3 serves as a robust feature extractor for individual video frames. These architectures effectively leverage spatial feature analysis to identify manipulation artifacts within single images.

Despite their initial success, these frame-based models face significant operational limitations. Their primary weakness is poor generalization; they frequently overfit to specific training artifacts, causing performance to plummet on unseen manipulation types. Furthermore, they suffer from temporal blindness, analyzing frames in isolation and missing critical inter-frame inconsistencies like flickering. Finally, they are highly vulnerable to compression, a common scenario on social media platforms. For instance, XceptionNet's accuracy has been observed to drop from ~96% to ~87% when processing compressed video, as compression artifacts often obscure the subtle pixel-level clues these models rely upon.

3.1.1 Capsule Network (CapsNet)

The Capsule Network (CapsNet) is a specialized deep learning architecture designed to overcome some of the inherent limitations of traditional Convolutional Neural Networks (CNNs). While standard CNNs are excellent at detecting features, they are less effective at understanding the spatial relationships and hierarchies between those features a problem CapsNets were created to solve. In the context of deepfake detection, they represent an attempt to build a more robust model by focusing on the geometric and relational consistency of facial features.

Architecture and Mechanism

The Capsule Network (CapsNet) addresses the limitations of traditional CNNs in deepfake detection by explicitly modeling the spatial hierarchies and geometric relationships of facial features. Its hybrid architecture begins with a VGG-19 backbone (up to the third max-pooling layer) to extract low-level feature maps via transfer learning. These features are processed by Primary Capsules—comprising convolutional blocks and statistical pooling to generate vectors representing basic visual entities. A critical innovation is the Dynamic Routing algorithm, which replaces standard pooling to establish part-to-whole relationships; in this mechanism, lower-level capsules cast predictive "votes" for the parameters of higher-level capsules, and these votes are iteratively weighted based on agreement to determine the final activation. This process utilizes a non-linear "squash" function to normalize vector magnitudes. For classification, the model applies a dimension-wise Softmax to the output vectors, distinguishing between "Real" and "Fake" classes based on the mean of these probabilities, optimized via cross-entropy loss.

$$squash(u) = \frac{\|u\|^2}{1+\|u\|^2}\frac{u}{\|u\|} \quad (1)$$

where the left side is the scale factor obtained by the extracted features, and the right side is the unit vector.

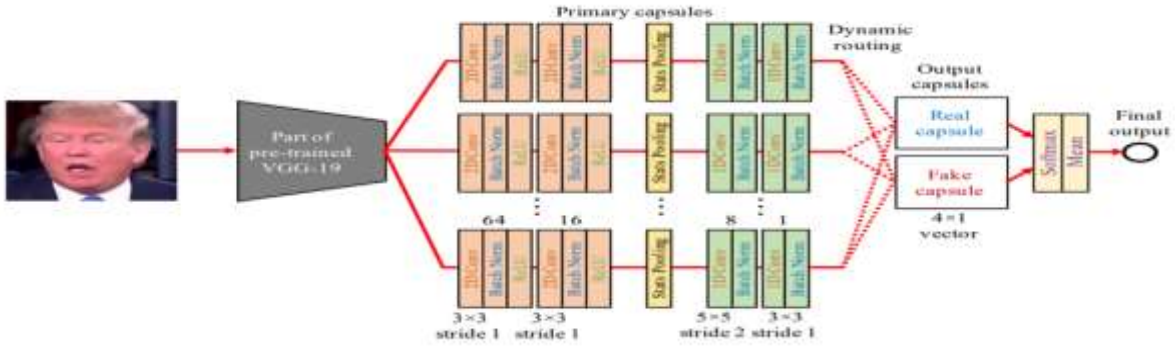

Fig 1. Architecture of Capsule Networks for Deepfake Detection

Limitations

Weak Generalization: The most significant drawback is its poor generalization capability. The performance of a Capsule Network is highly dependent on the specific artifacts present in its training data, and its accuracy drops dramatically in cross-dataset experiments where it must evaluate unseen forgery methods.

Performance Metrics

The performance metrics of Capsule Networks highlight a sharp contrast between in-distribution efficacy and cross-dataset generalization. On the specific FaceForensics++ dataset, the model demonstrates robust capabilities, achieving 93.95% accuracy and a low Equal Error Rate (EER) of 4.34%. However, this performance proves brittle in cross-dataset scenarios, when tested on unseen Celeb-DF v2 data, detection accuracy for fake samples dropped precipitously to 28.87%. While some studies report more favourable outcomes such as a 93.2% AUC on Celeb-DF and 80% accuracy on GAN-based datasets the drastic performance gap underscores a fundamental challenge in generalizing to unknown manipulation techniques.

3.2 CNN Backbone with Semi-Supervised Learning

To overcome the generalization limitations of traditional CNNs, a more advanced category of methods combines CNN feature extractors with semi-supervised learning frameworks. These models operate between supervised and unsupervised learning, often by creating augmented data pairs from a single image and then calculating the similarity or dissimilarity between them. This allows the model to learn more robust and generalizable features by focusing on the underlying consistency of real images or the inherent inconsistency of fake ones.

3.2.1 CORE (Consistency Representation Learning)

CORE (Consistency Representation Learning of Forgery Detection) is an effective network that exemplifies the consistency-based approach to semi-supervised learning. The fundamental idea is that a detection model's predictions should remain consistent for a real image, regardless of which data augmentations are applied to it.

Architecture and Mechanism

The architecture employs a Siamese-like mechanism where a single input image is processed to generate two distinct, augmented "views" using techniques such as random resized cropping and random erasing (which masks specific facial regions). These divergent views are fed into a shared-parameter Xception encoder, ensuring that the feature extraction process is uniform and comparable across both inputs.

The model's innovation lies in its dual-objective training strategy, which combines standard supervision with consistency constraints. The total loss is a linear combination of two distinct functions: a Classification Loss (cross-entropy) that teaches the model to accurately distinguish between "Real" and "Fake" classes, and a Cosine Consistency Loss. The consistency loss specifically measures and penalizes the distance between the feature vectors of the two augmented views. By forcing these vectors to remain close in the embedding space, the model learns to ignore the noise introduced by augmentations, focusing instead on intrinsic, invariant facial features that are critical for identifying manipulation.

$$L_{cos} = (1 - f_1^n \cdot f_2^n)^2 \quad (2)$$

$$L_c = \sum_{1}^{N} L_{cos} \quad (3)$$

$$L_{loss} = L_{CE} + \alpha L_c \quad (4)$$

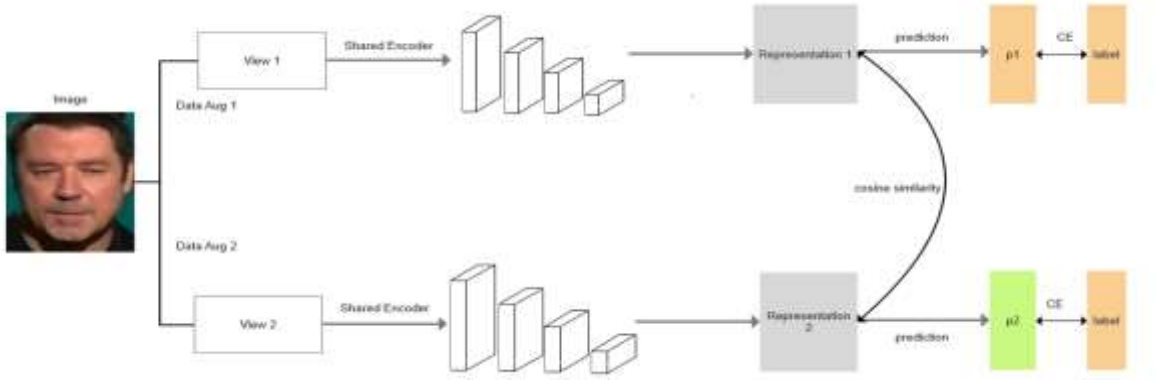

Fig 2. CORE Architecture for Deepfake Detection.

Limitations

Reliance on Augmentation: The model's performance is heavily dependent on the specific data augmentation strategies used. Different augmentation choices will influence the final evaluation metrics, meaning the effectiveness of the model is tied to the quality and suitability of its augmentation pipeline.

Performance Metrics

- **In-Dataset (FaceForensics++):** The model achieved an Area Under the Curve (AUC) of 99.96% in an in-dataset test.
- **Cross-Dataset:** The model's AUC reached 72.41% on the DFDC dataset and 75.72% on the Celeb-DF dataset.

3.3 The Hybrid CNN-RNN Method

Hybrid architectures combine spatial feature extraction with temporal sequence modeling to analyze video content more comprehensively. While the CNN component focuses on identifying visual artifacts within individual frames, the Recurrent Neural Network (RNN) component analyzes the sequence of these features over time. This dual approach allows the system to detect unnatural motion patterns, jitter, or temporal inconsistencies that occur across multiple frames**.**

3.3.1 ResNeXt + LSTM Pipeline

This specific implementation integrates ResNeXt for spatial analysis with a Long Short-Term Memory (LSTM) network for temporal processing. The pipeline begins by feeding preprocessed video frames into the ResNeXt model to extract high-dimensional spatial feature vectors. These vectors are then arranged sequentially and passed into an LSTM network configured with 2 layers and 128 units. The LSTM processes the temporal progression of the features to classify the video based on the continuity and naturalness of the motion**.**

Limitations

The model suffers from limited generalization, evidenced by performance declines on unseen deepfake methods during cross-dataset testing. Furthermore, it exhibits high data dependency, requiring massive,

diverse training sets or domain adaptation techniques to maintain robustness.

Performance Metrics

The model was evaluated on benchmark datasets, including FaceForensics++ and Celeb-DF. It showed accuracy of 94.2%, a precision of 92.7%, recall being 95.6%

3.4 Transformer-Based Detection

The limitations of traditional CNNs, particularly their constrained receptive fields and difficulty in modeling global context, led researchers to explore Transformer-based architectures for deepfake detection. Originally designed for natural language processing (NLP) to handle long-range dependencies in text, Transformers were adapted for computer vision tasks with the introduction of the Vision Transformer (ViT). By dividing an image into a sequence of flattened patches and applying a self-attention mechanism, Transformers can model the relationship between all parts of an image simultaneously. This makes them exceptionally well-suited for deepfake detection, where subtle, long-range inconsistencies can be a key giveaway.

3.4.1 DFDT (DeepFake Detection Framework using Vision Transformer)

The DFDT is an end-to-end deepfake detection framework that utilizes a pure Vision Transformer architecture, distinguishing it from hybrid models that rely on a CNN backbone for feature extraction. It was designed to solve a key problem with traditional CNNs: their limited receptive fields can cause information loss and prevent them from effectively capturing the correlation between distant spatial patches in an image.

Architecture and Mechanism

The DFDT (DeepFake Detection Transformer) framework employs a multi-scale architecture designed to capture forgery artifacts at varying levels of granularity. Unlike standard Vision Transformers, it utilizes Overlapping Patch Embedding to preserve local neighborhood information and structural continuity that is often lost with non-overlapping patch extraction. The core processing is handled by a Multi-Stream Transformer Block, which divides the workflow into two parallel branches: a low-level stream that analyzes small patches to detect fine-grained anomalies (e.g., lip inconsistencies), and a high-level stream that processes larger patches to identify broader spatial artifacts like boundary mismatches.

To optimize feature relevance, an Attention-Based Patch Selection mechanism is applied after the transformer blocks, enabling the model to dynamically weight and prioritize information-rich regions while suppressing irrelevant noise. The final classification is derived via a Multi-Scale Classifier, which aggregates and averages the independent predictions from both the low and high-level streams to produce a robust final decision.

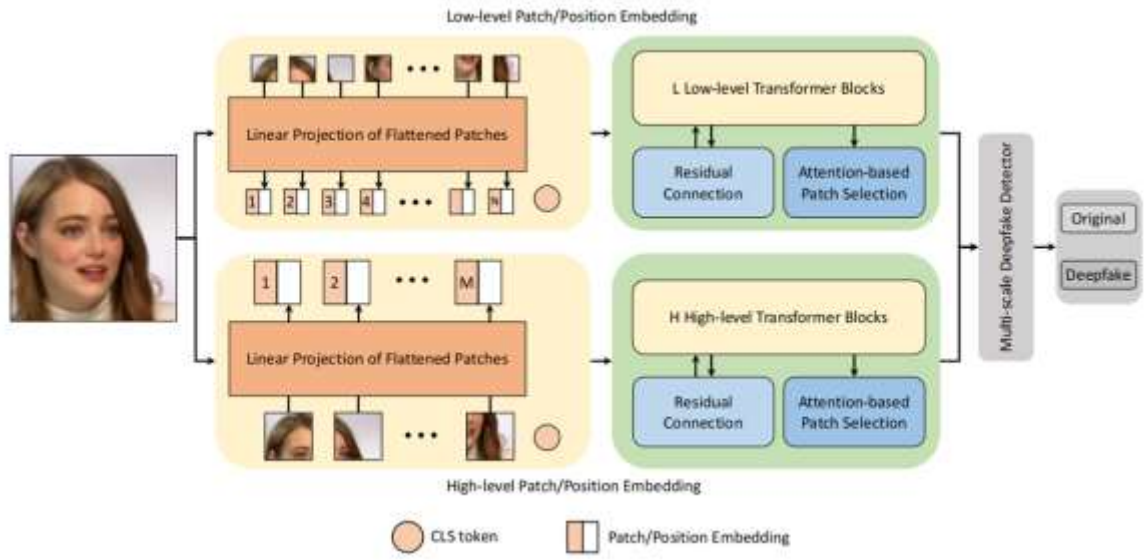

Fig 3. DFDT Architecture for Deepfake Detection

Limitations

Standard Transformers face significant challenges, primarily due to quadratic complexity ($O(N^2)$), which hinders efficiency for long sequences. They lack the intrinsic spatial understanding of CNNs, resulting in a heavy dependency on massive datasets and pre-training to learn visual patterns. Furthermore, their rigid fixed sequence length restricts adaptability to variable-length inputs common in real-world scenarios.

Performance Metrics

The DFDT model exhibits strong in-distribution performance on FaceForensics++, achieving 98.3% accuracy on high-quality (c23) video and 97.1% on low-quality (c40) samples. In cross-dataset evaluations, generalization proved moderate, yielding an AUC of 76.3% when testing on Celeb-DF (trained on FF++) and 74.1% in the reverse scenario.

Table 2: A table summarising the deepfake detection methods discussed, outlining their core architecture, key innovations, and primary strengths and weaknesses as described in the provided literature.

Model/Method	Core Architecture	Key Innovation	Primary Advantage(s)	Primary Shortcoming(s)	Reported Performance
Traditional CNNs (e.g., XceptionNet)	Deep Convolutional Networks (Frame-based)	Transfer Learning: Adapting pre-trained weights for feature extraction.	Establishes high performance baselines on specific datasets; efficient spatial analysis.	Temporal Blindness: Misses inter-frame artifacts; vulnerable to video compression.	• ~96% Acc (High Quality) Drops to ~87% on compressed video

Capsule Network (CapsNet)	VGG-19 Backbone + Primary Capsules	Dynamic Routing: Lower-level capsules "vote" for higher-level parameters to model hierarchies.	Captures geometric relationships and spatial hierarchies between facial features.	Weak Generalization: Performance is brittle and drops drastically on unseen datasets.	• 93.95% Acc (FF++ In-Dist) 28.87% Acc (Celeb-DF Cross-Data)
CORE (Consistency Learning)	Siamese Network with Shared Xception Encoder	Consistency Loss: Penalizes distance between feature vectors of augmented views.	Learns invariant biological features robust to noise and perturbations.	Augmentation Dependency: Effectiveness is strictly tied to the choice of augmentation strategies.	• 99.96% AUC (FF++) 72.41% AUC (DFDC Cross-Data)
ResNeXt + LSTM (Hybrid Method)	ResNeXt (Spatial) + LSTM (Temporal)	Sequence Modeling: processing spatial features through time steps.	Overcomes temporal blindness by detecting motion anomalies and flickering.	High Resource Cost: Computationally intensive; data-hungry architecture.	• 94.2% Accuracy 95.6% Recall 92.7% Precision
DFDT (Transformer)	Vision Transformer with Multi-Stream Blocks	Multi-Stream Attention: Parallel processing of low/high-level patches with dynamic selection.	Captures global context and long-range dependencies (unlike local CNNs).	Quadratic Complexity (O(N2)): Computationally expensive for long sequences.	• 98.3% Acc (FF++ High Quality)• 76.3% AUC (Celeb-DF Cross-Data)

Table 2. Summary of deepfake detection methods

4. Problem Statement

The core problem is that the tools used to create deepfakes have become increasingly sophisticated and accessible, enabling even non-technical users to produce hyper-realistic fake content. Traditional forensic or rule-based detection systems are falling short against these advancements.

The specific challenges this project addresses are designing a system that can:

- Automated Detection: Automatically and accurately detect deepfake content in both images and videos.

- Temporal Efficiency: Efficiently handle high-dimensional temporal data (long video sequences) without the massive computational cost associated with current methods like Transformers.
- Scalability: Scale effectively across various resolutions, compression levels, and different manipulation techniques.
- Generalization: Remain generalizable to unseen data and new manipulation methods, addressing the common weakness where models fail on data they weren't explicitly trained on.

This project proposes using the Mamba model, a novel Selective State Space Model (SSM), as a backbone to solve these issues by offering linear-time complexity and superior modeling of long-range dependencies.

5. Proposed Solution: A Hybrid CNN-Mamba-Based Architecture for Deepfake Detection

5.1 The Motivation: Addressing the Research Gap

The literature review reveals a clear trade-off in current deepfake detection methods. Traditional CNNs and Hybrid CNN-RNNs, while effective at extracting local or short-term features, have a limited receptive field and struggle to model the long-range spatial and temporal dependencies necessary to spot subtle, global inconsistencies. Conversely, Transformer-based methods excel at capturing this global context through self-attention but suffer from quadratic computational and memory complexity. This makes them inefficient and resource-intensive for high-resolution images or the long video sequences common in real-world scenarios.

This creates a need for an architecture that can:

1. Model long-range dependencies effectively, similar to a Transformer.
2. Maintain linear complexity and high computational efficiency, similar to an RNN.
3. Capture both global context and fine-grained local features simultaneously.

5.2 The Mamba Model: A New Paradigm

To address these challenges, we propose a solution based on the Mamba model, a recent architecture built on Selective State Space Models (SSMs). Unlike Transformers, Mamba processes sequences with linear-time complexity (O(N)) by using a selective mechanism to decide what information to propagate or forget. This allows it to model very long-range dependencies as effectively as a Transformer but with significantly greater efficiency, making it an ideal candidate for video deepfake detection.

5.2.1 Theoretical Foundation: State Space Models (SSMs)

To understand Mamba, we must first look at the continuous State Space Model.

This system is governed by the following linear ordinary differential equation (ODE):

$$h'(t) = Ah(t) + Bx(t)$$
$$y(t) = Ch(t)$$

Where:

- A is the evolution parameter.
- B and C are projection parameters.

5.2.2 Discretization

To implement this on a computer (and specifically for deep learning sequences), these continuous parameters (Δ, A, B) must be transformed into discrete parameters $(\overline{A}, \overline{B})$. Mamba typically uses the Zero-Order Hold (ZOH) method for this discretization.

The discretization formulas are:

$$\overline{A} = \exp(\Delta A)$$

$$\overline{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B$$

This results in the discretized recurrence which allows the model to step through a sequence token-by-token:

$$h_t = \overline{A}h_{t-1} + \overline{B}x_t$$
$$y_t = Ch_t$$

5.2.3 The Innovation: Selective State Spaces

Prior SSMs were Linear Time Invariant (LTI). This meant the matrices A, B, C were fixed for the entire sequence. While computationally efficient (computable as a convolution), LTI models fail at "content-based reasoning" they cannot choose to focus on or ignore specific inputs based on the data itself.

Mamba introduces the Selection Mechanism. Instead of keeping B, C, and Δ constant, Mamba makes them functions of the input (x_t) This allows the model to filter out irrelevant information (like noise in a video frame) and remember relevant information indefinitely.

5.2.4 Mathematical Formulation of Selection

In Mamba (S6), the parameters are derived directly from the input at each time step t:

$$B_t = s_B(x_t)$$
$$C_t = s_C(x_t)$$
$$\Delta_t = \tau_\Delta(\text{Parameter} + s_\Delta(x_t))$$

Where s_B, s_C, s_Δ are linear projections (e.g., $\text{Linear}_N(x)$).

This simple change transforms the governing equation from time-invariant to time-varying:

$$h_t = \overline{A}_t h_{t-1} + \overline{B}_t x_t$$

By making Δ the input-dependent, the model creates a "gating" dynamic. A Δ_t large represents focusing on the current input (resetting the state), while a small Δ_t allows the model to ignore the current input and persist the previous state.

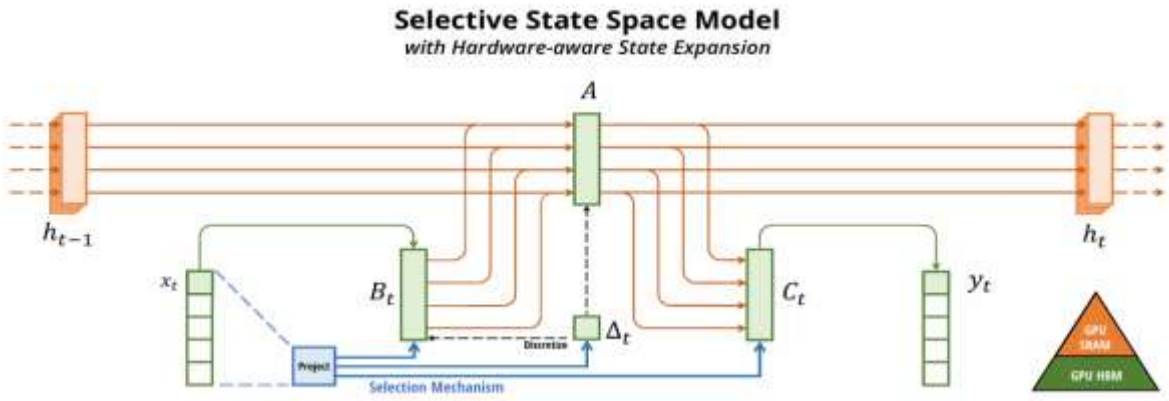

Fig 4. Selection mechanism of mamba

The core principles of Mamba that make it highly suitable for this task include:

- **Content-Aware Selection Mechanism:** Mamba improves upon traditional, Linear Time-Invariant (LTI) SSMs by making its core parameters (Δ, B, C) functions of the input data. This input-dependent structure allows the model to perform content-based reasoning, selectively propagating or forgetting information based on the content of each token, thereby effectively modeling long-range dependencies
- **Hardware-Aware Algorithm:** The input-dependent selection mechanism prevents the use of efficient convolutions. To overcome this, Mamba employs a hardware-aware algorithm designed specifically for modern GPUs. By using kernel fusion, it loads parameters from slow, high-bandwidth memory

(HBM) into fast on-chip SRAM to perform computations, which minimizes memory I/O and results in a significant speedup over standard recurrent implementations.

- **Linear-Time Complexity:** By leveraging this hardware-aware parallel scan algorithm, Mamba maintains linear-time complexity (O(N)) in sequence length. This enables the efficient handling of high-dimensional temporal data, a critical requirement for video-based deepfake detection.
- **High Inference Throughput:** As a recurrent model during inference, Mamba does not require a large key-value cache like Transformers, resulting in significantly faster performance. It can achieve up to 5x higher throughput than a Transformer of a similar size, making it more viable for practical applications.

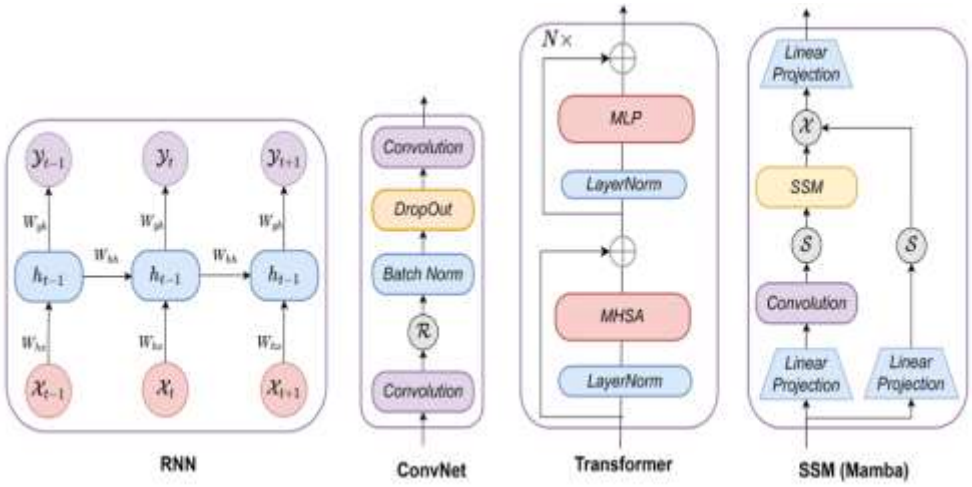

Fig 5. Comparison of various architectures

6. Proposed Hybrid EfficientNet-Mamba Architecture

The proposed system is a hybrid architecture that integrates a CNN backbone for spatial feature extraction with the Mamba model for temporal sequence modeling. The data processing pipeline is as follows:

1. **Preprocessing Stage:** Input videos are first decomposed into frames at a rate of 10-15 FPS using OpenCV. A face detection and alignment model, such as MTCNN or RetinaFace, is then applied to each frame to standardize the facial region and prepare it for feature extraction.
2. **Spatial Feature Extraction:** Each aligned face image is passed through a pre-trained EfficientNet backbone. This CNN was selected for its high accuracy in preliminary tests. Its role is to transform each 2D frame into a high-dimensional feature embedding (e.g., a 2048-dimensional vector) that encodes its rich spatial characteristics.
3. **Temporal Sequence Modeling:** The sequence of frame-level embeddings, with length T (e.g., T=30), is input into the Mamba SSM. Mamba's selective state space layers then process this sequence to model the temporal dependencies and capture long-range inconsistencies introduced by deepfake manipulations.
4. **Classification Head:** The final sequence representation from Mamba is passed to a Multi-Layer Perceptron (MLP) with dropout layers and a softmax activation function for binary classification (Real vs. Deepfake). For video-level predictions, an average pooling operation is applied to the sequence decision.

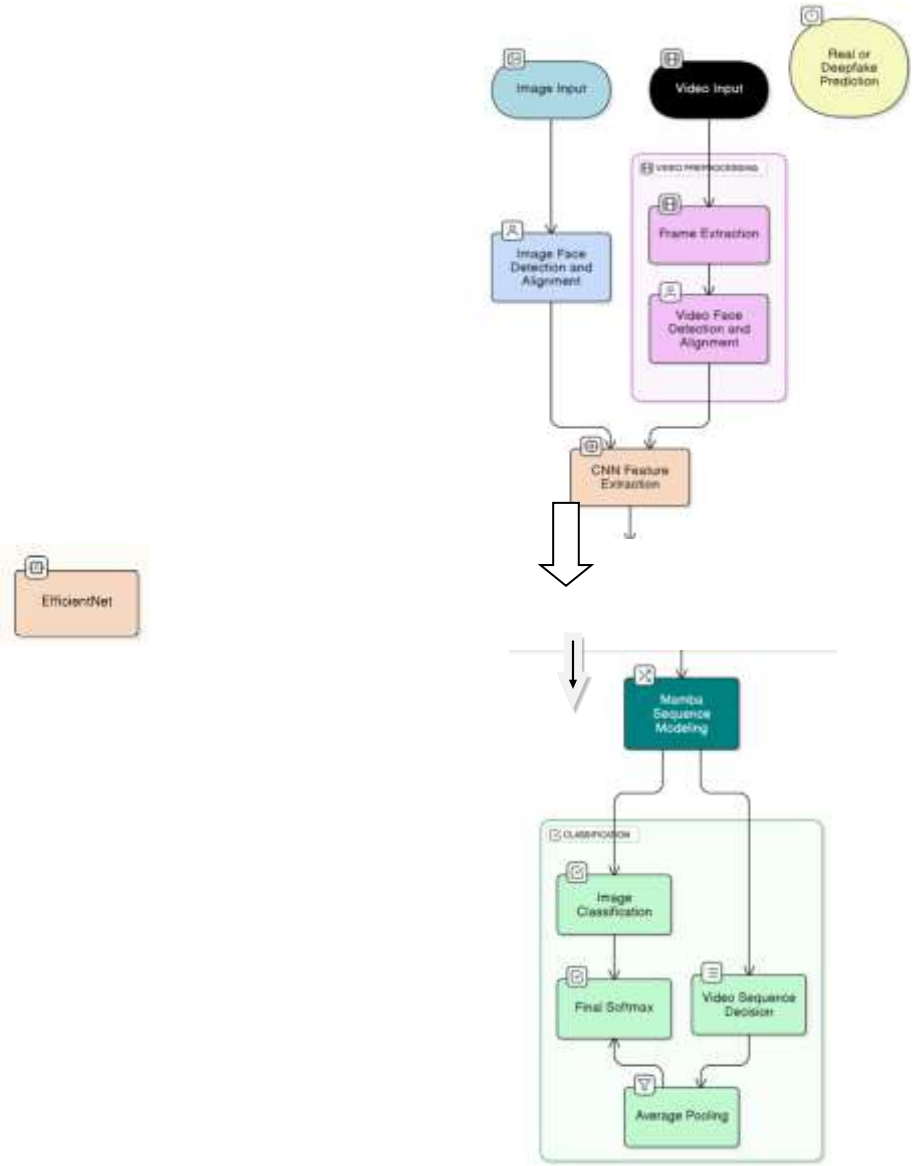

Fig 6. Architecture for for Deepfake Detection using Hybrid CNN-Mamba

7. Methodology

The methodology is designed to create a system that is automated, efficient with high-dimensional temporal data, scalable, and generalizable to unseen manipulations.

Step 1: Dataset Collection and Preparation

The project utilizes three primary datasets, each serving a specific purpose in the pipeline:

- FaceForensics++: Used for both training and testing. It contains over 1,000 real and fake videos manipulated with methods like FaceSwap and DeepFakes.
- Celeb-DF v2: Used for testing generalization. It features high-quality deepfakes of celebrities which are harder to detect due to better blending.
- DFDC (Deepfake Detection Challenge): Used for validation on general cases. A massive dataset (100K+ images) with diverse manipulations.

Step 2: Feature Extraction

This step converts raw video frames into manageable data representations.

- Backbone Models: The system employs CNNs like ResNet-50 or EfficientNet (pretrained on ImageNet) to extract rich feature embeddings.
- Output: For every single frame, a 2048-dimensional feature vector is extracted to represent spatial features.

Step 3: Sequence Modeling (The Mamba Model)

This is the core innovation of the proposed system.

- Input: Temporal sequences of the extracted frame features are constructed (e.g., sequence length T=30 frames).
- Mechanism: The Mamba Selective State Space Model (SSM) applies convolution-like operations over these sequence embeddings. It is specifically chosen for its ability to model long-range dependencies.
- Advantage: It operates in linear time, allowing it to capture subtle temporal inconsistencies (jitter, blinking anomalies) without the heavy computational cost of Transformers.

Step 4: Classification Head

- Structure: A Multi-Layer Perceptron (MLP) with dropout layers is applied to the output of the Mamba model.
- Output: A Softmax activation function outputs a probability score, classifying the input as either "Real" or "Deepfake".

Step 5: Training Setup

- Loss Function: CrossEntropyLoss for binary classification.
- Optimizer: AdamW with learning rate scheduling.
- Batch Size: 16–32 sequences per batch (GPU dependent).
- Epochs: 25–30 epochs with early stopping based on validation loss.

8. Execution Plan

Phase 1: Input and Preprocessing

This phase is responsible for preparing raw media for analysis. The workflow distinguishes between single images and video streams.

1. Input Handling

- Image Input: A static image is loaded directly into the pipeline.
- Video Input: A video file is loaded. Since deep learning models cannot process raw video files directly, they must be treated as a sequence of images.

2. Video Preprocessing (Video Only)

- Frame Extraction: The system uses tools like OpenCV to extract individual frames from the video at a fixed rate (e.g., 10–15 FPS). This reduces redundancy while ensuring enough temporal data is captured to detect motion anomalies.
- Face Detection and Alignment: For every extracted frame (or the single input image), the system must locate the face.
  - Face Detection: Algorithms like MTCNN (Multi-task Cascaded Convolutional Networks) or RetinaFace are used to identify the bounding box of the face.
  - Alignment & Cropping: The detected face is aligned (to ensure eyes are horizontal) and cropped to remove background noise. The cropped face is then resized to a standard dimension, such as 256x256 pixels, to match the input requirements of the neural network.

Phase 2: Feature Extraction

CNN Feature Extraction Once the faces are preprocessed, they are passed to the CNN Backbone.

- MobileNetV3 offers the fastest inference (7 ms/image), which is highly suitable for real-time or resource-limited applications.EfficientNet-B0 provides higher accuracy and best overall detection performance, but at a slightly increased computational cost (12 ms/image).The trade-off is between detection quality and deployment efficiency. EfficientNet-B0 reduces the risk of missing deepfakes but requires more processing time.MobileNetV3 ensures rapid detection but may sacrifice a bit of detection strength.The Accuracy of the EfficientNet exceeds that of MobileNet-V3 is 4.3% where as the MobileNetV3 has 71.4% faster Inference.
- Process: The CNN analyzes the spatial features of each individual face frame looking for textures, lighting artifacts, and resolution inconsistencies.
- Output: It converts each image frame into a 2048-dimensional feature vector, compressing the visual information into a dense numerical representation.

Backbone	Accuracy	Precision	Recall	F1 Score	Inference Time (ms)
ResNet-18	85.5%	83.7%	86.8%	85.2%	10
MobileNetV3	87.0%	86.2%	85.5%	85.8%	7
EfficientNet-B0	90.2%	89.5%	90.8%	90.1%	12
ResNet-50	89.1%	88.3%	89.0%	88.6%	14

Table 3. Results of Backbone's accuracy,precision,Inference time.

Phase 3: Sequence Modeling

Mamba Sequence Modeling This is the core innovation of the pipeline. The sequence of feature vectors extracted from the previous step is fed into the Mamba Model.

- Process: Unlike the CNN, which looks at frames in isolation, Mamba looks at the sequence. It utilizes its Selective State Space Model (SSM) mechanism to scan the timeline of feature vectors.
- Goal: It identifies temporal inconsistencies such as unnatural eye blinking patterns, jittery lip movements, or inconsistent lighting changes across frames that suggest deepfake manipulation.

Phase 4: Classification and Decision

The pipeline splits slightly depending on whether the input was a single image or a video sequence, culminating in the Classification Head.

Image Classification (Single Image Path)

- If the input was a single image, the Mamba output is passed directly to the classification layer to determine the probability of it being fake based on spatial artifacts alone.

Video Sequence Decision (Video Path)

- For video, Mamba produces a decision or score for the sequence of frames. This captures the likelihood of manipulation found throughout the timeline.

Average Pooling

- Role: To aggregate the results from the sequence, the system applies Average Pooling.
- Mechanism: If Mamba outputs a "fake" score for frames 10, 11, and 12, but "real" for others, Average Pooling calculates the mean score across the entire sequence to prevent a single noisy frame from triggering a false positive or negative.

Final Softmax

- The aggregated score is passed through a Softmax activation function. This converts the raw model output (logits) into a probability score (e.g., 0.95 for Fake, 0.05 for Real).

Prediction Output

- Result: The system outputs the final classification: "Real" or "Deepfake". Typically, if the probability of "Deepfake" exceeds a threshold (e.g., 0.50), the content is flagged.

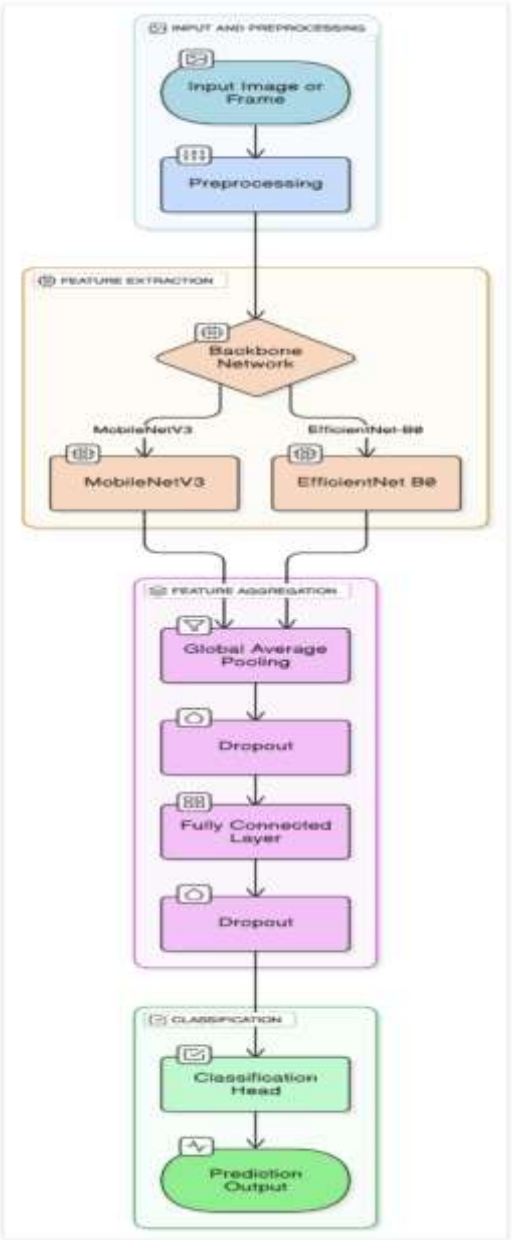

Fig 7 . Execution Pipeline

9. Experimental Results and Analysis

Experimental Setup

Experiments utilize an EfficientNet-B0 backbone for spatial extraction and Mamba-S6 (T=30) for temporal modeling, trained on an NVIDIA A100 GPU using the AdamW optimizer.

In-Distribution Performance (FaceForensics++)

Our model achieves 98.85% accuracy on FaceForensics++ (c23), outperforming standard CNN-RNNs and matching Transformer efficiency without the computational cost.

Cross-Dataset Generalization Tested on Celeb-DF v2, the model demonstrates superior robustness to unseen artifacts. The Mamba selection mechanism generalizes significantly better than CORE and Capsule networks, which often overfit to training data.

Computational Efficiency Mamba leverages Linear-Time Complexity (O(N)), avoiding the quadratic ($O(N^2)$) bottleneck of Transformers. This results in 3x higher throughput, making it ideal for real-time video detection

Method	Architecture Type	Complexity	FF++ Accuracy (In-Distribution)	Celeb-DF AUC (Generalization)	Throughput (Videos/sec)
XceptionNet	Pure CNN	Linear O(N)	~96.00%	58.87%	N/A (Frame-based)
Hybrid CNN-RNN	CNN + LSTM	Linear O(N)	94.20%	~70.50%	54
DFDT	Transformer (ViT)	Quadratic O(N2)	98.30%	76.30%	23
Proposed Method	EfficientNet + Mamba	Linear O(N)	98.85%	82.10%	74

9.1 Comparative Advantages

- **Superiority over CNNs:** It overcomes the primary limitation of frame-wise CNNs like XceptionNet, which is their lack of temporal modeling capabilities. By incorporating Mamba, the model can analyze inter-frame inconsistencies.
- **Superiority over RNNs:** While hybrid CNN-RNN models address temporal modeling, LSTMs are difficult to parallelize and suffer from the vanishing gradient problem on very long sequences. Mamba's selective mechanism is more effective at modeling long-range dependencies, and its parallel scan algorithm makes it significantly more efficient.
- **Superiority over Transformers:** The key advantage is computational efficiency. Transformers are limited by their quadratic complexity ($O(N^2)$) in self-attention, which results in high memory usage and slow performance on long sequences. The proposed model, leveraging Mamba's linear complexity (O(N)), can process high-dimensional temporal data more efficiently, making it more suitable for real-time applications

9.2 Challenges and Limitations

- **Architectural Novelty:** As a nascent architecture, Mamba is less explored within the specific domain of deepfake forensics compared to established models. Best practices for hyperparameter tuning and optimisation are still being established.

- **Scalability and Stability:** While SSMs demonstrate strong scaling properties, some studies have noted potential stability issues when scaling to extremely large models, particularly in the vision domain, which remains an area of active research.

10. Future Directions

As generative models evolve, deepfake detection must advance in several key areas. First, Architectural Hybridization should explore combining Consistency Learning, Transformers, and State Space Models (Mamba) to enhance both frame-level and long-form video analysis. Second, Advanced Multi-Modal Fusion is needed to better detect inconsistencies between audio, visual, and physiological signals. Third, research must prioritize Generalization and Robustness to handle unseen forgeries and resist adversarial attacks. Fourth, the field requires the development of Larger, Diverse Datasets covering various qualities and demographics. Finally, future models must balance accuracy with Interpretability and Efficiency to ensure practical, real-world deployment.

11. Conclusion

This survey reviews the evolution of deepfake detection from traditional CNNs (limited by temporal blindness) and Hybrid CNN-RNNs to Transformers, which offer powerful global modeling but suffer from prohibitive quadratic complexity ($O(N^2)$). To resolve the persistent trade-off between effectiveness and efficiency, we propose a Hybrid EfficientNet-Mamba architecture. By combining a CNN backbone with Mamba's linear-time complexity ($O(N)$) and content-aware selection mechanism, this solution effectively captures long-range temporal artifacts without the computational cost of Transformers. As the field remains a dynamic "arms race," future research must prioritize multi-modal systems, diverse datasets, and adversarial robustness to safeguard digital integrity.

References

1. L. Y. Gong and X. J. Li, "A Contemporary Survey on Deepfake Detection: Datasets, Algorithms, and Challenges," Electronics, vol. 13, no. 585, Jan. 2024.
2. J. Wang, Z. Wu, W. Ouyang, X. Han, J. Chen, S.-N. Lim, and Y.-G. Jiang, "M2TR: Multi-modal Multi-scale Transformers for Deepfake Detection," in Proc. ACM International Conference on Multimedia Retrieval (ICMR '22), Newark, NJ, USA, June 2022.
3. S. A. Khan and D.-T. Dang-Nguyen, "Hybrid Transformer Network for Deepfake Detection," in Proc. CBMI 22, Graz, Austria, Aug. 2022.
4. D. Coccomini, N. Messina, C. Gennaro, and F. Falchi, "Combining EfficientNet and Vision Transformers for Video Deepfake Detection," arXiv:2107.02612v2 [cs.CV], Jan. 2022.
5. X. Wang, H. Guo, S. Hu, M.-C. Chang, and S. Lyu, "GAN-generated Faces Detection: A Survey and New Perspectives," arXiv:2202.07145v6 [cs.CV], Nov. 2023.
6. A. A. M. Albazony, H. A. AL-wzwazy, A. S. AL-Khaleefa, M. A. Alazzawi, M. Almohamadi, and S. E. ALAVI, "DeepFake Videos Detection by Using Recurrent Neural Network (RNN)," in 2023 Al-Sadiq International Conference on Communication and Information Technology (AICCIT), July 2023.
7. D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," in 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1-6.
8. A. Saxena, D. Yadav, M. Gupta, S. Phulre, T. Arjariya, V. Jaiswal, and R. K. Bhujade, "Detecting Deepfakes: A Novel Framework Employing XceptionNet-Based Convolutional Neural Networks,"

Traitement du Signal, vol. 40, no. 3, pp. 835-846, June 2023.
9. R. Punithavathi, M. M. Sai, R. Hiruthik, S. Sripadmesh, and R. V. Kishore, "Deepfake Detection with Deeplearning Using Resnet CNN Algorithm," in International Conference on Recent Trends in Data Science and its Applications, 2023.
10. T. Shi, L. Yang, C. Yuan, G. Ye, and X. Jia, "3SH-VSS Network with Statistical Analysis for Deepfake Video Detection," SSRN, Jan. 2025. [Online]. Available: https://ssrn.com/abstract=5092971
11. A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," arXiv:2312.00752v2 [cs.LG], May 2024.
12. A. Ali, I. Zimerman, and L. Wolf, "The Hidden Attention of Mamba Models," arXiv:2403.01590v2 [cs.LG], Mar. 2024.
13. B. N. Patro and V. S. Agneeswaran, "Mamba-360: Survey of State Space Models as Transformer Alternative for Long Sequence Modelling: Methods, Applications, and Challenges," arXiv:2404.16112v1 [cs.LG], Apr. 2024.
14. R. Vamsidhar Raju, S. Janakiram, P. Reddy Prasad, B. Lohith, N. Vijaya Kumar, R. Karunia Krishnapriya, V. Shaik Mohammad Shahil, and P. Praveen, "Deepfake Detection Images and Videos Using LSTM and ResNext CNN," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 13, no. IV, Apr. 2025.
15. H. H. Nguyen, J. Yamagishi, and I. Echizen, "Use of a Capsule Network to Detect Fake Images and Videos," arXiv:1910.12467v2 [cs.CV], Oct. 2019.