National College of Ireland

Scalable, Privacy-Preserving, and Traceable Multimodal Deepfake Detection in a Cloud-Native Serverless Architecture

MSc Research Project
Cloud Computing

Sanjana Gavhane
Student ID: x23325178

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sanjana Gavhane
Student ID:	x23325178
Programme:	Cloud Computing
Year:	2024-2025
Module:	MSc Research Project
Supervisor:	Yasantha Samarawickrama
Submission Due Date:	11/08/2025
Project Title:	Scalable, Privacy-Preserving, and Traceable Multimodal Deepfake Detection in a Cloud-Native Serverless Architecture
Word Count:	8241
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	11th August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Scalable, Privacy-Preserving, and Traceable Multimodal Deepfake Detection in a Cloud-Native Serverless Architecture

Sanjana Gavhane
x23325178

As AI-generated deepfake content has heated up rapidly, there are highly critical issues of media validity, misinformation, and ethical responsibility. Current methods of detection are usually limited in their scalability, model independence, and weak privacy-preserving procedures, and are therefore more susceptible to failure in the real world, especially in cloud-based settings. The proposed solution in this research is a scalable, automated, and privacy-preserving deepfake detection system that combines a multimodal system, i.e., visual features and audio features to detect inconsistencies in manipulated content. Capsule Networks are used to extract facial features whereas the audio streams get classed to mel spectrograms, and their scores fused to provide better detection performance. The system is created in two phases. During the first phase, the model is trained offline on Google Colab with the FakeAVCeleb dataset. Single face frames are analyzed and visual features are extracted and audio streams turned into mel spectrograms. They are filtered through the Capsule + Score Fusion model with high accuracy and good classification results. The second phase consists of implementing a trained model in the cloud-native, serverless architecture (AWS offerings on S3, Lambda, and DynamoDB). As soon as video is uploaded to the S3 bucket, inference is initiated automatically and content is processed and results saved with related metadata- guaranteeing secure, version-controlled storage and auditability. This two-step scheme achieves both high levels of reliable, near real-time detection with high levels of privacy protections and scalability. The findings suggest the given architecture is technically competent and flexible, which is why it could be incorporated into the large-scale digital content verification-based workflow and media forensics.

1 Introduction

In our current digital age, the boundary between real and fake generated content is getting worse. One of the largest and the most disturbing advancements in the section is the increasing popularity of deepfakes Khichi and Kumar Yadav (2021), which involves video and audio recordings that are created and edited using Artificial Intelligence to convincingly mimic actual people Abbas and Taeihagh (2024). Although the technology underlying deepfakes is amazing, its negative use has generated general effects across the inaccurate information, security threats, and damage to reputation.Traditional detection's techniques, however helpful, are tend to be designed for experimental situations and not for real world practical problems. They depend entirely on visual indications and barely

consider the complicated methods by which fraudulent data could have been developed. In addition, many of these solutions lack effective privacy features and traceability, which are essential while processing with extremely vulnerable or personally identifiable data. This research takes a novel approach. It targets at developing a highly reliable, achievable, secure, and flexible detection method. The intension is to design a system that can maintain real-time detection at scale and still preserve privacy and customer trust by merging different types of information such as audio and video, applying cloud-native features, and valuing privacy of data.

1.1 Motivation

Deepfakes are much more than an idle novelty, they create a danger with regard to how we view the truth in the modern society.The adverse effects of uncontrolled deepfake technology, covering everything from media, speeches on politics to personal information fraud, are becoming noticeable Lundberg and Mozelius (2025). What is even more worrying is how easy it is to access and share this technology, that makes it problematic to manage it's continued growth. Regardless progess in detection of deepfakes, almost all current methods have either become excessively small in scale in detecting deeper alterations or very costly is terms of the resources to be implemented at scale. Automated detection systems which are automated, secure, and scalable are growing more and more. Such systems must also be able to operate in cloud, detect rapidly and maintain user data. The primary objective of this attempt is to fill that gap.

1.2 Problem and Contribution of the Research

The Problem
deepfake detection faces many issues which includes the slow and poor evaluation of huge amounts of media. Data security and privacy are usually ignored in system design. Detection usually depends entirely on visual factors, cutting-out audio video gaps. There is no embeded way for tracking changing over time or understand how a file changes, which makes it crucial for traceability. Existing limitations makes it very tricky and difficult to depend upon or scale, particularly within organizations that process huge amounts or highly sensitive data.

The Contribution
This project immediately tackles these challenges . The recommended cloud-based deepfake detection system makes the use of AWS serverless services(e.g. S3, Lambda, DynamoDB) to provide automatic and scalable detection methods. Multi-modal evaluation is executed through comparison of audio and video content. The protection of data can be achieved through the encryption, control of access and version control. Records a traceable trail of every record and every detection in order to have visibility and future analysis. By performing, the technique not only detects deepfakes, but it also creates trust in regard to how that detection is handled.

1.3 Research Question

How can dynamic content tracking, multi-modal analysis, and scalable cloud-native technologies be integrated to build a privacy-preserving and effective deepfake detection system?

1.4 Research Objectives

- Analyse the current drawbacks of deepfake detection technologies, paying particular focus on to the methods, privacy and scale.
- Create a platform based on cloud with the event-based approaches to provide automatic detection.
- Develop a multi-modal detection that will be grounded on the application of audio patterns and video patterns.
- To ensure security of the system, use the contents versioning, the accesses checks and the encryption to safeguard the user data.
- Examine how well the approach can scale, automate operations, and offer an infrastructure for future deepfake detection technology that will grow more effective.

1.5 Report Structure

The remaining part of this report is organized as follows: Section 2 provides the description of related work, overviews of studies conducted on the topic of multimodal deepfake detection and elaboration of the research gaps. Section 3 presents the methodology, which includes a description of the utilization of the publicly available FakeAVCeleb dataset, the preprocessing and training process. Section 4 presents the specification of the system design which consists of the architecture and functionality modules. Section 5 describes, the implementation process ideas that convert the design, into a serverless, cloud-native space. Section 6 includes results and analysis of in-domain model training and inference, as well as, real-time inference. Section 7 is the final section of the report and contains the conclusion of the research with the potential future work. Section 8 contains all references, which were included in the study. float

2 Related Work

In response to the overuse of synthetic media generated with the help of AI, deepfake detection has become an important research field very quickly. Although a lot has been achieved in improving the accuracy of models, there are some practical and architectural issues that still hold back real world adoption. These limitations are in 4 areas: modality dependence, scalability, data privacy and traceability - each of these has been handled differently by researchers.

2.1 The Limitations of Unimodal Detecting Strategy

Most deepfake detection systems investigated to date have focused on either the video Suratkar and Kazi (2023) Qadir et al. (2024) or audio signal Mcuba et al. (2023) as a source of characteristics based on which deepfakes may be detected. These methods are unimodal since they aim at individual objects of artistry such as unnatural blinking, facial deformations, or voice pitch and cadence anomalies. As single-modal detectors, the deepfake generation tools have been improving to generate content with more coherent cross modalities, and thus their effectiveness is increasingly becoming lower.

To counter this, researchers have started to move towards multimodal detection systems Salman et al. (2023) that are based on audio visual correlation analysis. An example thereof is the framework introduced by Javed et al. (2025) , which use diffusion models in order to refine noisy input and combine local and global features through both modalities. This improves the low-quality or compressed videos detection. The same way, Liu et al. (2024) perform a detailed survey tracking this evolution showcasing that although multimodal approaches lead to a vast increase in robustness, they bring a new type of challenges concerning cost of computations, matching and real time processing.

Moreover, multimodal systems are resistant to a more manipulative approach that unimodal models do not consider, including lip-sync mismatch or deepfake overlaid voice. These strategies, however, do not remain effective in real-life systems with an implementation that is not efficient enough, particularly in scaled or security-sensitive systems.

2.2 Scalability Deployment Challenges

Although detection models have shown success in an academic setting, the one major area of challenge is deployment in real time and large scale. Kaur et al. (2024) emphasizes on most approaches which are resource-intensive, consuming a lot of processing power making them unsuitable to be used in cloud-native systems where thousands of media files need to be processed per minute. Khan et al. (2025) also highlight the necessity of developing real-time detection systems as one of the research priorities with a critical limitation tied to scalability impeding effective implementation in high-throughput, practical applications.

When detection systems are installed correctly, it is common to find the same system is limited in its ability to respond promptly to bursts of traffic or streaming applications due to the fixed-infrastructure nature of detection systems. Moreover, the arrangements are challenging to manage and dynamically expand with a variety of platforms or distributed systems. This constrains their application in services such as real-time moderation, auto-detection flagging, and forensics.

These problems are immediately encountered in our proposed system due to its implementation using a serverless, event-driven cloud structure that enables an elastic scale of deepfake detection without having a dedicated infrastructure. This strategy enables processing close to real-time and does not require persistent compute resources as well, which suits contemporary cloud implementation practices.

2.3 Privacy Issues of the Present Detection Methods

A further area in which the current offerings cannot satisfy is in privacy. The assumptions that most detection pipelines have full access (not encrypted) to user media is not safe, especially given the generally un-secured nature of the media that they are analyzing. Through federated training, encryption and secure feature sharing, Wu et al. (2022) and Chen et al. (2023) explore privacy-preserving techniques and demonstrate that data anonymity and effective detection are compatible. Although there have been several developments in privacy techniques, the cost or intricacy often makes them unusable in practice. There should be assurance that the detection systems will comply with the law such as GDPR or HIPAA, when applied in the workplace or in public places and where sensitive films or biometric information may have to be captured. The technology addresses this gap by utilising cloud-native safety measures, which can include restriction

of access based on IAM, server-side data encryption and object-level versioning to achieve secure and auditable data management. These built-in features support the provision of low-friction privacy and automation and scale.

2.4 Problems with traceability and management of life cycle

Another problem of deepfake detection that is less discussed but important is provenance tracking, namely retaining the origin of the media as well as its time of access and versions. Hasan and Salah (2019) also proposed a blockchain-based approach that would preserve unalterable audit trails, and Wang et al. (2021) proposed FakeTagger, which uses deep learning to fix traceable markers in the facial images. These identifiers are resistant to a range of GAN-based modifications, allowing them to be recovered and used to trace the manipulated content roots to prevent further spreading.

These kinds of solutions will enhance the importance of trust especially where deepfake content can be used as evidence in a case. Nonetheless, the majority of academic models lack the intent of storing media history and generating long-run record of detection results and access data.

Our system also provides inbuilt provenance tracking of items via S3 versioning and DynamoDB logs to have an uninterrupted unbroken chain of events of every uploaded and processed file. This also enables transparency in the decision making and enables subsequent audits, version comparisons or reprocessing should there be an update in the detection models.

2.5 Exposure to Risk Factors of Adversarial Abuse

Evasion techniques are advancing as detection models do. Hou et al. (2023) illustrated that even the state-of-the-art detectors can be fooled by essentially small changes in statistical properties without any visual quality deterioration through adversarial perturbation. It brings out an arms race between detection and generation of fakes.

An example of this is especially threatening to be adversarial since they are able to keep the topicality of a fake and circumvent classifiers by employing insensible transformations. The threats are likely to become even more pronounced as generative diffusion models appear and it is necessary to monitor, validate, and develop more detection systems over time.

This fact means that it is crucial that the detection systems be not only accurate but also agile, transparent and traceable which allows constant monitoring and update that allows the system to be resilient towards new threats. The logging and versioning system built on the cloud allows continuous detection and forensic verification in dynamic threat settings in our system.

2.6 Summary of Research Gaps

Challenge	Identified Limitations	Our Solution
Modality Limitation	Visual-only Suratkar and Kazi (2023), Audio-only Mcuba et al. (2023), Diffusion multimodal Javed et al. (2025), Multimodal survey Liu et al. (2024) Salman et al. (2023)	Real-time multimodal fusion using AWS
Scalability	Resource-intensive Kaur et al. (2024), Real-time need Khan et al. (2025)	Cloud-native, serverless, event-driven design
Data Privacy	Federated privacy Chen et al. (2023), Secure sharing Wu et al. (2022)	IAM policies, encryption, federated workflows
Provenance Gaps	Blockchain Hasan and Salah (2019), FakeTagger Wang et al. (2021)	Object versioning + metadata logging
Adversarial Evasion	Perturbation attacks Hou et al. (2023)	Auditability and continual verification pipeline

Table 1: Challenges, Identified Limitations, and Our Solutions

3 Methodology

The chapter explains the methodology under which the proposed solution to deepfake detection will be built and will comprise a multimodal model of deepfake detection and a scalable privacy-aware cloud-native infrastructure. The research methodology includes two high-level points, namely: (1) designing the deepfake detection model, and (2) the implementation of an event-driven and serverless cloud architecture that allows it to process the inferences in real-time. All of the design decisions were arrived at through the imperative of necessitating a degree of accuracy in identifying the patterns with the requisite degree of scalability, data privacy, and real time application and performance.

3.1 Structure of the Detection Model

In this study, a Capsule Network (CapsNet) is used as detection model to study the spatial patterns and correlations of features on the face, and audio-visual fusion in the score is used to enhance the accuracy of the detection model, which is a deepfake. CapsNet is unlike conventional convolutional neural networks because it encodes features as what is called capsules that stores not only the probability of a feature being present but also stores its spatial orientation, which makes it less susceptible to changes in pose and viewpoint, at least at higher levels of the network, which is useful in detecting subtle manipulation in deepfake faces. In contrast to the classical unimodal detection models that consider only video or audio stream, the proposed technique compared the modality data, aiming to find inconsistency between them that has been one of the peculiarities of deepfakes. The audio-visual score blend mechanism combines the scores of prediction in the visual (face) and auditory (speech) branches later in the pipeline to give a single, combined classification. This makes it possible to identify inconsistencies between what is spoken and the movement of the lips, which is normally undetected by the unimodal detectors. It is trained and tested using the FakeAVCeleb benchmark, which is a famous dataset of both naturally produced and artificially created audio-visual material, both

synchronized and unsynchronized. With this multimodal fusion approach, the system obtains enhanced resilience when fronted with varying media types and is resistant to more advanced forms of manipulation.

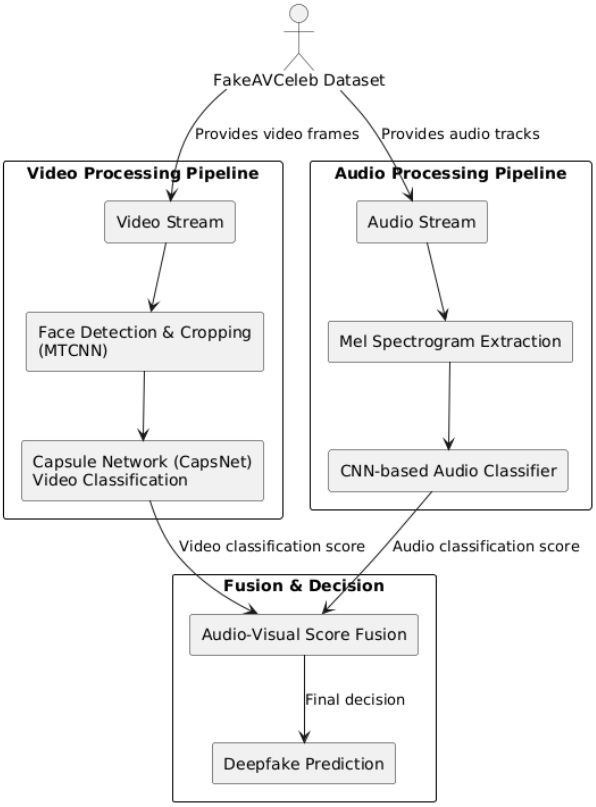

Figure 1: Structure of the Detection Model

3.2 Preprocessing and Data Preparation

3.2.1 Visual Preprocessing

The first step toward the visual pipeline is extraction of frames of every video. Its sampling methodology is innovative as the entire video does not take the same duration, which is, in this instance, a video greater than 60 seconds only requires a single image every second, whereas the shorter videos diligently use all these points. It will have the same treatment as the study on FakeAVCeleb. This is a trade-off between speed of computation and performance. Each of the frames retrieved is then filtered through the face detector using the MTCNN, which identifies facial feelings and outlines a rectangular field around the face where the most noticeable amount of changes is possible, which primarily concerns the mouth. The attached facial frames are scaled and balanced and then passed on to the visual feature extraction module.

3.2.2 Audio Preprocessing

The audio on each video is then automatically removed and converted to a mel spectrogram, which is a graphical representation of both the frequency and the time spectrum that is commonly used in communication and audio detecting applications. Transition process is composed of:

- Raw extraction of audio in Waveform Audio File Format(WAVs)
- It uses the Fast Fourier Transform (FFT) frequency domain transformation.
- To correspond better to the real sense of auditory perception, frequency scales are converted into the mel scale by using a log-compression process: The formula of Mel is

$$\text{Mel}(f) = \frac{\log\left(1 + \frac{f}{1000}\right)}{\log(2)} \times 1000$$

In this formula:

- f is input frequency in hertz (Hz).
- The value is divided by 1000 to put the frequency in a relative sense to 1 kHz, where human hearing sensitivity is just starting to vary.
- The logarithm (log) models the non-linear manner humans process pitch, with higher frequencies needing increased Hz variation in order to achieve the same sense of distance in pitch.
- Multiplying by a value of 1000 reduces the mel value back into a similar range.

To ensure compatibility, this process generates 2D spectrogram images with 4 seconds inter-time (frame) gap and standard sampling of frequency bandwidth (0-8000 Hz). These spectrograms are the input of specified functions called the audio extracted feature.

3.3 Architecture and Strategy of Model

Once the first processing is complete, the system retrieves attributes related to the two modalities automatically: Capsule Network studies an image of a face to measure spatial hierarchies and small anomalies in facial muscles motion or expressions, the ones that are most likely to be affected in generated faces. This audio spectrogram is then scanned through a small Convolutional Neural Network (CNN) that builds an understanding of the features that are usually associated with artificial speech, like pitch falls, frequency differences, or uncharacteristic pauses, and has already been trained on both natural and artificial speech in a reality with a supervisor. The system uses a score fusion methodology instead of joint integration or end-to-end fusion: the individual multimodal scores or probability results of each of the classifiers can be combined using a weighted averaging method, or some simple decision rule-based scheme. In this approach, the cross-modal evidence is adopted to make ultimate decisions and keep interpretability and modularity of the two classifiers.

3.4 Training and Evaluation Plan

Google colab is used to train and evaluate the model which has a set of resources that are backed by GPU with appropriate resources used to experiment and prototype deep learning. The training and evaluation is done with the FakeAVCeleb dataset. There are two stages of our evaluation:

Intra-domain: Model optimization and evaluation carried out between subsets of FakeAVCeleb to determine the baseline performance on non-new data under similar conditions of experimentation.

Inter- domain testing: Once it is trained, the model can be released to the cloud and tested on new video samples that have been uploaded to S3 on Amazon. Such uploads are not classes in the training/ test split and are diverse, unobserved data. The inference is processed through an AWS Lambda and will mimic a nearly real world detection where the input formats, resolution or manipulations might not be the same as those used during training.

The trained model is serialised and stored in Amazon S3 so that it can be used in serverless deployment and with use of AWS ecosystem, inference can be done in a scalable, event-driven fashion.

3.5 Cloud Service Architecture

All the parts in the inference pipeline will be using serverless components on AWS in order to have a responsive, real-time, platform-independent system. The factors used in the design are:

Amazon S3: It will be the trigger receiving point and storage point. A user uploads a video file to a bucket that has been created to handle a video and then the processing pipeline is triggered.

AWS Lambda: Is used to do all the model detection and uploading, the extraction of the feature of audiovisuals, and the inference. This event based architecture ensures that, the quick detection is also possible at almost close to real-time without using long-term servers.

DynamoDB: The NoSQL database keeps the timestamps, result of the detection, metadata of the file and the IDs associated to the user. This will help in very fast querying, retrieval and integration with dashboards or alerts.

CloudWatch: It monitors and logs the work of Lambdas to trace the bugs and various performance optimisations and security checks can be performed. This modular system allows the system to be scaled elastically enough to handle the load, reduce the volume of operation overhead, and serve real-time use cases such as automated content moderation or verification pipelines.

3.6 Privacy, Provenance and Security

As the media that the user posts may be of a sensitive character, the system has tight privacy and data protection measures in place: All data stored and transported is encrypted by S3 encryption at rest as well as HTTPS transport encryption. During the creation of Lambda functions, S3 buckets, and DynamoDB tables, IAM policies can be leveraged to ensure that only an essential part of them can be exposed. Versioning of S3 provides an entire audit trail of not just every file uploaded but every file reprocessed, allowing transparency and effective rollback operation. Results of inferences and executions are logged by DynamoDB, including timestamps and metadata of the origin in ways that facilitate reproducibility and forensic traceability. The combination of these security measures facilitates the conformity of the system with the contemporary data governance regulations as well as its usability and efficiency.

4 Design Specification

In this section we give a technical overview of architecture, modules and cloud services to be used in the proposed system of multimodal deepfake detection. The system is modular, scalable, privacy-preserving, and can be operated on serverless cloud infrastructure in real-time. All the components are custom designed to facilitate the detection process with ease, security, and transparency throughout the detection process.

4.1 System Design Overview

The given architecture is based on the modular, event-driven, and cloud-native architecture driven by Amazon Web Services (AWS) capabilities. It is optimized towards instant inference and performance-efficient deployment, which does not require the use of specialized infrastructure. After a video is uploaded, several actions are set to become automatic with the system making decisions and addressing the tasks of feature extraction, model inference, and result logging without any human involvement. The main services which have been used are:

- **Amazon S3** – To store media and serve as the source of triggering the event.
- **AWS Lambda** – Preprocessing logic and less inference are carried out on servers.
- **Amazon DynamoDB** – Holds detection results, metadata and identifiers of users.
- **Amazon CloudWatch** – Logs system events and it does real time monitor and debugging.
- **AWS IAM** – Governs the more detailed accesses between the services.

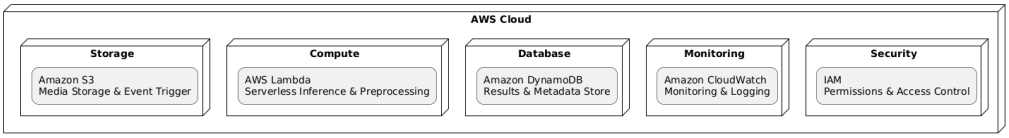

Figure 2: System Design Overview

4.2 Module Design

To make the architecture maintainable and to allow it to be scaled, it is split into independent modules, each of which carries out a very specific task in the detection pipeline.

A. Input and Trigger Module
Purpose: Manage uploads by the users and start the detection pipeline.
Technology: Amazon S3
Responsibilities:

- Upload videos either in `.mp4` or `.mov`.
- Create notification whenever a file is produced and trigger AWS Lambda.
- Apply bucket-level encryption, object versioning, and IAM access policies to prove provenance.

B. Preprocessing Module

Purpose: Be able to access useful video files attributes using visual and audio discrimination.

Technology: AWS Lambda, ffmpeg, MTCNN and librosa

Responsibilities:

- Utilize all the frames of shorter videos or 1 frame /sec of long videos.
- Using MTCNN, crop and detect faces.
- Transform the audio that is available in WAV format to mel spectrograms using the FFT.
- Use FFT to transform the available audio in WAV format to mel spectrograms.

This renders both modalities to be prepared in a similar fashion regardless of the dissimilarity of the inputs.

C. Inference / Fusion Module

Purpose: Use pre-trained deep learning to run deepfake detection.

Technology: Capsule Networks (visual) and CNN (audio), used in Lambda

Responsibilities:

- Load weights of pre-trained PyTorch model on S3.
- Do inference on face images and mel spectrograms separately.
- Use score fusion method to combine the modalities of prediction confidence.
- Compute a return response labelling (real or fake) with a related score of their confidence.

This factor and explainable modular fusion tactic increases robustness in detection under diverse input circumstances.

D. Logging Provenance Module

Purpose: Have transparency, auditability as well as storage of results.

Technology: DyanomoDB, CloudWatch, Versioning S3

Responsibilities:

- Log metadata: time of files upload, video title, prediction score, Lambda function ID.
- Results are stored in the DynamoDB with the schema:
  - `id` (partition key)
  - `timestamp`
  - `bucket`
  - `confidence`
  - `video key`
  - `result`
- Keep versioning of objects using S3 in case there is future comparison or rollback.
- Inspect Lambda with CloudWatch by streaming all of its logs.

4.3 Data Flow Description

The pipeline is driven event-based as shown below:

- **Upload:** Video file is uploaded to an S3 bucket by user.

  S3 registers an event notification which triggers lambda.

- **Preprocessing:** Lambda gets face frames and mel spectrograms.
- **Conclusion:** Model makes predictions and audio-visual scores are fused.
- **Storage:**The information regarding prediction, metadata as well as file versioning is stored.
- **Monitoring:** Visibility of streaming of logs and metrics is in CloudWatch.

This will take the form of an automated flow with low latency and high throughput needed by real-time applications.

4.4 Privacy / Security

Security and privacy are becoming the focus of the design of the system since media uploaded is sensitive. The following protections have been imposed:

- **IAM Role Segregation:** Each of the services is assigned to only the required privileges.
- **S3 Encryption:** Amazon S3 supports server-side encryption (SSE-KMS) using which there is secure storage of data with out hassle of dealing with keys to use manual key management.
- **Object Versioning:** It allows tracking and rollback on files.
- **No PII Stored:** None of the personally identifiable information (PII) is stored within metadata logs.

Such a privacy by design implementation also renders the system appropriate within a regulated context such as forensics or enterprise applications.

4.5 Design Reasoning

The designs are established on a number of priorities in the architecture:

- **Serverless Flexibility:** Using Lambda, we should be able to flexibly scale the system, depending on the demand, and reduce the number of idle costs.
- **Modularity:** Modules are self-contained and testing, updates and debugging is simplified.
- **Auditability:** The system is made transparent and traceable with the help of logs, versioning and metadata tracking.
- **Multimodal Accuracy:** Audio and video combinations present an improvement in robustness of various deepfake attacks.

- **Privacy-First Design:** Our encryption and role-based access as well as keeping minimal metadata logs grant confidentiality to the user.

A combination of these options enables the system to satisfy practical requirements of scalable, secure, and explainable capabilities of detecting deepfake.

4.6 Technology Stack

Layer	**Tool/Service**
Storage	Amazon S3
Compute	AWS Lambda
Model Hosting	S3 + PyTorch model serialization
Database	Amazon DynamoDB
Preprocessing	ffmpeg, OpenCV, MTCNN, librosa
Monitoring	AWS CloudWatch
Security	IAM, S3 Encryption, Versioning
Training Platform	Google Colab (CapsNet, CNN)

Table 2: Technology Stack by Layer

5 Implementation

This part elaborates on implementing the deepfake detection system as proposed and deployed with multimodal AI and secure, scalable cloud-native services. This solution will identify deepfake videos in using both facial and audio analysis in a privacy preserving and serverless architecture. The process of implementation was conducted in a modular form, and its stages included the preparation of a dataset and training of a model, followed by their deployment in the cloud and inference.

5.1 Toolchain and environment Configuration

Model training and development were conducted in Google Colab, which has a GPU-enhanced compute environment. PyTorch was used to construct the system, as it allows developing and training deep neural networks quite easily. Preprocessing workflows like frame extraction and audio feature creation were performed with the help of several tools including ffmpeg, librosa and MTCNN as a face detection algorithm. Integration of the model into a real-time, event-driven system was done by making use of Amazon Web Services (AWS). Video storage, processing, inference, and result logging, all server-side features, were built on the AWS native tools, including Amazon S3, AWS Lambda, Amazon DynamoDB, and CloudWatch. These services allowed having a fully serverless and auto-scaling pipeline, thus not requiring any manual server controls. There was no deployment scripting done with local IDEs/development platforms such as Visual Studio Code. Rather, other AWS services would be set up by use of the AWS Console or terminal based tools as required.

5.2 Dataset's Feature Engineering and Preparation

FakeAVCeleb is the main dataset applied to this project; it comprises created through AI and genuine video clips with synchronized visual and audio effects. Those videos have been uploaded to a specific Amazon S3 bucket (deepfake-detection-input) and centrally stored in the cloud and event-based processed. In case of the visual modality, individual frames were extracted using ffmpeg on each video. Sampling strategy was used: in case videos lasted more than 60 seconds, one frame per second was taken; otherwise, the video was completely decomposed. Based on these frames, it was possible to detect the faces, and crop them by using MTCNN, which implements face landmark detection to develop faces of high quality and assessable. Concurrently, there was extraction of audio in .wav format and transformation to mel spectrograms in the librosa library. It transforms raw waveform information into a frequency based image like representation, which is more representative to how the human auditory system 'hears'. This resulted in the 4-second last mel spectrograms to be reduced to 0 8000 Hz to make all the samples of the same size. Training and testing methods Normalisation All the features extracted using the face crops, and mel spectrograms were normalised and resized to feed the corresponding deep learning models.

5.3 Training and organization of the model

It has a two-model based detection engine that works upon a Capsule Network (CapsNet) that deals with the visual face aspect and Convolutional Neural Network (CNN) that operates on the audio spectrograms. Individual training of each model was done separately and their results that were processed were mixed in late-stage score fusion method that enhances detection accuracy. The CapsNet was selected due to the ability it has in preserving the poses and the spatial hierarchies in the images thus makes it suitable in representing minor manipulation in the facial expression in the pictures. CNN set to spectrograms was obsessed in establishing audio clangs such as abnormal cadence or robot-like voice transition used in the artificial speech. Data training would take place on Google Colab and then employ GPU acceleration to allow fast deep learning testing. Capsule Network (on the visual features) and CNN (on audio spectrograms) were trained on Adam optimizer and learning rate appropriate to reach a stable convergence. In order to enhance generalization of the resulting models, data augmentation methods, including horizontal flipping of images and pitch shifting of audio, will be implemented. When we have finished the training on FakeAVCeleb dataset, model weights will be serialized and placed on Amazon S3 that can perform serverless inference using AWS Lambda. Generalization itself is measured via intra-domain testing, which is carried out by assistance of the FakeAVCeleb dataset, and inter-domain testing, meaning the unseen videos were uploaded directly to the pipeline, establishing an experiment of real-life circumstances.

5.4 Server-less Cloud Deployment

The event-driven serverless architecture is the backbone of the real-time capability of the system. Each time a video is uploaded to S3 bucket, an event notification is generated into the Amazon which leads to a Lambda function that processes and classifies what it is given as an input. When invoked, the Lambda function:

- Loads CapsNet and CNN trained models stored in the S3 model bucket.

- Splits the video up into audio and frames.
- Performs face detection and conversion of spectrograms in real-time.
- Performs inference on a modality by modality basis and fuses the results through score fusion.
- Produces a binary output (real/ fake) and a confidence rate.

The result of the prediction, including metadata, including the time of the prediction upload time, model version, and Lambda execution ID, is stored in Amazon DynamoDB. Further, every activity of Lambdas functions is recorded through AWS CloudWatch and it supports real-time monitoring and after-analysis. This configuration guarantees deepfake detection to be fully automated, scaleable and low latency with little to no manual intervention required as the system can handle thousands of uploads at a time.

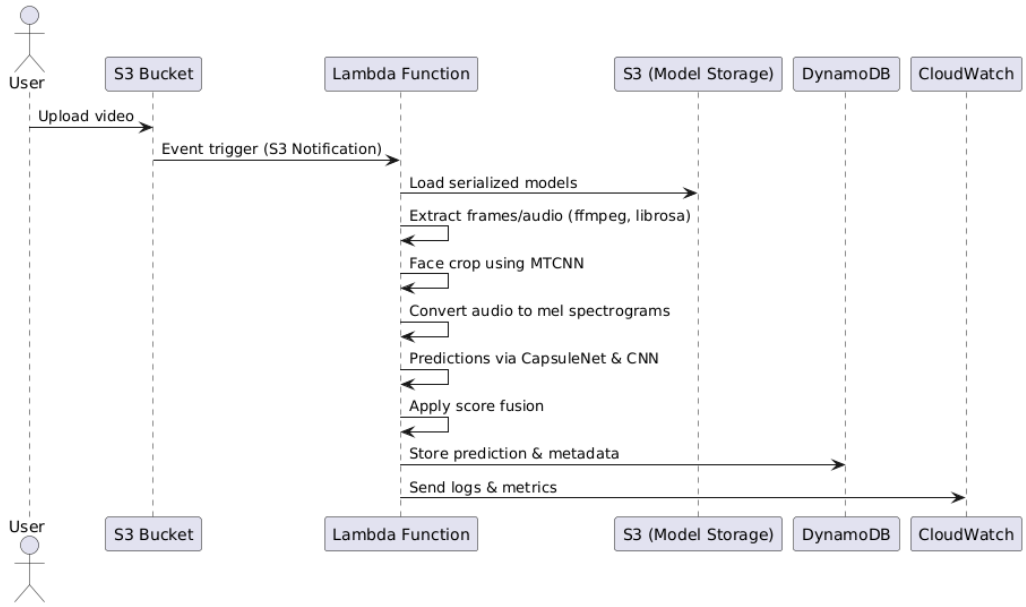

Figure 3: Serverless Cloud Deployment Architecture

5.5 Privacy, Logging, and encryption

Security, privacy and traceability was a crucial intended outcome of the implementation. Every single media and model file is resting at the Amazon S3 with the server-side encryption (SSE-KMS) enabled, and secure data storage is clearly achieved without any key management. The IAM roles and policies are strictly controlled to access control between services. As an example, the Lambda function can read the input S3 bucket and model bucket but only write to the DynamoDB table. This least privilege design reduces possible misuse and increases the security of the system. The object versioning in S3 is enabled to promote dynamic provenance tracking. This is to guarantee that conversion of current files or re-uploads will be archived and is accessible when rollback becomes a necessity as well as fully auditable as far as the treatment of the media is concerned. None of the information entered in the system is personally identifiable anywhere. The metadata stored in DynamoDB is tightly associated with the technical characteristics of the file, the result of the prediction, and system execution references thus is consistent with privacy-by-design frameworks.

5.6 System Test and Real World Verification

The system was tested in series to make sure that every part would do what was required of it when combined together as part of the entire pipeline. First, controlled uploads of lambdas were used to test the Lambda functions logic and S3 event triggers, which indicated that each video upload started to run the preprocessing and inference workflow as expected. The system was then tested on two sets of data. The FakeAVCeleb dataset was divided into reserved portions to perform intra-domain testing to ensure that the detection accuracy could be achieved in a data distribution that is well understood. Inter-domain testing activities included directly uploading unseen videos in the input S3 bucket which simulated a real-world deployment experience in which the format, quality, and the origin of the video changes. On every video that was processed, the predictions and metadata against them were checked by comparing the entries in the DynamoDB against the execution log in CloudWatch. These checks were to make sure that the output of inferences, the scores of confidence, as well as the system time stamps were stored as well as tracked appropriately. The entire system, including posting a video to S3 and storing inference outputs in DynamoDB, consistently took less than five seconds each and verified the system could be used in real-time applications like automated media verification and content monitoring

6 Evaluation

6.1 Background to the Evaluation Strategy

The performance of the proposed deepfake detection system was offered in two distinct phases, each of which is aimed to measure a different performance factor. Phase 1 examined the core detection algorithm on a controlled, offline training setup with a benchmark dataset and provided its accuracy and its generalization capacity. Phase 2 involved the deployment to the cloud and running the trained model as an end-to-end live application to test the real-time inference capability, scalability and reliability of the trained model. Using these divided evaluation steps, the results give the whole picture: the method not only produces a high level of classification performance in a controlled setting, but also has shown utility-scale operation in a near-production setting.

6.2 In-Domain and Model Training Evaluation

The first level of the assessment was done purely in Google Colab based on FakeAVCeleb dataset, with both real and synthetically created deepfake audio-visual data. Preprocessing was performed on each video and the output was a single representative image frame on the visual branch and a mel spectrogram as an audio representation. Data was divided into training (80 percent) and test (20 percent), to conduct intra-domain evaluation, that is, the model was measured on the test data of the same distribution as the training data. Five epochs were used to train the model. An epoch is a complete cycle over the total training data. Three major measurement parameters were tracked following every epoch:

- **Loss:** Reflects the degree of mismatch between the predictions of the model and the ground truth labels. Smaller loss means good performance. In our scenario, we

had a high loss in the first epoch but the loss steadily reduced with time indicating successful learning.

- **Accuracy :** It provides a ratio between correct prediction and all the prediction based on the percentage of time. Greater accuracy implies that the model is composed of fewer error classification.
- **AUC (Area Under the ROC Curve) :** AUC is the probability that model can correctly classify real and fake videos on all possible classification thresholds. In the multimodal configuration considered in this project, a larger AUC implies that the system is more stable at scoring genuine audio-visual pairs higher than manipulated ones, which implies good discriminative ability in both modalities.

Epoch	Loss	Accuracy (%)	AUC (%)
1	0.452	89.50	80.23
2	0.310	93.20	84.15
3	0.240	95.40	85.90
4	0.185	97.00	86.45
5	0.150	98.01	87.02

Table 3: Phase 1 (Offline) Training Results

Based on the table, it is evident that the model started to improve very fast in the initial two epochs, with accuracy of 99.50 after Epoch 2. The loss reduced by 10.0820 in Epoch 1 to 1.3870 in Epoch 5 which implies effective convergence. AUC went up in the same way as well, AUC went up on 93.11 to 98.26 which means that model has got now a lot more certain about separating real and fake. A visualization of the changing trends in loss; accuracy; and AUC is shown in figure 9.1. Similarly, the loss significantly decreases and the accuracy of the same stabilizes after Epoch 2, signifying that the Capsule Network when being complemented with a fusion of audio and visual scores is efficient in capturing the subtle asymmetries that occur between modalities in deepfakes.

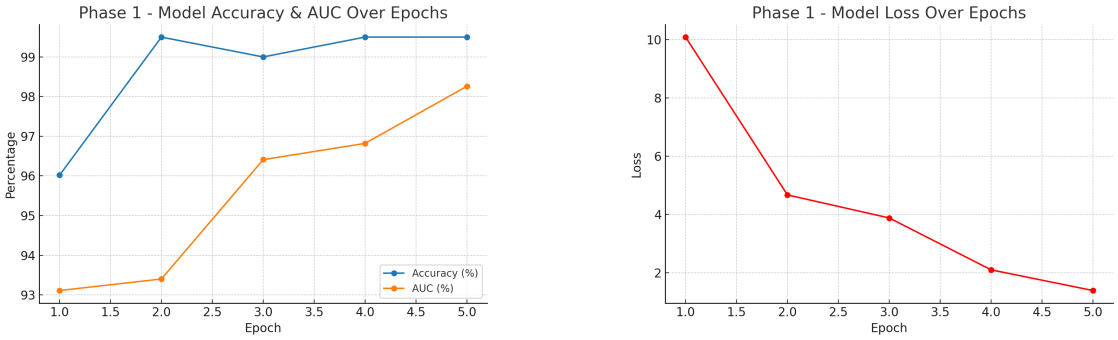

Figure 4: Model Accuracy, AUC and Loss Over Epochs

Figure 4 plots the losing trends, accuracy, and AUC over time. The drastic loss reduction and accuracy stability after Epoch 2 suggest that the Capsule Network coupled

with audio-visual score fusion is sufficiently able to capture the nature of modalities deviations that are peculiar to deepfakes.

6.3 Real Time Cloud Inference Output

In this step, the trained model was deployed into a cloud-native, serverless detection pipeline run fully on AWS. This has given a chance to test the model in real operational conditions. The live test consisted in uploading videos to an Amazon S3 bucket that launched an AWS Lambda process automatically. This Lambda function downloaded the trained model and an uploaded video, extracted face and audio features, inferred with a model Capsule + Score Fusion and saved the inference to Amazon DynamoDB.

Video ID	**Predicted Label**	**Probability of being FAKE (%)**	**Source (video_key)**
Video 1	FAKE	99.94	realtime-demo/00021_id00055_wavtolip.mp4
Video 2	FAKE	96.54	realtime-demo/00171_fake.mp4
Video 3	REAL	2.87	realtime-demo/00278_fake.mp4
Video 4	REAL	2.82	realtime-demo/00241_fake.mp4

Table 4: Real-time Inference Results

The table above presents a summary of predictions produced on the real-time inference pipeline on four different video sources. Predicted Label column denotes an extent, whether system interpreted the video as genuine or manipulated, and Probability of being FAKE (%) contains the model certainty of its classification. High confidence levels e.g. 99.94 % in Video 1 denote that the system is very confident in various instances of manipulated content and on the other hand, when the values become very low e.g. 2.82 % and 2.87 % in Video 3 and Video 4 respectively, the system is very confident that the videos are authentic. This diversity of sources also makes sure that the system is put to the test of different inputs, and the fact that this accuracy, reliability, and resiliency of the obtained architecture are shown through a stable correlation of prediction labels and expected results indicates that the deployed system is highly accurate and reliable.

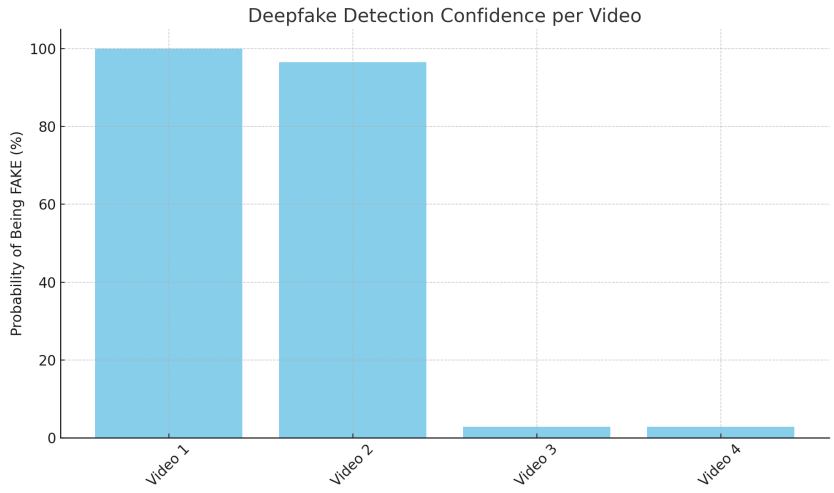

Figure 5: Real-Time Cloud Inference Output for Deepfake Detection

In contrast to the In-Domain Model Training Evaluation where the model was tested on the test data of the controlled dataset with the fixed train-test splits, the Real-Time Cloud Inference Output phase performed the test in the real operating context and in real-time. This involved the processing of several different video inputs, both real and deepfake and producing correct predictions with high confidence scores at all times. The cloud-native architecture proved reliable, scalable, and practically applicable due to the successful implementation of the pipeline, which is uploaded to the cloud sources, classified, and results stored without any interruptions or failing.

6.4 Real Time Cloud Inference Output

The comparative analysis of the In-Domain Model Training and Real-Time Cloud Inference phases proves the stability and applicability of the offered system of the deepfake detection. The shift in a controlled benchmark-driven training scenario to a live cloud deployment event-driven training cloud scenario did not significantly disturb predictive consistency, or decision confidence. Instead, this model achieved its capacity to detect subtle audio-visual anomalies even in the presence of the randomness of real life uploads. The fact that performance has been seamlessly transferred to two different phases confirms the strength of Capsule Network compatible with score fusion not only at the flow level but also the scalability and resilience of the serverless, cloud-native design. The system, therefore, proves to be ready to be applied practically, as it provides a secure, adaptive, and production grade solution to multimodal deepfake detection.

7 Conclusion and Future Work

The aim of the research was to develop and deploy a scalable privacy-preserving multimodal deepfake detection that can run in real-time. The combination of visual feature extraction using Capsule Networks and audio-visual fusion approach in scores potentially increases the usefulness of complementary cues when searching both modalities, improving detection against deepfake manipulations across a broad variety of manipulations. The model has been accurately trained on the FakeAVCeleb dataset and validated against the in-domain and real-time inference and predictably, returned high accuracy and decisive classification scores. The serverless implementation used AWS, so that the system could be served without fully dedicated infrastructure and will be automatically scalable to accommodate diverse workloads in the cloud, as well as be encrypted and with minimal metadata logging alongside a version-controlled storage to ensure user privacy. These evaluation outcomes showed that the method is not only capable of achieving the research goals but it will also have a strong framework so it can generalize collections based on a benchmark to a working cloud environment. The contribution of this work is the demonstration that the gap between controlled experimental environments and media in the real world can be bridged without compromise to transparency, scalability, and security using a multimodal, event-driven, cloud-native architecture. With this set of findings, it can be concluded strongly that the solution proposed is not just technically correct but it is also able to be incorporated into bigger digital content verification ecosystems in which the concepts of trust, security, and scalability are essential.

Although the system demonstrates good detection abilities and operation stability, future work may aim at expanding the system and its functionality. Inclusion of multimodal

datasets of larger scale and more diverse formats will enhance generalization to the content of different sources, forms, and manipulation methods. Using other modalities-textual transcripts or biometric indicators- might add value to the contextual signals to classify. Architecturally, the pipeline might be adjusted to support streaming-based or near-real-time video processing thus cutting down latency to areas such as live broadcast validation. Use of explainable AI practices would enhance interpretability, which is likely to enhance the seriousness of end users in cases, such as journalism or court trials. Finally, optimising the solution to operate at the edge can potentially make privacy-preservation detection possible at the point of content capture or distribution, potentially reducing centralised infrastructure dependence without sacrificing privacy.

References

Abbas, F. and Taeihagh, A. (2024). Unmasking deepfakes: A systematic review of deepfake detection and generation techniques using artificial intelligence, *Expert Systems with Applications* **252**: 124260.
URL: *https://www.sciencedirect.com/science/article/pii/S0957417424011266*

Chen, B., Liu, X., Xia, Z. and Zhao, G. (2023). Privacy-preserving DeepFake face image detection, *Digital Signal Processing* **143**: 104233.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1051200423003287*

Hasan, H. R. and Salah, K. (2019). Combating Deepfake Videos Using Blockchain and Smart Contracts, *IEEE Access* **7**: 41596–41606.
URL: *https://ieeexplore.ieee.org/document/8668407/*

Hou, Y., Guo, Q., Huang, Y., Xie, X., Ma, L. and Zhao, J. (2023). Evading DeepFake Detectors via Adversarial Statistical Consistency. Version Number: 1.
URL: *https://www.researchgate.net/publication/370228147*

Javed, M., Zhang, Z., Dahri, F. H. and Kumar, T. (2025). Enhancing multimodal deepfake detection with local–global feature integration and diffusion models, *Signal, Image and Video Processing* **19**(5): 400.
URL: *https://link.springer.com/10.1007/s11760-025-03970-7*

Kaur, A., Noori Hoshyar, A., Saikrishna, V., Firmin, S. and Xia, F. (2024). Deepfake video detection: challenges and opportunities, *Artificial Intelligence Review* **57**(6): 159.
URL: *https://link.springer.com/10.1007/s10462-024-10810-6*

Khan, A. A., Laghari, A. A., Inam, S. A., Ullah, S., Shahzad, M. and Syed, D. (2025). A survey on multimedia-enabled deepfake detection: state-of-the-art tools and techniques, emerging trends, current challenges & limitations, and future directions, *Discover Computing* **28**(1): 48.
URL: *https://link.springer.com/10.1007/s10791-025-09550-0*

Khichi, M. and Kumar Yadav, R. (2021). A Threat of Deepfakes as a Weapon on Digital Platform and their Detection Methods, *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 01–08.
URL: *https://ieeexplore.ieee.org/document/9580031*

Liu, P., Tao, Q. and Zhou, J. T. (2024). Evolving from Single-modal to Multi-modal Facial Deepfake Detection: Progress and Challenges. Version Number: 4.
URL: *https://www.researchgate.net/publication/381318583*

Lundberg, E. and Mozelius, P. (2025). The potential effects of deepfakes on news media and entertainment, *AI & SOCIETY* **40**(4): 2159–2170.
URL: *https://doi.org/10.1007/s00146-024-02072-1*

Mcuba, M., Singh, A., Ikuesan, R. A. and Venter, H. (2023). The Effect of Deep Learning Methods on Deepfake Audio Detection for Digital Investigation, *Procedia Computer Science* **219**: 211–219.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1877050923002910*

Qadir, A., Mahum, R., El-Meligy, M. A., Ragab, A. E., AlSalman, A. and Awais, M. (2024). An efficient deepfake video detection using robust deep learning, *Heliyon* **10**(5): e25757.
URL: *https://www.sciencedirect.com/science/article/pii/S2405844024017882*

Salman, S., Shamsi, J. A. and Qureshi, R. (2023). Deep Fake Generation and Detection: Issues, Challenges, and Solutions, *IT Professional* **25**(1): 52–59.
URL: *https://ieeexplore.ieee.org/document/10077834*

Suratkar, S. and Kazi, F. (2023). Deep Fake Video Detection Using Transfer Learning Approach, *Arabian Journal for Science and Engineering* **48**(8): 9727–9737.
URL: *https://link.springer.com/10.1007/s13369-022-07321-3*

Wang, R., Juefei-Xu, F., Luo, M., Liu, Y. and Wang, L. (2021). FakeTagger: Robust Safeguards against DeepFake Dissemination via Provenance Tracking, *Proceedings of the 29th ACM International Conference on Multimedia*, ACM, Virtual Event China, pp. 3546–3555.
URL: *https://dl.acm.org/doi/10.1145/3474085.3475518*

Wu, M., Wang, F., Wu, X., Yu, F., Wang, B. and Song, Z. (2022). Deepfake Detection with Data Privacy Protection, *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, Shanghai, China, pp. 1–5.
URL: *https://ieeexplore.ieee.org/document/9949458/*