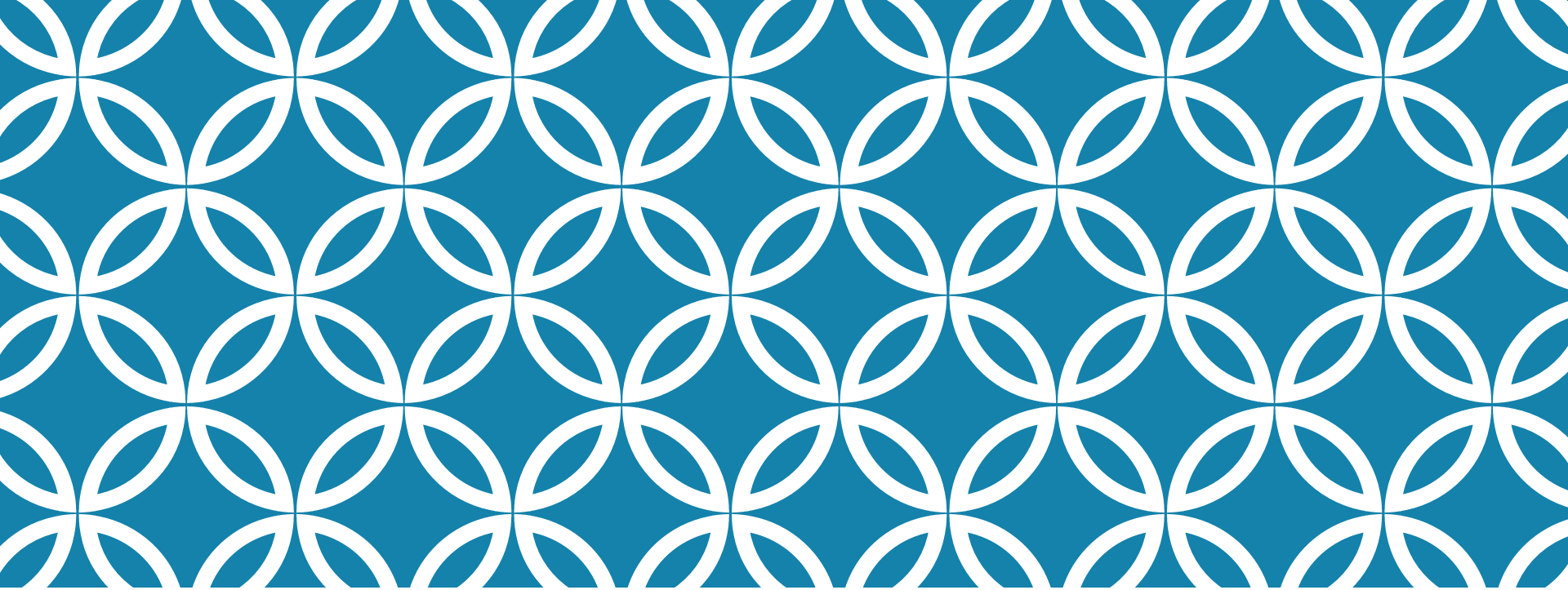


BASE PAPER DETAILS

TITLE : Artificial Intelligence for Detection, Estimation, and Compensation of Malicious Attacks in Nonlinear Cyber Physical Systems and Industrial IoT

Published in: IEEE Transactions on Industrial Informatics (Volume: 16, Issue: 4, April 2020)

Date of Publication: 28 November 2019

ABSTRACT

cyber security mechanisms, such as intrusion detection and prevention systems and access control do not have the capability to detect, prevent and block this category of cyber-attacks since the zero-day threats exhibit an unknown misbehavior that are not defined in signatures' database of the security systems. Recently, a new era of cyber security mechanisms based on artificial intelligence (AI) are under development to protect CPSs from these zero-day attacks. In the context of cyber security, machine learning technologies are used to manage a huge amount of heterogeneous data that come from different sources of information with a goal of generating automatically different attack patents and hence predicting accurately the future attackers' misbehavior. Here in this we have approached fuzzy logic that have been used in the context of cyber defense to solve the suspect is an attacker and to eradicate it.

LITERATURE REVIEW

TITLE	AUTHOR	YEAR	PAPER
Detection and compensation of covert service-degrading intrusions in cyber physical systems through intelligent adaptive control	F. Farivar, M. Sayad Haghighi, S. Barchinezhad, and A. Jolfaei	2019	In an intrusion detection and compensation framework is designed based on system identification to fight covert attacks. Errors of the output estimation are collected during the learning phase of system operation and after that, the system behavior is monitored to see if it significantly deviates from the expected outputs. A compensating controller is also designed to intervene and replace the classic controller once the attack is detected
Intelligent robust control for cyber-physical systems of rotary gantry type under denial of service attack	M. Sayad Haghighi, F. Farivar, A. Jolfaei, and M. H. Tadayon	2019	In an control approach is presented for tolerant control and compensation of cyber attacks occurred in inputs and outputs of a CPS of rotary gantry type. The malicious attacks are assumed to be of Denial of Service (DoS) kind and cause packet loss with high probability in the two signals; control input and output sensor. In the paper, some classic and intelligent control strategies are studied in terms of robustness and effectiveness against cyber attacks.
A sliding-mode scheme for monitoring malicious attacks in cyber-physical systems Robust detection and reconstruction of state and sensor attacks for	M. Corradini and A. Cristofaro M. L. Corradini and A. Cristofaro	2019 2019	A WECC network power system under attack is modelled as linear systems subject to unknown inputs altering the state attack and the sensor attack. sliding mode observers are designed for both attack monitoring and reconstruction within a finite-time. In[3], the extension of [4] is done. The attack compensation is carried out when the output tracks a given trajectory.

INTRODUCTION

- Data security is now more vital than ever. Updating existing cyber security solutions and enforcing every possible applicable security layer doesn't ensure that your data is breach-proof.

- But, having a strong support of advanced technologies will ease the task of security professionals. AI can efficiently analyze user behaviors, deduce a pattern, and identify all sorts of abnormalities or irregularities in the network. With such data, it's much easier to identify cyber vulnerabilities quickly.

- Contrarily, the responsibilities which are now dependent on human intelligence will then be susceptible to malicious cyber programs imitating legitimate AI-based algorithms. Several organizations are rushing into getting their machine-learning-based products out in the market.

- Relying on "supervised learning" is another threat. Under this, the algorithms label the data sets as per their nature. It could be malware, clean data, or some other tag. Cybercriminals, if they get access to the security firm, can alter the label as per their convenience. Also, routine tasks relying on AI can be manipulated by advanced hacking campaigns through the use of machine learning.

PROBLEM STATEMENT

Protecting the security and privacy of data in computer network systems is a major challenge in the modern computer age. Furthermore, social network systems (SNS) are convenient and easy to use albeit they pose a considerable amount of risks and concerns to all active members. The ease of assessing information has made SNS prone to cybercriminals.

EXISTING SYSTEM

- An intelligent classic control system is developed to compensate cyber-attacks.
- Neural network (NN) is designed as an intelligent estimator for attack estimation and a classic nonlinear control system based on the variable structure control method is designed to compensate the effect of attacks and control the system performance in tracking applications.
- In the proposed strategy, nonlinear control theory is applied to guarantee the stability of the system when attacks happen.
- In this strategy, a Gaussian radial basis function NN is used for online estimation and reconstruction of cyber attacks launched on the networked system. An adaptation law of the intelligent estimator is derived from a Lyapunov function. Simulation results demonstrate the validity and feasibility of the proposed strategy in car cruise control application as the test bed

DRAWBACKS

Level of prediction is low

Traditional algorithms are major drawbacks

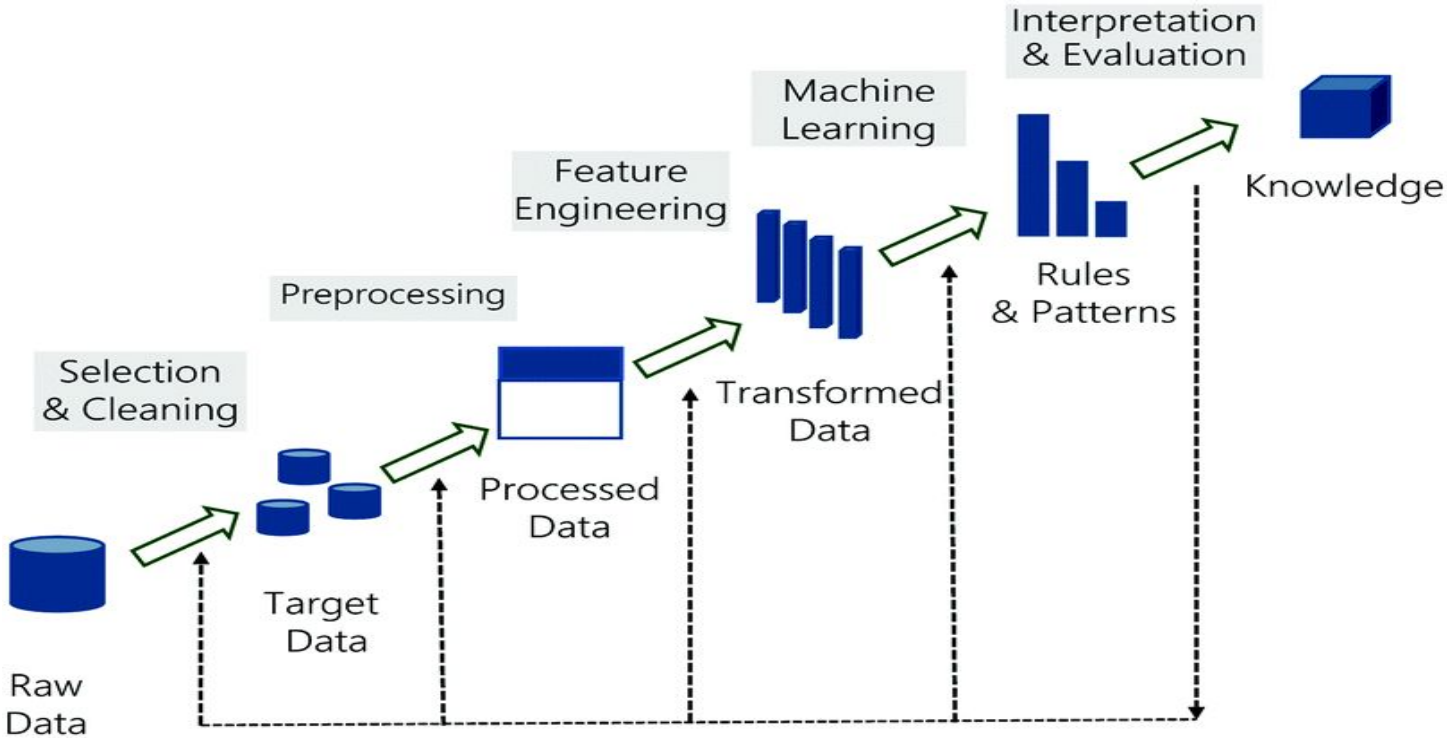
PROPOSED SYSTEM

Cyber-physical Systems (CPSs) are vastly used in today's cities critical infrastructure. The cyber part of these systems usually has a network component through which cyber-attacks can be launched. In this paper, we first design an SVM for initial phase and the security point of view we use encryption and decryption to overcome cyber-attack. In the AES method we use henon map a novel image encryption scheme using integer-order and fractional-order chaotic discrete-time systems. The message was encrypted by the AES algorithm before sending it by a chaotic carrier. The application of the proposed scheme for the transmission of an image has given good results. Simulations results confirm the efficiency of the proposed strategy when applied in the image processing

Changes made in the existing system

Existing system	Proposed system
Lyapunov function	Fuzzy algorithm,Svm algorithm, aes and des for encryption and decryption

BLOCK DIAGRAM OF PROPOSED



Modular description

DATA COLLECTION

Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes. The data collection component of research is common to all fields of study including physical and social sciences, humanities, business, etc. While methods vary by discipline, the emphasis on ensuring accurate and honest collection remains the same.

Preprocessing

Pre-processing is a common name for operations with images at the lowest level of abstraction — both input and output are intensity images. These iconic images are of the same kind as the original data captured by the sensor, with an intensity image usually represented by a matrix of image function values (brightnesses). The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g. rotation, scaling, and translation) are classified among pre-processing methods here since similar techniques are used.

AES AND DES

AES AND DES

The AES encryption algorithm describes various transformations to be carried out on data in an array. The first step is to add the data to the array — after which, over several encryption ranges, cypher transformations are repeated. The first transformation in the AES encryption chip is data replacement with a replacement table; the second transformation is data row replacement; and the third one is column mixing. The last transformation is done with a different part of the encryption key on each column. Longer keys need to complete more rounds. The aim of DES (Data Encryption Standard) algorithm is to provide a standard method for protecting commercially sensitive and unclassified data. In the same key for encryption and decryption.

PROPOSED ENCRYPTION ALGORITHM USING HENON MAP

The inputs to the chaotic Henon system are the image to be encrypted and the initial values of Henon map which are treated as a key. In this paper, I denote an image of size m×n. Shuffling of Image Shuffling is useful to disturb the correlation among the adjacent pixel. Shuffling of the image depends upon the number of rows and columns. Here, shuffling of pixel is done in two steps. Step 1: With each iteration, a quadrant is subdivided into sub-quadrants. Step 2: For the kth iteration, if it is odd then shuffling of quadrant is in clockwise direction otherwise anti-clockwise direction

ATTACK SIMULATION

The simulation of attacks shows you how your network and security measures can function against real-world attack scenarios. The method proposed makes s Sophisticated simulation violations and attacks to authenticate and control safety positions. Measure safety threats to identify vulnerable areas and track safety improvements. Recommendations for remedying security gaps and optimizing safety measures. Security specialists with the ability to leverage threat intelligence through life cycles of attacks and recognize occult threat vectors. Proven expertise in developing governance systems, procedures, duties and obligations to conduct threat hunting simulations based on the result. Advanced ability to simulate threats to assess and motivate security programmers to minimize hazards.

IMPLEMENTATION OF AI TO DETECT THE ATTACK

Computational Intelligence is a form of artificial intelligence (CI). Other nature-inspired techniques used in CI include neural networks and fuzzy logic, both of which provide versatile decision-making frameworks for complex environments including cyber-security applications. Artificial intelligence tools can be used to filter out noise and unnecessary data, as well as to help security experts understand the cyber environment and identify unusual behavior.

PLATFORM USED

SOFTWARE:

Operating System : Windows 7/8/10

Language : Python 3.7

Tools : Pandas , Numpy , Scikit , MatplotLib

HARDWARE:

Processor : Intel Core i3

RAM : 4GB

Hard Disk : 1TB

Mother board : Intel

Speed : 3.3GHZ

Clustering_First_View - Jupyter N | Final Project/

http://localhost:8888/notebooks/Final%20Project/Clustering_First_View.ipynb

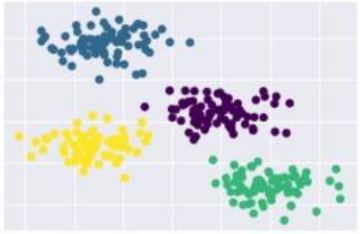
Apps | Overview - CTF 101 | bi0s wiki | Download wallpape... | Satisfactory - Calcul... | Medium – a place t... | Online Penetration... | Online IP Subnet C... | The Hacker News... | Free Cyber Security... | Meet

jupyter Clustering_First_View Last Checkpoint: 7 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code

In [6]:
```
from sklearn.metrics import pairwise_distances_argmin

def find_clusters(X, n_clusters, rseed=2):
    # 1. Randomly choose clusters
    rng = np.random.RandomState(rseed)
    i = rng.permutation(X.shape[0])[:n_clusters]
    centers = X[i]

    while True:
        # 2a. Assign labels based on closest center
        labels = pairwise_distances_argmin(X, centers)

        # 2b. Find new centers from means of points
        new_centers = np.array([X[labels == i].mean(0)
                                for i in range(n_clusters)])

        # 2c. Check for convergence
        if np.all(centers == new_centers):
            break
        centers = new_centers

    return centers, labels

centers, labels = find_clusters(X, 4)
plt.scatter(X[:, 0], X[:, 1], c=labels,
            s=50, cmap='viridis');
```



Type here to search

05:28 PM
2/1/2021

Clustering_First_View - Jupyter N | Final Project/

http://localhost:8888/notebooks/Final%20Project/Clustering_First_View.ipynb

Apps | Overview - CTF 101 | bi0s wiki | Download wallpape... | Satisfactory - Calcul... | Medium – a place t... | Online Penetration... | Online IP Subnet C... | The Hacker News... | Free Cyber Security... | Meet

jupyter Clustering_First_View Last Checkpoint: 6 hours ago (autosaved)

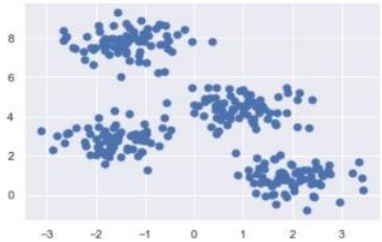
Logout

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3

Run | Code

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()  # for plot styling
import numpy as np
```

In [3]:
```
from sklearn.datasets.samples_generator import make_blobs
X, y_true = make_blobs(n_samples=300, centers=4,
                       cluster_std=0.60, random_state=0)
plt.scatter(X[:, 0], X[:, 1], s=50);
```



In [4]:
```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
```

In [5]:
```
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')
```


Type here to search

05:26 PM
2/1/2021

Clustering_First_View - Jupyter N | Final Project/

http://localhost:8888/notebooks/Final%20Project/Clustering_First_View.ipynb

Apps | Overview - CTF 101 | bi0s wiki | Download wallpape... | Satisfactory - Calcul... | Medium – a place t... | Online Penetration... | Online IP Subnet C... | The Hacker News... | Free Cyber Security... | Meet

jupyter Clustering_First_View Last Checkpoint: 6 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run | Code

```
y_kmeans = kmeans.predict(X)
```

In [5]: `plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')`

Clustering_First_View - Jupyter N | Final Project/

http://localhost:8888/notebooks/Final%20Project/Clustering_First_View.ipynb

Apps | Overview - CTF 101 | bi0s wiki | Download wallpape... | Satisfactory - Calcul... | Medium – a place t... | Online Penetration... | Online IP Subnet C... | The Hacker News... | Free Cyber Security... | Meet

jupyter Clustering_First_View Last Checkpoint: 6 hours ago (autosaved)

Logout

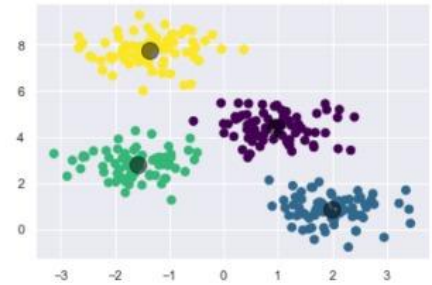
File Edit View Insert Cell Kernel Widgets Help

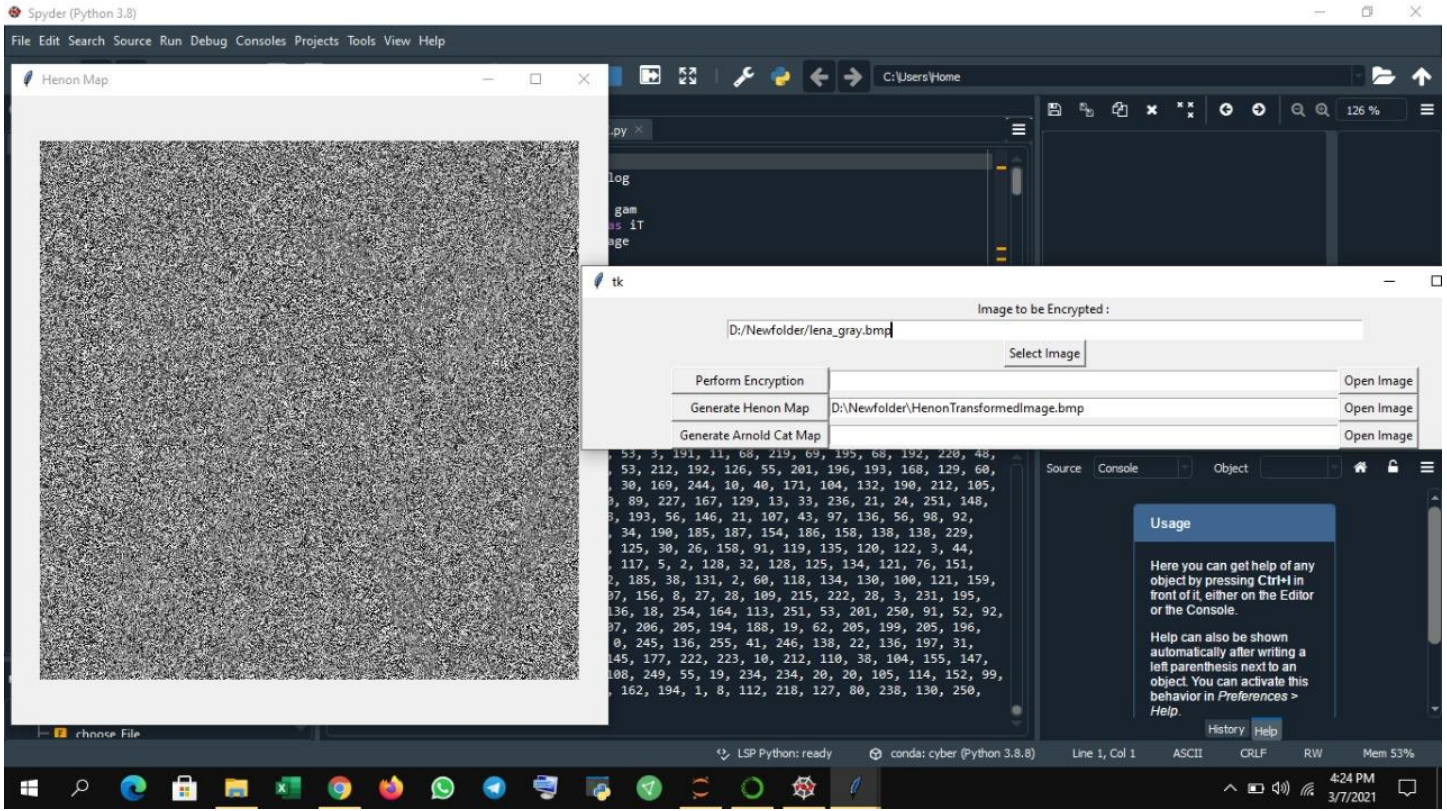
Not Trusted | Python 3

Run | Code

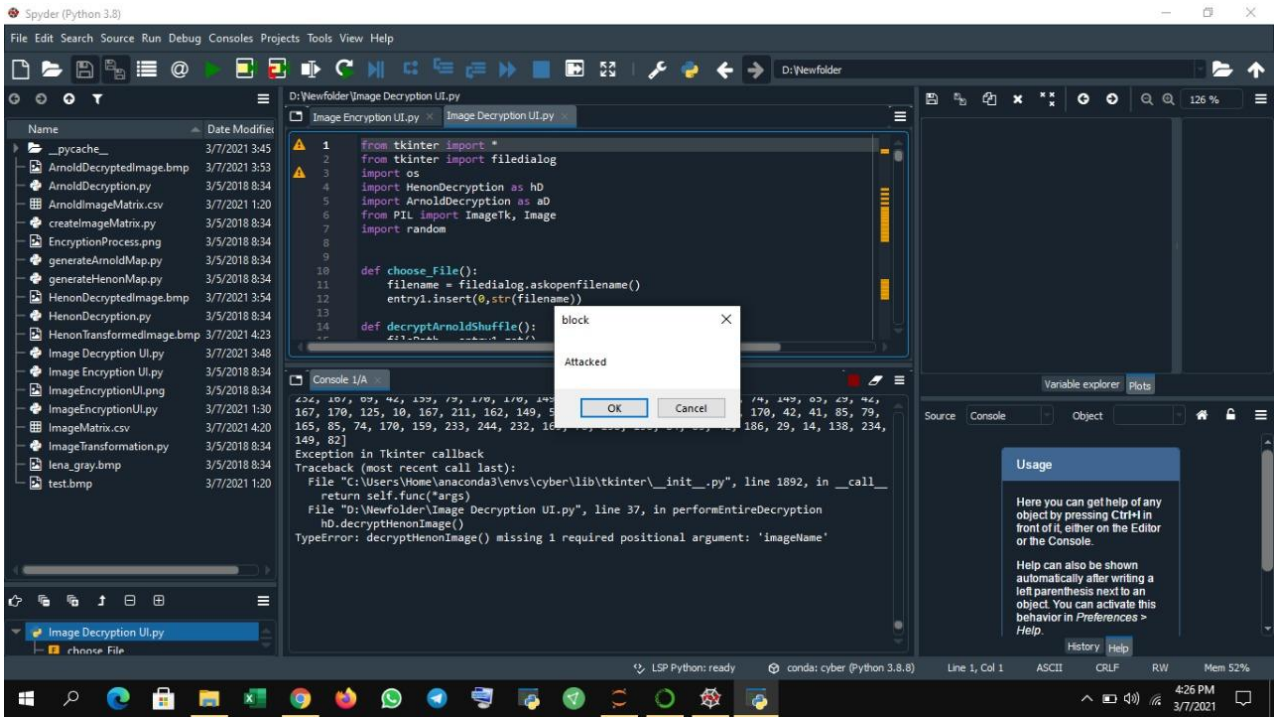
In [4]:
```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
```

In [5]:
```
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
```

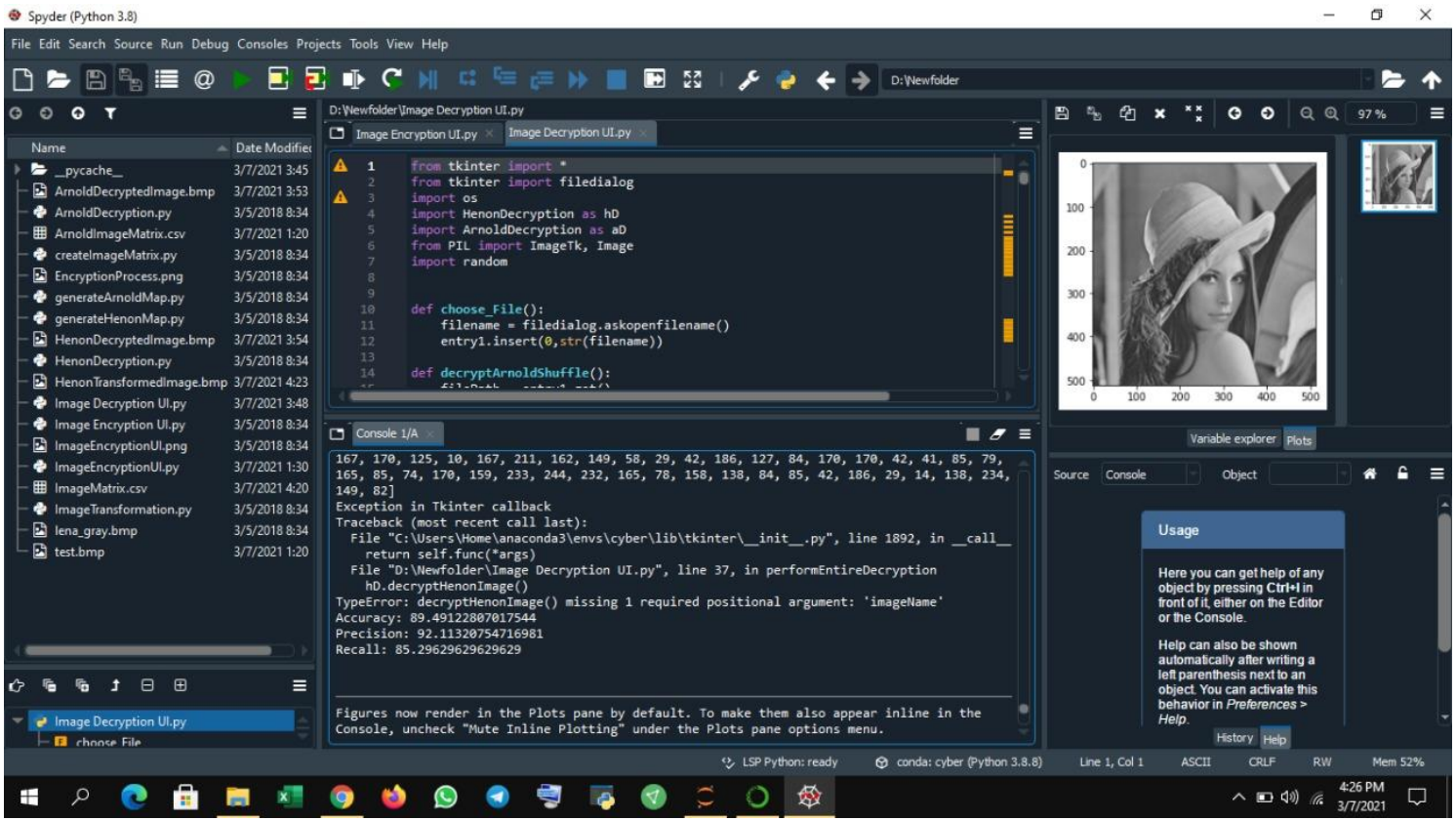




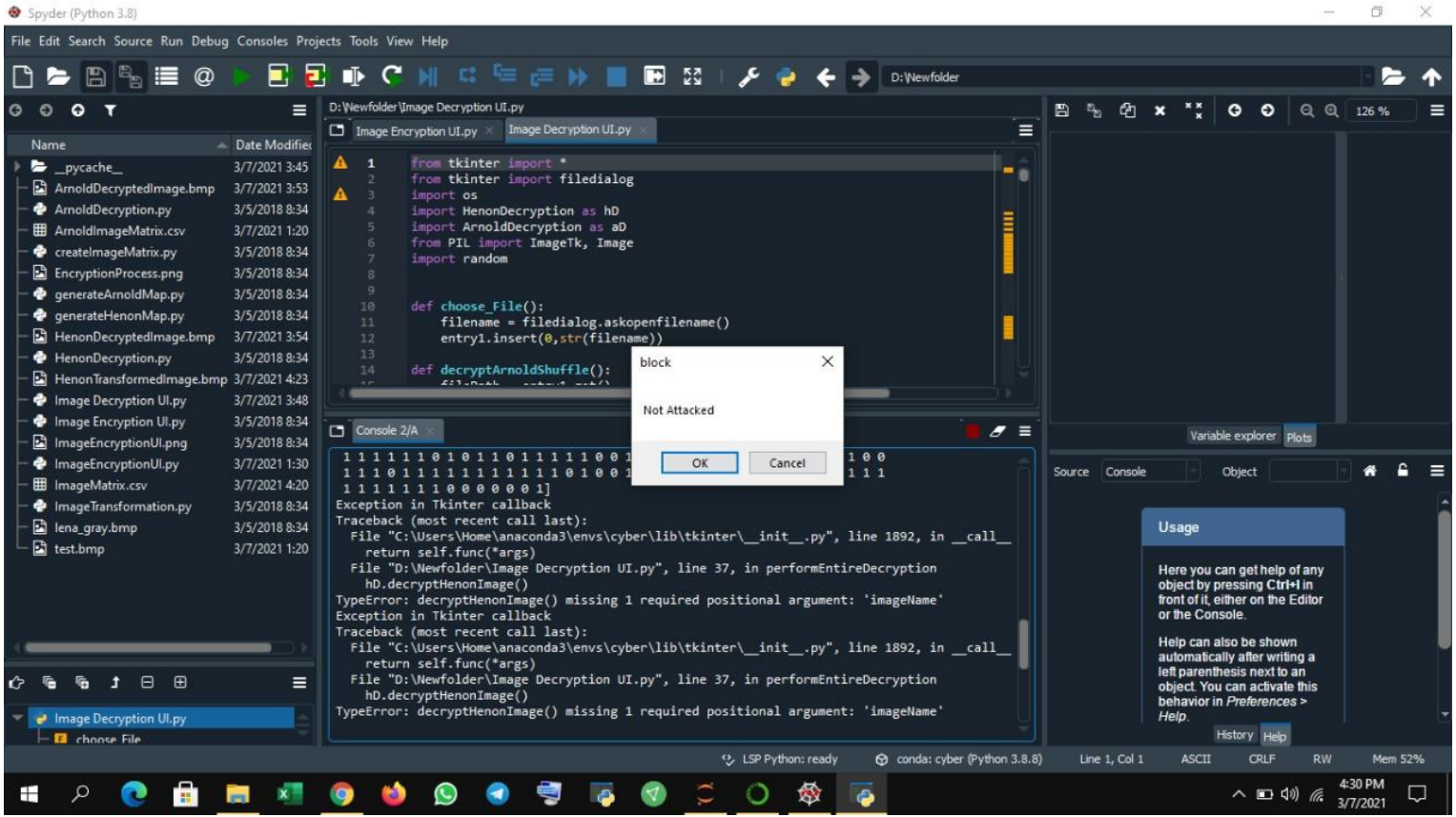
Encryption using Henon map



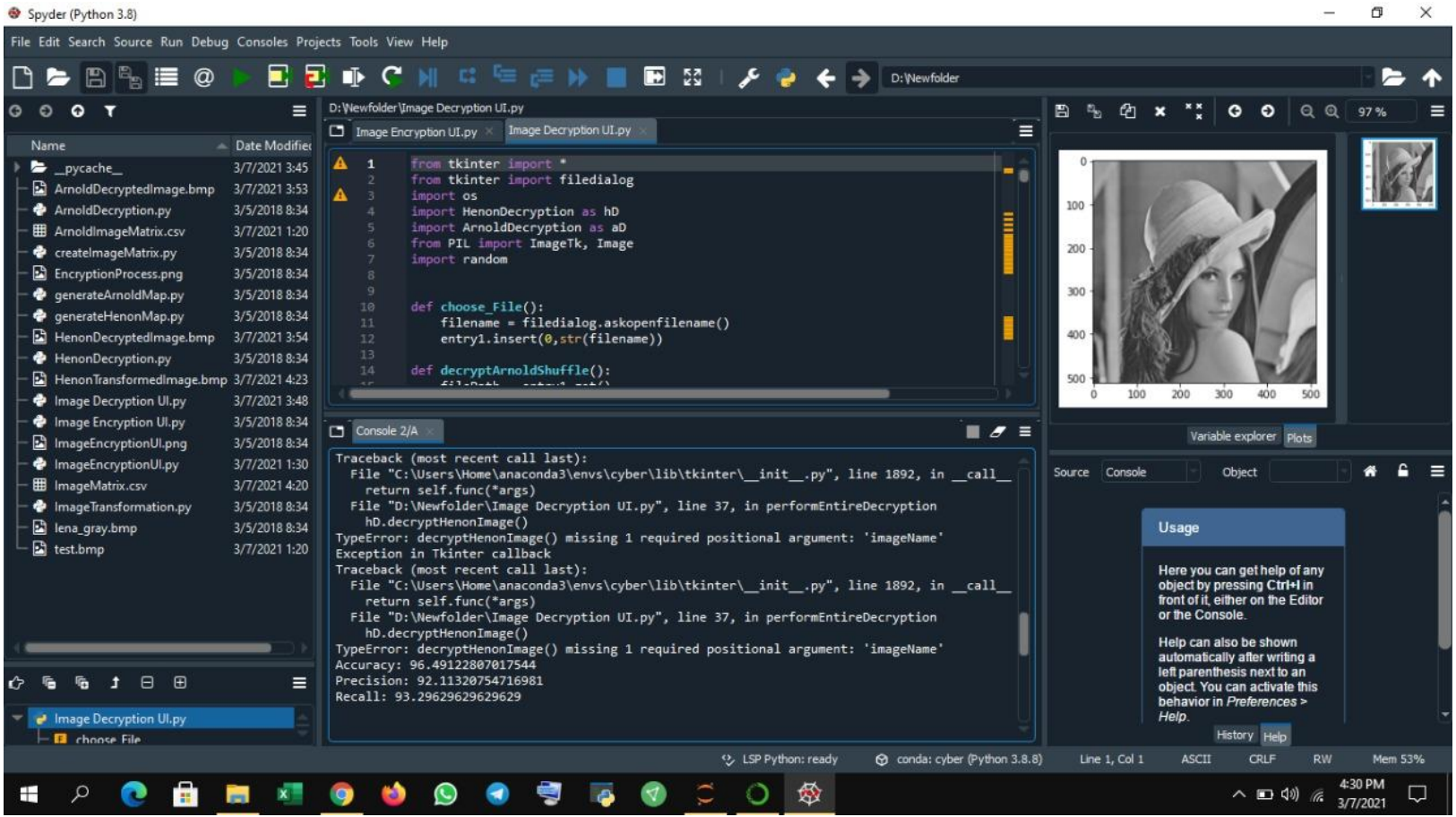
Attack detected


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Newfolder
D:\Newfolder\Image Decryption UI.py
Image Encryption UI.py
Image Decryption UI.py
from tkinter import *
from tkinter import filedialog
import os
import HenonDecryption as hD
import ArnoldDecryption as aD
from PIL import ImageTk, Image
import random
def choose_File():
filename = filedialog.askopenfilename()
entry1.insert(0,str(filename))
def decryptArnoldShuffle():
Name
Date Modified
__pycache__ 3/7/2021 3:45
ArnoldDecryptedImage.bmp 3/7/2021 3:53
ArnoldDecryption.py 3/5/2018 8:34
ArnoldImageMatrix.csv 3/7/2021 1:20
createImageMatrix.py 3/5/2018 8:34
EncryptionProcess.png 3/5/2018 8:34
generateArnoldMap.py 3/5/2018 8:34
generateHenonMap.py 3/5/2018 8:34
HenonDecryptedImage.bmp 3/7/2021 3:54
HenonDecryption.py 3/5/2018 8:34
HenonTransformedImage.bmp 3/7/2021 4:23
Image Decryption UI.py 3/7/2021 3:48
Image Encryption UI.py 3/5/2018 8:34
ImageEncryptionUI.png 3/5/2018 8:34
ImageEncryptionUI.py 3/7/2021 1:30
ImageMatrix.csv 3/7/2021 4:20
ImageTransformation.py 3/5/2018 8:34
lena_gray.bmp 3/5/2018 8:34
test.bmp 3/7/2021 1:20
Image Decryption UI.py
choose_File
Console 1/A
167, 170, 125, 10, 167, 211, 162, 149, 58, 29, 42, 186, 127, 84, 170, 170, 42, 41, 85, 79, 165, 85, 74, 170, 159, 233, 244, 232, 165, 78, 158, 138, 84, 85, 42, 186, 29, 14, 138, 234, 149, 82]
Exception in Tkinter callback
Traceback (most recent call last):
File "C:\Users\Home\anaconda3\envs\cyber\lib\tkinter__init__.py", line 1892, in __call__
return self.func(*args)
File "D:\Newfolder\Image Decryption UI.py", line 37, in performEntireDecryption
hD.decryptHenonImage()
TypeError: decryptHenonImage() missing 1 required positional argument: 'imageName'
Accuracy: 89.49122807017544
Precision: 92.11320754716981
Recall: 85.29629629629629
Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.
97 %
Variable explorer
Plots
Source
Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
History
Help
LSP Python: ready
conda: cyber (Python 3.8.8)
Line 1, Col 1
ASCII
CRLF
RW
Mem 52%
4:26 PM
3/7/2021

Accuracy


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Newfolder
D:\Newfolder\Image Decryption UI.py
Image Encryption UI.py
Image Decryption UI.py
from tkinter import *
from tkinter import filedialog
import os
import HenonDecryption as hD
import ArnoldDecryption as aD
from PIL import ImageTk, Image
import random
def choose_File():
filename = filedialog.askopenfilename()
entry1.insert(0,str(filename))
def decryptArnoldShuffle():
block
Not Attacked
OK
Cancel
Console 2/A
Exception in Tkinter callback
Traceback (most recent call last):
File "C:\Users\Home\anaconda3\envs\cyber\lib\tkinter__init__.py", line 1892, in __call__
return self.func(*args)
File "D:\Newfolder\Image Decryption UI.py", line 37, in performEntireDecryption
hD.decryptHenonImage()
TypeError: decryptHenonImage() missing 1 required positional argument: 'imageName'
Exception in Tkinter callback
Traceback (most recent call last):
File "C:\Users\Home\anaconda3\envs\cyber\lib\tkinter__init__.py", line 1892, in __call__
return self.func(*args)
File "D:\Newfolder\Image Decryption UI.py", line 37, in performEntireDecryption
hD.decryptHenonImage()
TypeError: decryptHenonImage() missing 1 required positional argument: 'imageName'
Variable explorer
Plots
Source
Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
History
Help
LSP Python: ready
conda: cyber (Python 3.8.8)
Line 1, Col 1
ASCII
CRLF
RW
Mem 52%
4:30 PM
3/7/2021

NO ATTACK


Spyder (Python 3.8)
File Edit Search Source Run Debug Consoles Projects Tools View Help
D:\Newfolder
D:\Newfolder\Image Decryption UI.py
Image Encryption UI.py
Image Decryption UI.py
1 from tkinter import *
2 from tkinter import filedialog
3 import os
4 import HenonDecryption as hD
5 import ArnoldDecryption as aD
6 from PIL import ImageTk, Image
7 import random
10 def choose_File():
11 filename = filedialog.askopenfilename()
12 entry1.insert(0,str(filename))
14 def decryptArnoldShuffle():
Console 2/A
Traceback (most recent call last):
File "C:\Users\Home\anaconda3\envs\cyber\lib\tkinter__init__.py", line 1892, in __call__
return self.func(*args)
File "D:\Newfolder\Image Decryption UI.py", line 37, in performEntireDecryption
hD.decryptHenonImage()
TypeError: decryptHenonImage() missing 1 required positional argument: 'imageName'
Exception in Tkinter callback
Traceback (most recent call last):
File "C:\Users\Home\anaconda3\envs\cyber\lib\tkinter__init__.py", line 1892, in __call__
return self.func(*args)
File "D:\Newfolder\Image Decryption UI.py", line 37, in performEntireDecryption
hD.decryptHenonImage()
TypeError: decryptHenonImage() missing 1 required positional argument: 'imageName'
Accuracy: 96.4912280701754
Precision: 92.11320754716981
Recall: 93.29629629629629
Variable explorer
Plots
Source
Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
History
Help
LSP Python: ready
conda: cyber (Python 3.8.8)
Line 1, Col 1
ASCII
CRLF
RW
Mem 53%
4:30 PM
3/7/2021

Accuracy

CONCLUSION

In this paper, we first design an, FUZZY clustering and SVM for initial phase and the security point of view we use encryption and decryption to overcome cyber-attack. In the AES method we use henon map a novel image encryption scheme using integer-order and fractional-order chaotic discrete-time systems. The message was encrypted by the AES algorithm before sending it by a chaotic carrier. The application of the proposed scheme for the transmission of an image has given good results. Simulations results confirm the efficiency of the proposed strategy when applied in the image processing. The proposed method was applied to a test image and results thus obtained proved a higher level of security of images. Confusion has been done by pixel movement form actual position to a new position and diffusion has been done through byte sequence generated through Henon map. So both the processes of increasing confusion and diffusion resulted in increasing the security of cryptosystem