CHAPTER-1
INTRODUCTION

1.1 Introduction

Dams and power plants are becoming more IoT-connected. Industrial Control Systems (ICS) typically include IIoT devices. which ensures infrastructure safety. PLC, DCS, and SCADA systems are part of ICS.Connecting ICS/IIoT equipment to public networks increases attack surfaces and intrusion risk. In 2010, the Stuxnet campaign appeared to have targeted Iranian nuclear enrichment centrifuges,severely damaged the machinery, is one well-known example [1], [2]. Another example is the pump-related event that led to the collapse of an Illinois water facility in 2011 [3].

BlackEnergy3 caused about 230,000 power outages in 2015 [4]. Cyberattacks against three U.S. gas pipeline companies in April 2018 disabled their electronic customer communication systems [1].ICS security solutions may not immediately apply to IT and OT systems, despite the fact that they are rather established security solutions for those systems. This could be the case, for instance, since the cyber systems and the controlled physical environment are so closely integrated. In order to analyse physical behaviour and preserve system operating availability, system-level security techniques are required [1].

ICS security objectives emphasise availability, integrity, and secrecy, unlike other IT/OT systems [5]. The feedback control loop's intimate link between physical processes and variables makes ICS hacks dangerousand even catastrophic repercussions for society and the environment. To identify and prevent ICS attacks, comprehensive safety and security measures are essential. [1]. For detecting and attributing attacks, signature- and anomaly-based methods are frequently employed. Attempts have been made to develop hybrid-based systems to mitigate signature-based and anomaly-based attribution drawbacks [6].

Hybrid-based approaches can detect anomalous activity, but they are unreliable since networks are continually updated, IDS typologies are numerous [7]. Traditional attack detection and attribution methods assess IP addresses, transmission ports, traffic volume, and packet intervals.

ML and DNN-based assault detection and attribution methods have garnered attention recently. Additionally, network-based and host-based attack detection methodologies may be classified.

Network traffic attack detection employs supervised clustering, single-class or multi-class SVM, fuzzy logic, ANN, and DNN. Real-time traffic data analysis detects malicious assaults. Detection of attacks that only considers network and host data. however, may miss more complex assaults or insider attacks.

Since unsupervised models don't need in-depth knowledge of the cyber-threats, they may supplement system monitoring by including process/physical data. A nation state advanced persistent threat actor with enough time and information can usually bypass strong protective mechanisms. Additionally, the majority of current techniques represent just a system's usual behaviour and record departures from that behaviour as anomalies, ignoring the unbalanced nature of ICS data.

This may be because there aren't many attack examples in real-world circumstances and datasets currently available. Although employing majority class samples is an excellent way to prevent problems brought on by unbalanced datasetsThe trained model won't see attack sample patterns. Thus, it produces many false positives and misses unsuspected threats [8].

As a result, people have tried to model large ideas from smaller ones [9] without using human-made features [10] by employing DL techniquesautomated feature (representation) learning. Our two-stage, ensemble deep learning-based approach to attack detection and attribution for imbalanced ICS datasets was motivated by these findings.

Assaults in non-uniform settings can be identified using a combination of ensemble representation learning and Decision Tree (DT) analysis. In the second step, when an attack has been identified, many one-vs-all classifiers will ensemble to build a bigger DNN and identify attack features with a confidence interval. The framework also finds new attack samples. Here is an overview of how we conducted this study:

1) Novel two-phase ensemble ICS attack detection may detect known and unknown attacks. The proposed method surpasses existing methods f-measure and accuracy. Deep representation learning prevents imbalanced data.

2) Unique self-tuning two-phase assault attribution decreases false alarms technique based on a DNN architecture that combines several deep one-vsall classifiers. Using the recommended approach, attacks with high similarity may be accurately ascribed. At the time of this research, ICS/IIoT's first ML-based attack attribution method.

3) The suggested architecture for attack detection and attribution performs better than previous DNN-based strategies, although its computational complexity is comparable.

1.2 Related Work

Dams and power plants are becoming more IoT-connected. Industrial Control Systems (ICS) typically include IIoT devices. which ensures infrastructure safety. PLC, DCS, and SCADA systems are part of ICS.Connecting ICS/IIoT equipment to public networks increases attack surfaces and intrusion risk. In 2010, the Stuxnet campaign appeared to have targeted Iranian nuclear enrichment centrifuges,severely damaged the machinery, is one well-known example [1], [2]. Another example is the pump-related event that led to the collapse of an Illinois water facility in 2011 [3].

BlackEnergy3 caused about 230,000 power outages in 2015 [4]. Cyberattacks against three U.S. gas pipeline companies in April 2018 disabled their electronic customer communication systems [1].

ICS security solutions may not immediately apply to IT and OT systems, despite the fact that they are rather established security solutions for those systems. This could be the case, for instance, since the cyber systems and the controlled physical environment are so closely integrated. In order to analyse physical behaviour and preserve system operating availability, system-level security techniques are required [1].

ICS security objectives emphasise availability, integrity, and secrecy, unlike other IT/OT systems [5]. The feedback control loop's intimate link between physical processes and

variables makes ICS hacks dangerousand even catastrophic repercussions for society and the environment.

To identify and prevent ICS attacks, comprehensive safety and security measures are essential. [1]. For detecting and attributing attacks, signature- and anomaly-based methods are frequently employed. Attempts have been made to develop hybrid-based systems to mitigate signature-based and anomaly-based attribution drawbacks [6].

Hybrid-based approaches can detect anomalous activity, but they are unreliable since networks are continually updated, IDS typologies are numerous [7]. Traditional attack detection and attribution methods assess IP addresses, transmission ports, traffic volume, and packet intervals.

ML and DNN-based assault detection and attribution methods have garnered attention recently. Additionally, network-based and host-based attack detection methodologies may be classified.

Network traffic attack detection employs supervised clustering, single-class or multi-class SVM, fuzzy logic, ANN, and DNN. Real-time traffic data analysis detects malicious assaults. Detection of attacks that only considers network and host data. however, may miss more complex assaults or insider attacks.

Since unsupervised models don't need in-depth knowledge of the cyber-threats, they may supplement system monitoring by including process/physical data. A nation state advanced persistent threat actor with enough time and information can usually bypass strong protective mechanisms. Additionally, the majority of current techniques represent just a system's usual behaviour and record departures from that behaviour as anomalies, ignoring the unbalanced nature of ICS data.

This may be because there aren't many attack examples in real-world circumstances and datasets currently available. Although employing majority class samples is an excellent way to prevent problems brought on by unbalanced datasetsThe trained model won't see attack sample patterns. Thus, it produces many false positives and misses unsuspected threats [8].

As a result, people have tried to model large ideas from smaller ones [9] without using human-made features [10] by employing DL techniquesautomated feature (representation) learning. Our two-stage, ensemble deep learning-based approach to attack detection and attribution for imbalanced ICS datasets was motivated by these findings.

Assaults in non-uniform settings can be identified using a combination of ensemble representation learning and Decision Tree (DT) analysis. In the second step, when an attack has been identified, many one-vs-all classifiers will ensemble to build a bigger DNN and identify attack features with a confidence interval. The framework also finds new attack samples. Here is an overview of how we conducted this study:

1) Novel two-phase ensemble ICS attack detection may detect known and unknown attacks. The proposed method surpasses existing methods f-measure and accuracy. Deep representation learning prevents imbalanced data.

2) Unique self-tuning two-phase assault attribution decreases false alarms technique based on a DNN architecture that combines several deep one-vsall classifiers. Using the recommended approach, attacks with high similarity may be accurately ascribed. At the time of this research, ICS/IIoT's first ML-based attack attribution method.

3) The suggested architecture for attack detection and attribution performs better than previous DNN-based strategies, although its computational complexity is comparable.

CHAPTER-2

LITERATURE SURVEY

[1] F. Zhang, J. W. Hines, H. A. D. E. Kodituwakku, and J. Coble, "Multilayer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System, and Process Data," IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp. 4362-4369, 2019.

Concerns concerning industrial control system (ICS) cybersecurity have grown in recent years as cyber-physical system threats have intensified. Assaults are detected using an unsupervised ensemble of normal and assault representations. It assigns samples to a two-part DNN using an ensemble of numerous one-vs-all classifiers trained on each attack attribute.

Defense-in-depth cyber-attack detection using network traffic, host system data, and process characteristics increases ICS cybersecurity. Assault detection has several defences. allowing the defenders critical time before the physical system is irreparably damaged. The suggested detection method uses real-time ICS testbed data. Five cyberattack simulations—MITM, DoS, data exfiltration, manipulation, and fake data injection—feed data-driven detection algorithms.

KNN, decision tree, bootstrap aggregating (bagging), and random forest (RF) are common classification modelsare examined as cyber-attack defences if the intrusion prevention layer fails.. These models use host and network data. According to intrusion detection research, KNN, bagging, and RF identify MITM and DoS assaults accurately and reliably with low rates of missed alerts and false alarms.

Data monitoring on the network and host system might reveal command manipulation and fraudulent data injection from inside the system. To identify potential threats early on, auto-associative kernel regression is employed. The findings indicate that this technology can detect cyberattacks with physical consequences before they have a significant effectA data-driven, multi-layer cyber-attack detection system using network, system, and process data could secure an ICS.

[2] "Stealthy Attack against Redundant Controller Architecture of Industrial CyberPhysical System," by R. Q. Wang, Ma, P. Cheng, Z. Zhang, W. Liu, and Q. In Wei, appeared in IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9783-9793, in 2019.

The controller of an industrial Cyber-Physical System is essential for ensuring dependability and stability. As a result, redundant controller architecture has been employed effectively in DCS, SCADA, and other typical industrial Cyber-Physical Systems. They keep an eye on and manage crucial industrial processes including those involved in power production, the chemical sector, water treatment facilities, etc.

Unpredictable mechanical failures prompted the invention and widespread use of redundant controller architecture. However, the original structure proposed for assuring dependability and security may increase the cyber-attack surface, thereby increasing the likelihood that an attacker will employ this architecture for covert attacks. In this study, we investigate the vulnerabilty of redundant controller architecture and propose a strategy for a covert combined attack against systems employing redundant controller architecture.

We execute a unified attack against these real-world devices from three manufacturers, each of which has a number of zero-day vulnerabilities. Our test findings on numerous real-world device We show that the redundant controller design can secretly exploit any system we evaluated. We also provide recommendations for reducing this risk.

[3] E. Nakashima, "Experts claim foreign hackers attacked U.S. water plant." Hackers remotely sabotaged an Illinois water pump last week, triggering a federal probe.

Joe Weiss, a well-known authority on defending infrastructure from cyber assaults, According to the report obtained by Weiss, A software provider for industrial equipment gave the attackers credentials to a rural water utility west of Springfield. It made no mention of the assailants' motivation. Using credentials obtained from the same software manufacturer, he said, the same organisation may have targeted other industrial targets or be preparing attacks.

FBI and DHSare investigating the matter, according to DHS spokeswoman Peter Boogaard. He refused to elaborate, but said that as of yet"No reliable, verifiable evidence

suggests a danger to key infrastructure or public safety." An Illinois FBI spokeswoman declined comment.

[4] G. C. Caldera, Falco, and H. Shrobe, IEEE Internet of Things Journal, vol. 1, no. 2, "IIoT Cybersecurity Risk Modelling for SCADA Systems," 5, no. 6, pp. 4486–4495, 2018.

Electric grids, water networks, and transportation systems are some examples of urban vital infrastructure that is often the target of cyberattacks. IoT, which makes up these systems, is a network of interconnected objects. Urban IIoT critical infrastructure attacks would significantly disrupt civilization. Urban critical infrastructure is often controlled by SCADA systems.

There is no data-driven approach for assessing the cyber risk to urban vital infrastructure, despite the obvious necessity for such an assessment. Using cosine similarity tests, we analyse SCADA and non-SCADA systems in this research and find that SCADA, as a software subclass, has distinct risk characteristics for the IIoT. We then dispute the widely held belief that the SCADA software subclass is not vulnerable to attack, as claimed by the typical vulnerability score system risk criteria of exploitability and effect.

Several statistical methods are used to calculate SCADA risk metrics that can be used to predict SCADA vulnerability exploitation. Our research led us to develop a modifiable SCADA risk prioritisation schema that the security community may utilise to comprehend SCADA-specific risk. A data-driven prioritisation schema will assist researchers in identifying security flaws particular to this software subclass that is vital to the functioning of our civilization, taking into account the distinctive characteristics of SCADA systems.

[5] J. Yang, S. Yang, H. Xu, C. Zhou, and B. In Hu's article "Anomaly Detection Based on Zone Partition for Security Protection of Industrial Cyber-Physical Systems," published in IEEE Transactions on Industrial Electronics, vol. 65, no. 5, pp. 4257–4267, 2018.

Security risks to industrial control systems (ICSs) are becoming worse. Zone isolation, a widely used concept for halting the spread of attacks in general information systems, ICS security is being explored. Generally, perimeter security measures are employed for this purpose. Inter-zone communication can spread abnormal physical processes in a damaged

field zone to nearby zones. Because physical processes in numerous zones are interconnected, ICS attack effect spreads easily.

This paper presents an SDSec solution. Network communications and physical process conditions are examined by the hybrid anomaly detection module. The multi-level security response module aids in isolating any compromised zone by preventing unauthorised transmissions from communicating. To safeguard physical processes, it designs assault mitigation techniques. Hardware-in-the-loop simulations show the method's efficiency.

[6] S. Ponomarev and T. Atkison, "Telemetry-based industrial control system network intrusion detection," IEEE Transactions on Dependable and Secure Computing.

ICSs used "air-gap" security to physically isolate each node from other networks like the Internet. Companies and engineers that utilise ICS networks gain by connecting them to the Internet. The protocols employed by ICSs, on the other hand, offer little to no security measures and are susceptible to a variety of assaults due to their air-gapped security design.

The method suggested in this research uses network telemetry, which is data that is transferred across the network but is not necessarily utilised by the transmission protocol to identify intrusions into network-attached ICSs. The created IDS was able to distinguish between computers belonging to an attacker and an engineer on the same network with 94.3 percent accuracy while using simulated PLC units, and over the Internet with 99.5 percent accuracy.

[7] J. F. Clemente's doctoral dissertation, "No cyber security for critical energy infrastructure," was published by the Naval Postgraduate School in 2018.

Given that it provides energy to all of the country's crucial infrastructures, A massive cyberattack on the US power infrastructure is an obvious target. Attackers often target bulk power system weaknesses and threaten energy delivery. Network security and defence protect digital data and system controls in our connected world, from government infrastructure to online banking. Concerns that computer network hacking and other security-related assaults might compromise the nation's critical infrastructure have drawn attention to cyber threats.

A stable, functioning vital infrastructure is required to ensure the United States' national and economic security. Understanding the effects of a major power outage could change cyber security on our most essential asset, national power. This study examines power grid network resilience, public-private collaboration in fighting cyber threats, and project management, access controls, application software development, system software, and service continuity controls are all areas that need to be tightened up.

[8] Japkowicz, C. S. Sharma, Bellinger, and N. "One-class versus binary classification: Which and when?" in the proceedings of the 2011 IEEE International Conference on Machine Learning and Applications, vol. 2, 2012, pp. 102-106.

In the machine learning field, binary classifiers have usually been the standard for creating classification models. One-class classification, on the other hand, seeks to create models utilising just a single class of data as an alternative to binary classification. This is especially helpful when there are too many data points of a certain class. Binary classifiers may not perform well in such unbalanced situations, in which case one-class classifiers are a more practical choice. In this study, we examine how binary and one-class classifiers perform as the degree of imbalance and, therefore, uncertainty in the second class, grow.

Our goal is to learn which categorization paradigm is more appropriate when imbalance and uncertainty rise. In order to do this, we run tests on a variety of fake and real datasets from the UCI repository and track how well the binary and one-class classifiers perform as the size of the second class rapidly shrinks, therefore escalating the amount of imbalance. The findings demonstrate that although one-class classifiers exhibit relatively consistent performance as the amount of imbalance rises, binary classifiers perform worse.

[10] Y. A. Courville, YoshuaBengio, and P. Vincent, "Representation learning: Review and new perspectives,"

We hypothesise that this is due to the fact that different representations might entangle and, to a greater or lesser extent, conceal the many explanatory variables of variation hidden behind the data. Data representation is often regarded to be a need for machine learning algorithms to succeed. Even while generic priors can help design representations, AI is improving representation-learning algorithms to use them.

Probabilistic models, autoencoders, manifold learning, deep networks, and unsupervised feature learning are contrasted. Learning effective representations, calculating representations (inference), representation learning, density estimation, and manifold learning geometric linkages remain unsolved.

[11] M. L. Gupta, K. M. Khan, Zolanvari, M. A. Teixeira, and R. IEEE Internet of Things Journal, vol. Jain, "Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things," 6, no. 4, pp. 6822–6834, 2019.

IIoT devices must be secured to prevent catastrophic attacks. ML and big data analytics can investigate and secure IoT technologies. These methods may improve IIoT security. This post will begin by discussing prominent IIoT protocols and the vulnerabilities that come with them.

We then evaluate cyber-vulnerabilities and describe how machine learning might minimise them. Following this is a review of existing machine learning (ML)-based intrusion detection algorithms. Our final topic of discussion is our case study, which consists of information about a realistic testbed we constructed to execute cyberattacks and design IDS. We tested a machine learning (ML)-based anomaly detection solution for backdoor, command injection, and SQL injection threats. To gain an accurate understanding of the effectiveness of the strategies, we analysed their performance using representative metrics.

[12] I. Z. U. Khan, Y. Hussain, A. Khan, D. Pi, and A. IEEE Access, vol. Nawaz, "HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," 7, 2019.

SCADA systems are utilised to manage and monitor vital infrastructures such as gas distribution and chemical processing facilities and energy production and distribution networks. Since intrusion detection has been a prominent research issue for years, numerous ICS and cyber-physical system intrusion detection solutions have been presented. Seismic net, duqu, and flame against ICS attack viruses have lately seriously destroyed critical infrastructure,including nuclear power plants, in various nations.

These more severe breaches have raised concerns about the security of the ICS in several nations. Building an intrusion detection system presents a problem when dealing with imbalanced intrusion datasets, or when one class is represented by a smaller number of instances (minority class). In order to address this problem, we present a strategy and suggest an ICS anomaly detection mechanism. Our proposed solution employs a hybrid model that exploits the predictable and regular communication patterns among ground devices in ICS situations. To scale and standardise the data, we initially utilised numerous preprocessing approaches.

SecondDimensionality reduction techniques are utilised to enhance the anomaly detection process. To balance the dataset, we applied an updated nearest-neighbor rule approachFourth, After watching the system without irregularities, the Bloom filter creates a signature database. Finally, our package content-level detection and instance-based learner found additional dangers.. HML-IDS beats benchmark models by 97% on a gas pipeline SCADA system dataset.

[13] T. S. Adepu, K. Das, and J. Computers & Security, vol. Zhou, "Logical data analysis for industrial control system anomaly detection," 96, p. 101935, 2020.

Over the last 10 years, there has been a lot of research done on ICS attacks. Malicious changes to ICS may manifest itself in a variety of ways, like ICS data or network traffic changes. Even though various heuristics and machine learning approaches may spot anomalies in ICS data, no research has employed TIA Portal data. TIA Portal, a popular programme, the ICS can be organised and configuration and programming data may be examined, modified, and removed. It is possible to recover previous iterations of the present system settings by preserving the single project datasets historically.

In this early effort, we provide heuristics for TIA Portal data anomaly detection. By looking into long-term backups, we specifically analyse the historical adjustments made to the TIA Portal data. Our method protects against harmful alterations performed by staff members who have direct access to the machines as well as changes to the data brought on by infiltrating attackers. As a result, we began to look at actual TIA Portal project data from a manufacturing line of an automobile company, three years of historical data for irregularities.

[14] J. Q. Yu, Y. Hou, and V. O. K. Li, "Article: "Online False Data Injection Attack Detection Using Wavelet Transform and Deep Neural Networks," IEEE Transactions on Industrial Informatics, Volume 14, Issue 7, Pages: 3271-3280, 2018.

Modern power systems require state estimation, False data injection attacks may disrupt grid operations. AC attack detection is important since electrical companies employ AC state estimation increasingly. We describe a neural network attack detection method for AC systems.

Malicious data in state vectors may change geographical and temporal data correlations. Wavelet transform and deep neural network methods analyse temporally successive predicted system states to correctly capture such disagreement. IEEE 118- and 300-bus power system case studies assess the method. The approach accurately detects attacks. We also test the control parameters' sensitivity.

[15] Gidlund, "Reference: "A machine learning-based technique for detecting false data injection attacks in industrial IoT," IEEE Transactions on Industrial Internet of Things, vol. A. M. H. Eldefrawy, K. G. Seddik, M. Y. Gadallah, M. N. Aboelwafa, M. Y.

The increasing deployment of IIoT has caused many security issues. The IIoT's "False Data Injection" (FDI) attack is a big security risk. FDI assaults fake sensor signals to trick industrial platforms. Traditional threat detection systems have been successfully defeated by FDI attacks.

We provide a novel Autoencoder (AE)-based approach for FDI attack detection in this work. We make advantage of sensor data's temporal and geographical correlation, which may be exploited to detect bogus data. Furthermore, AEs are used to denoise the produced data. Performance testing shows that our solution is successful in detecting FDI attacks. Furthermore, it outperforms a comparable approach based on a support vector machine (SVM). The denoising AE data cleaning method's capacity to recover clean data from damaged (attacked) data is also shown to be fairly powerful.

CHAPTER-3

PROPOSED METHOD

The suggested framework's architecture is shown in Figure 1. By using ensembled unsupervised DNNs and a decision tree to analyse the ICS input characteristics, the attack detection approach in this framework finds attacks. The sample is sent to numerous DNNs for in-depth examination if an attack is found.

The unseen attack detection module would identify the assaults as unseen if they had never been observed or known before. This will be distributed for thorough security review. If not, the attack attribution technique finds the assault attribute.

A. Proposed Ensemble Attack Detection Method

Attack detection and representation learning. Unsupervised DNNs on unbalanced datasets learnt majority class patterns and ignored minority class characteristics. Most academics try adding or eliminating samples before feeding the dataset to a DNN. ICS/IIoT security programmes cannot generate or remove samples. ICS/IIoT systems are delicate. attack samples must be evaluated in a real network, which is difficult because they may damage the network and harm the environment or humans.

Additionally, verifying the generated samples takes time. Typically, less than 10% of ICS/IIoT datasets contain assault samples; Thus, deleting 80% of the dataset removes most of its data. Normal data should not be removed.

This work created a revolutionary deep representation learning technique that lets the DNN manage unbalanced datasets without changing. adding, or removing samples to overcome the aforementioned problems. Two unsupervised stacked autoencoders identified class patterns in this model. Each model extracted abstract patterns from one class while ignoring another, so its output matched its inputs.

There are three levels of input and output representations in a stacked autoencoder's decoders and encoders. Input representations were transformed into representations in dimensions of 8, 4, and 16 by the encoder layers.

The encoding function of an autoencoder is shown by Equation 1. By mapping the 16-dimensional new representation to the 400-, 800-, and input-dimensional spaces, the decoder layers effectively created a copy of the input representation. Equation 2 depicts an autoencoder's decoder function. Through trial and error, these hyperparameters were chosen to have the greatest f-measure performance and the least amount of architectural complexity.

$$h_i = \sigma(w_i x_i + b_i) \qquad (1)$$

In the equation above, represents an activation function, w the encoder's weight matrix, x, b, h, and i "Normal, Attack."

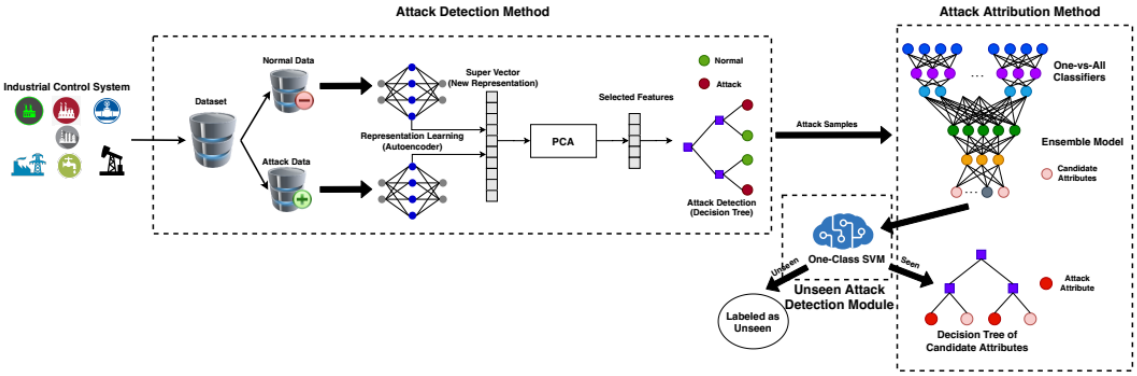


Fig. 1. Proposed attack detection and attribution framework

$$\hat{x}_i = \sigma'(w_i' h_i + b_i') \qquad (2)$$

In the equation above, i is "Normal, Attack," w 0 is the decoder's weight matrix, where h is the encoded representation, b0 is the decoder's bias, x is input x's reconstruction, and 0 is the activation function.Individual autoencoders were trained using the loss function shown in Equation 3 using this method.

$$\mathcal{L}(x, \hat{x}) = ||x - \hat{x}||^2 = ||x - \sigma'(w'(wx + b) + b')||^2 \qquad (3)$$

The loss between input and reconstruction x is L(x, x). All observations were put through both autoencoders then concatenated to create a new dataset.

$$X_{new} = [H_{normal}, H_{attack}] \qquad (4)$$

The new dataset, Xnew, is a supervector comprising each sample's normal and attack autoencoder model representations. The Hattack and Hnormalhattack and hnormal matrices show how sample x may be an attack or normal sample, respectively.

In the second stage, PCA calculated the hybrid representation of the super-vector [21], and the recovered features were detected by a DT classifier. The fifth and sixth equations show how PCA speeds up DT classifier training and testing.

For limited feature sets, DT is a robust, basic model that trains quicker than DNNs. DT also performs well with ICS and CPS data, as demonstrated by our earlier tests [22] and a few other investigations [11]. The DT was trained using the Gini function (see equation 7).

$$X_{new}^{T} X_{new} = PDP^{-1} \qquad (5)$$

In the equation above, eigenvalues are allocated to the major diagonal of the diagonal eigenvalue matrix D, and all other values are treated as zero. P is the eigenvector matrix. After sorting the eigenvectors according to their eigenvalues, P was the first k vectors. Equation 6 shows how to collect k features from dataset Xnew.

$$X^{*} = X_{new} P^{*} \qquad (6)$$

PCA dimension reduction yields X in the equation.

$$gini = 1 - \sum_{i=1}^{c} p_i^2 \qquad (7)$$

c is the number of classes, and pi is the probability of class i in the branch under examination. OCSVMs surrounded regular data and reported the rest as previously detected assaults. The suggested attack detection component's approach is shown in approach 1 of this document.

Algorithm 1: The proposed two-phase attack detection component

Data: Dataset including *Normal* and *Attack* samples (X) and their labels$(y = \{0, 1\})$

Training Phase:

$X = z(X)$: $z = \dfrac{x - min(x)}{max(x) - min(x)}$;

$X_{attack} = X[y == 1]$;

$X_{normal} = X[y == 0]$;

♯ Training Representation Learning Models:

for *number of epochs* **do**

  for *number of batches in Normal set* **do**

    Train the Normal autoencoder (AE_{normal}):

    $\min \mathcal{L}(X_{normal}, \hat{X}_{normal})$;

  end

  for *number of batches in Attack set* **do**

    Train the Attack autoencoder (AE_{attack}):

    $\min \mathcal{L}(X_{attack}, \hat{X}_{attack})$;

  end

end

♯ Fusion Layer:

$newRep_{normal} = AE_{normal}.predict(X)$;

$newRep_{attak} = AE_{attack}.predict(X)$;

$X_{superVector} = concat(newRep_{normal}, newRep_{attack})$;

♯ Detection Model:

Feature selection using PCA:

$Selected\_Features(X_{superVector}) = PCA(X_{superVector})$;

Train a DT using the new features:

$DT = Train\_DT(Selected\_Features)$

Testing Phase:

$x_{test} = z(x_{text})$;

$newRep_{normal} = AE_{normal}(x_{test})$;

$newRep_{attack} = AE_{attack}(x_{test})$;

$superVector = concat(newRep_{normal}, newRep_{attack})$;

$\hat{x}_{test} = Selected\_Features(superVector)$;

$\hat{y} = DT(\hat{x}_{test})$;

Output: Normal/Attack Label $(\hat{y})$

B. Proposed Self-Tuning Attack Attribution Method

There are two parts to the suggested self-tuning assault attribution mechanism. A one-vs-all classifier is learned for each characteristic in the first stage. Attack samples are divided by attribute and a DNN model is generated for each subset to train classifiers. Sigmoid and ReLU functions activate the output and buried layers, respectively.

One-vs-all DNNs use first-phase outputs in the second phase. In the second stage, one-vs-all classifiers and a DNN ensemble model improve the DNN. This DNN has many one-vs-all classifiers and a totally connected element that creates multiple classes using sample characteristics and the results of the first component.

The ensemble DNN's hidden layers use ReLU, whereas the output activation function uses softmax (equation 8). Equation 9's DNN loss function is categorical cross-entropy (CE). DNN results likely explain the sample. Attack attribution relies on this concept. Secondary attack attribution trains DT classifiers for each pair of attack characteristics needed for the final attack attribution from two alternatives.

$$\sigma(s)_i = \frac{e^{s_i}}{\sum_{j=1}^{K} e^{s_j}} \qquad (8)$$

where K is the number of classes, and z = (z1, ..., zk) ∈ R K.

$$CE = -\sum_{i=1}^{K} y_i \log(\sigma(s)_i) \qquad (9)$$

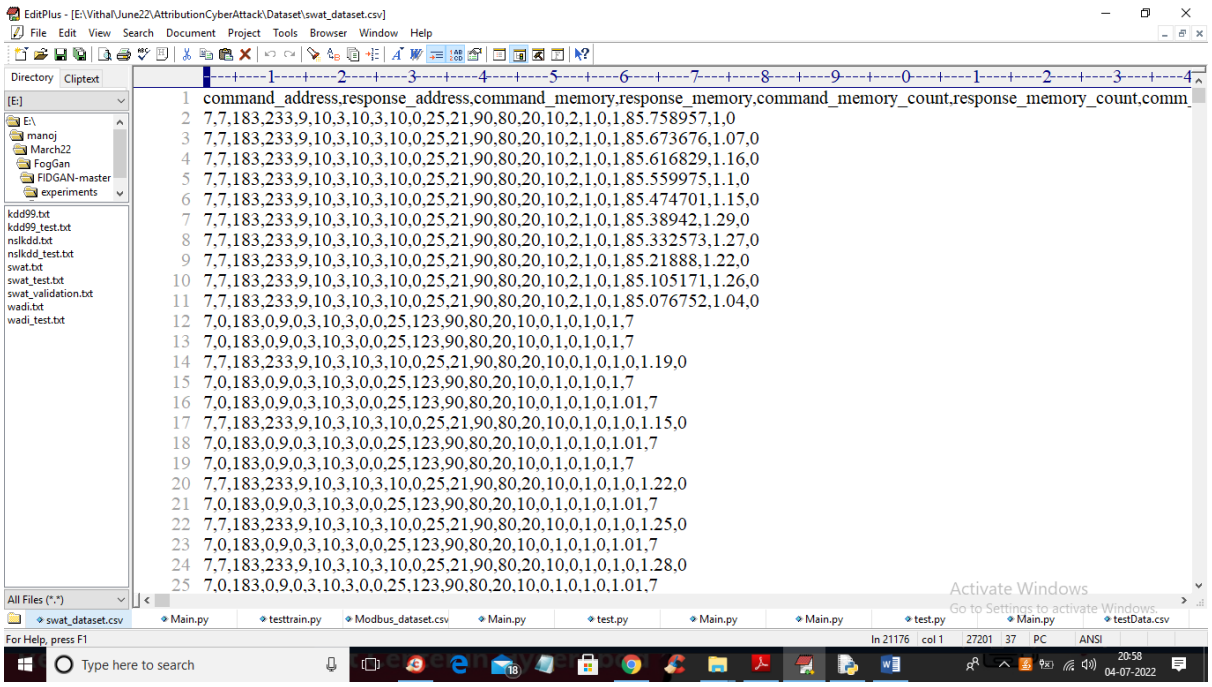
where log((s)i) is the softmax function and yi is the i-th class label. Changing assault patterns allows this technique to self-modify without pre-processing. This is the outcome of concurrently updating the ensemble model and all one-vs-all classifier weights using the gradient descent approach.

The assault attribution system benefits from adding additional attack attributes. The new dataset with the new attack attribute is attributed using the approved manner. Algorithm 2 illustrates the attack attribution component algorithm.

CHAPTER-4

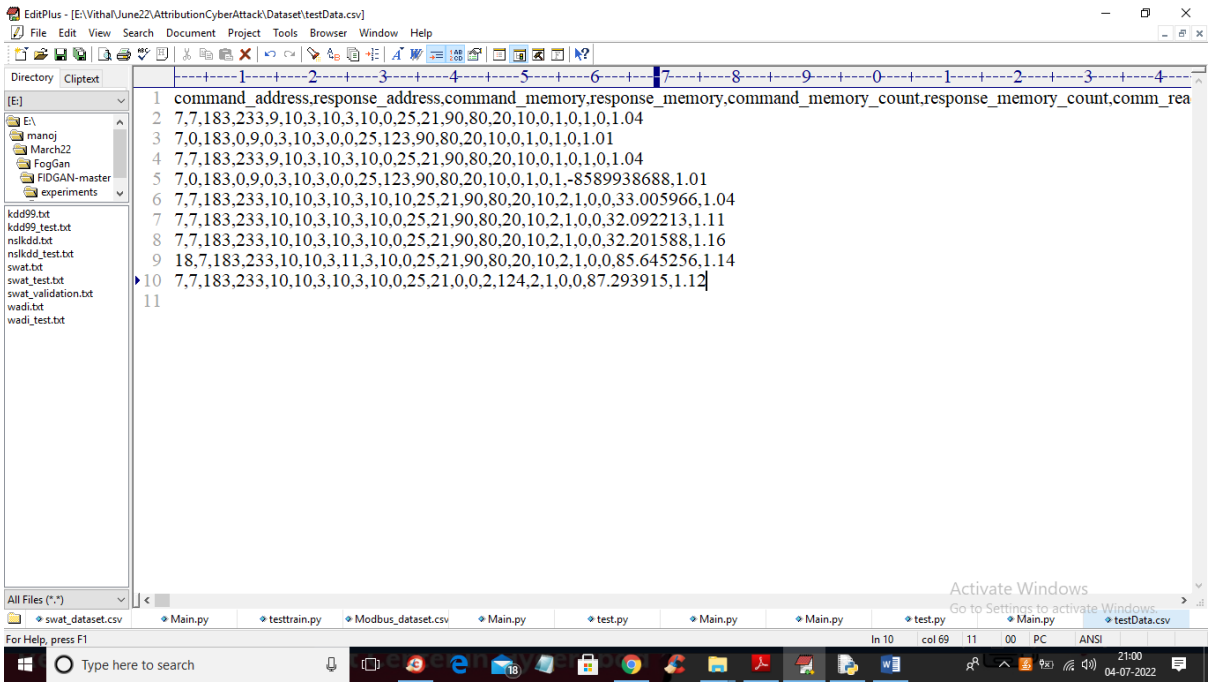
RESULT

SWAT (secure water treatment) was used to implement this project, This collection also includes IOT request and response signatures.Each dataset has a unique cyber-attack label: "Normal," "NMRI" ,"CMRI"," "MSCI," "MPCI," and "Malic" The attacks listed above were discovered in the dataset, which includes the labels listed above as integer values for their indexes. The NORMAL label index ranges from 0 to 8 class labels. Dataset screenshot below.



The first row of the dataset screen shown above lists the dataset column names. The next rows include the dataset values, and the final column lists the attack types, labelled 0 through 7. We will train the proposed Auto Encoder, decision tree, and DNN algorithms using the aforementioned dataset.

The test data on the screen below is NEW and just includes a signature; no class labels are present. The proposed technique will identify and attribute class labels.



The test data has a classless IOT request signature.

In order to carry out this project, we created the following modules.

1)In the previously mentioned test data, there is an IOT request signature lacking class identifiers. We'll read the dataset when it's uploaded, then use it to discover various assaults.

2) Using this module, we will set any missing values to zero, After MIN-MAX scaling normalised feature values, divide the dataset into train and test with 80% training and 20% testing.

3) Execute the AutoEncoder method: This module trains the deep learning methodology and extracts model features.

4) Run the Decision Tree with PCA: Before retraining the Decision Tree, the AutoEncoder-extracted features will be modified using PCA to reduce their size. Decision trees use dataset signatures to label records.

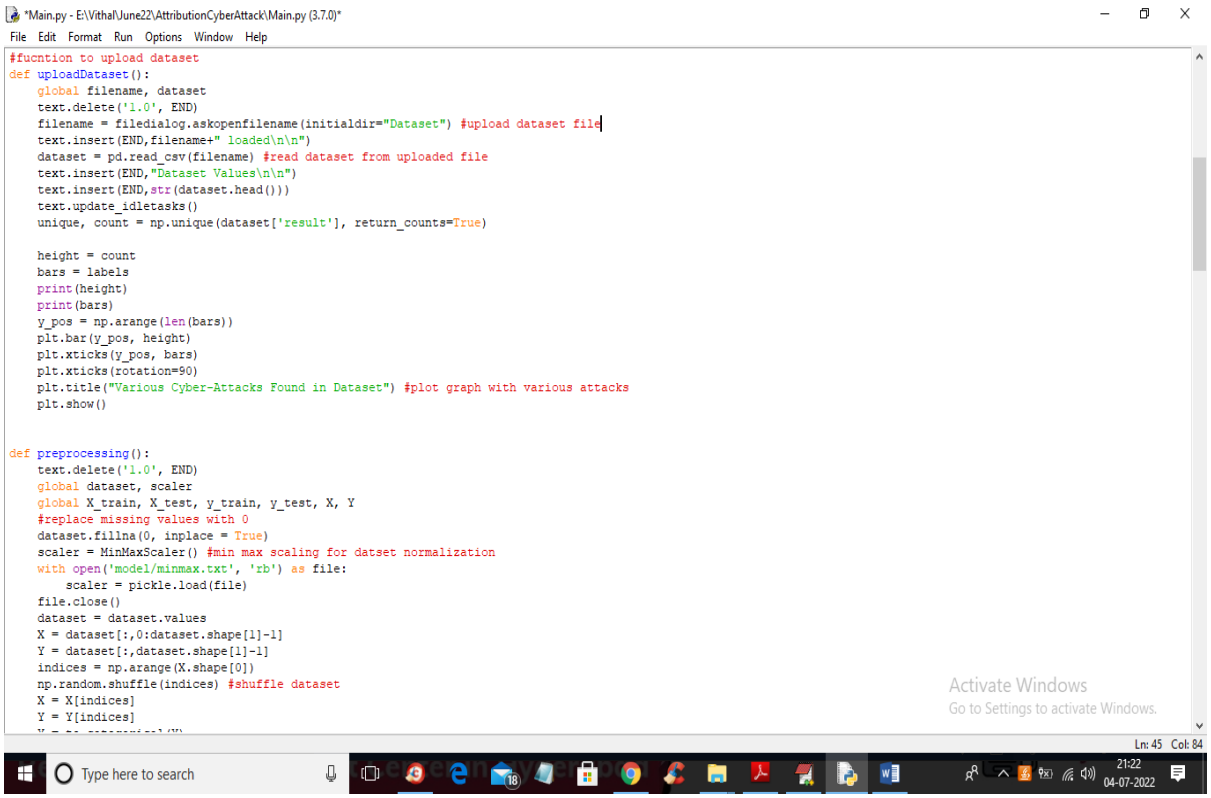
5) Execute the DNN technique: The projected decision tree label will be further trained to identify and attribute assaults using the DNN (deep neural network) technique.

6) Detection and Attribute Attack classification: By using this module, we will input TEST DATA that is unlabeled or unidentified, and DNN will then forecast the sort of attack.

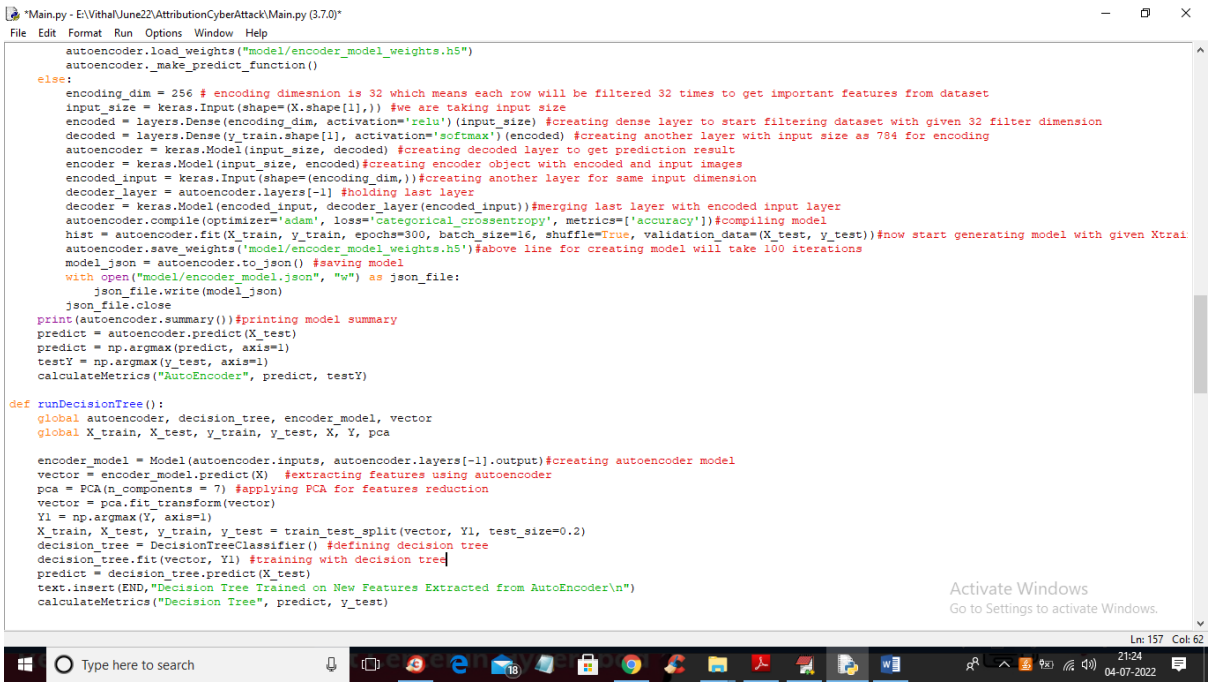
7) Comparison Graph: We will create a comparison graph between all methods using this module.

8) Comparison Table: This module will compare all algorithms by recall, accuracy, and FSCORE.

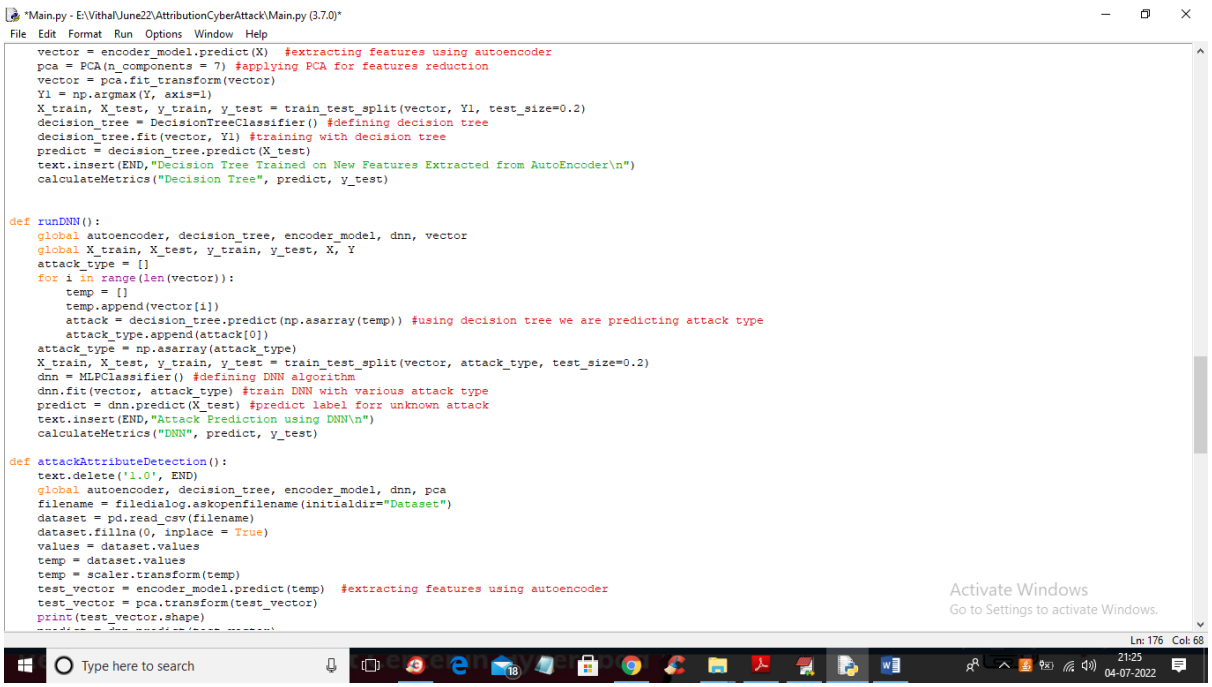
You may read the red-colored comments on the screen below to learn about how the algorithms are implemented.



Read the red-colored comments in the above image to learn about dataset loading and min-max normalisation.

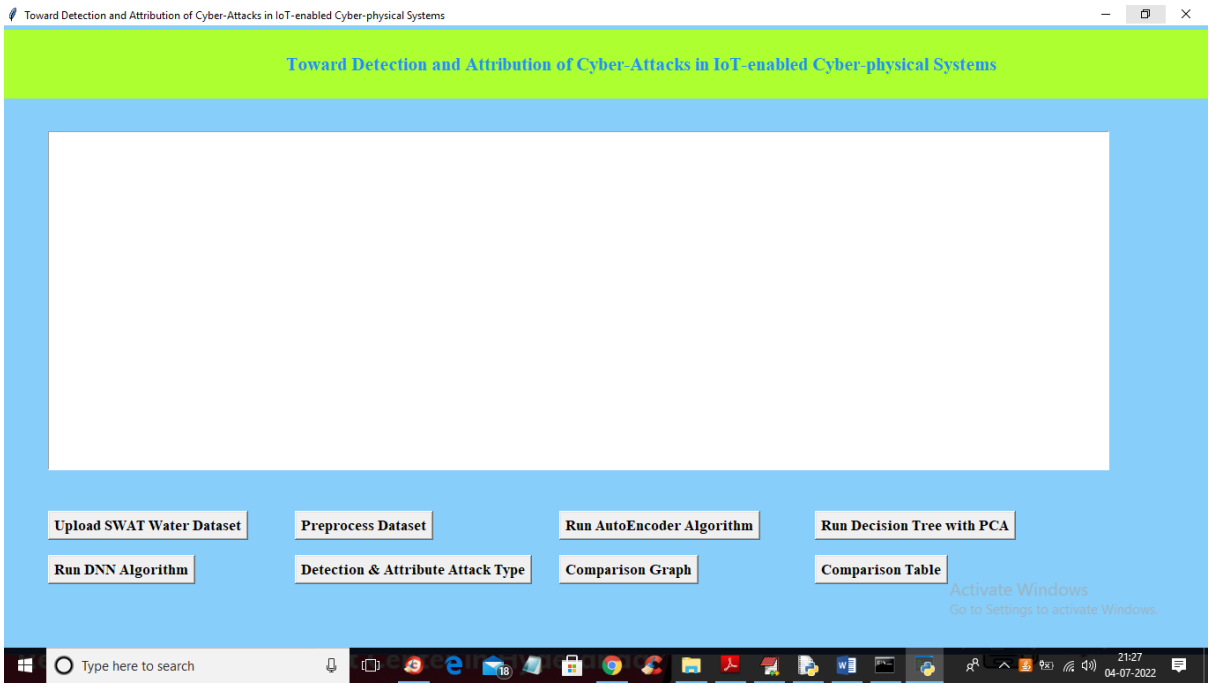


You can see that we used AutoEncoder, PCA, and a decision tree to train the dataset on the screen above. on the screen below, we used DNN methods to train the dataset using labels predicted by the decision tree.

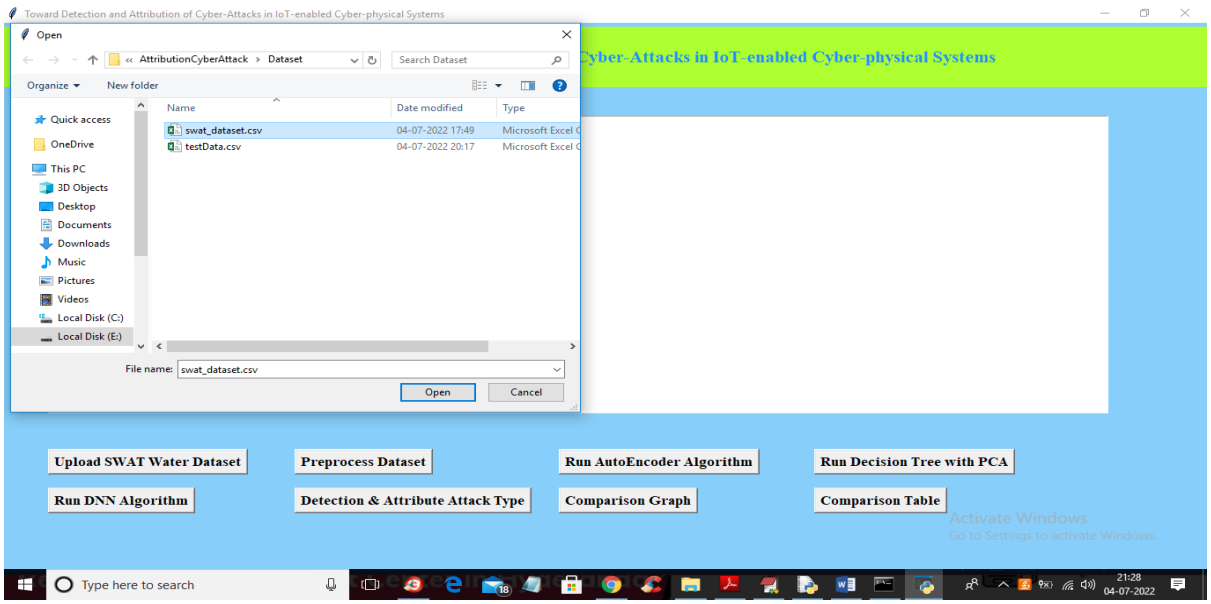


We are using DNN techniques to train a dataset in the screen above.

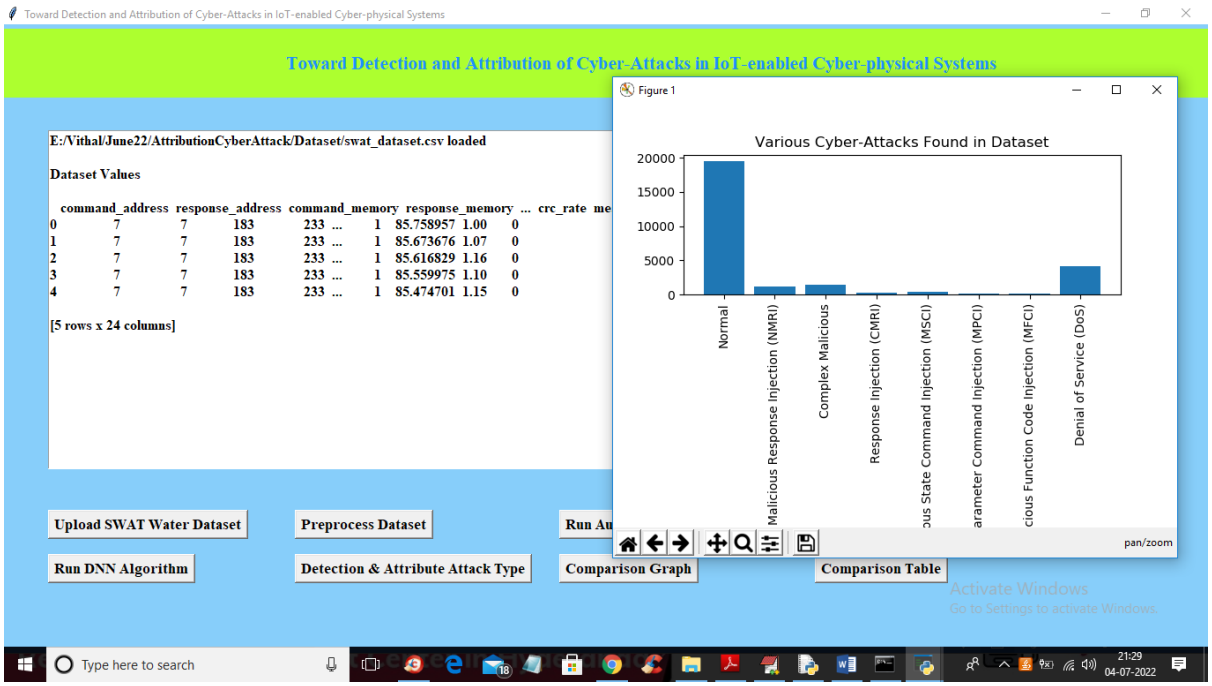
To reach the screen below, double-click the 'run.bat' file to launch the project.



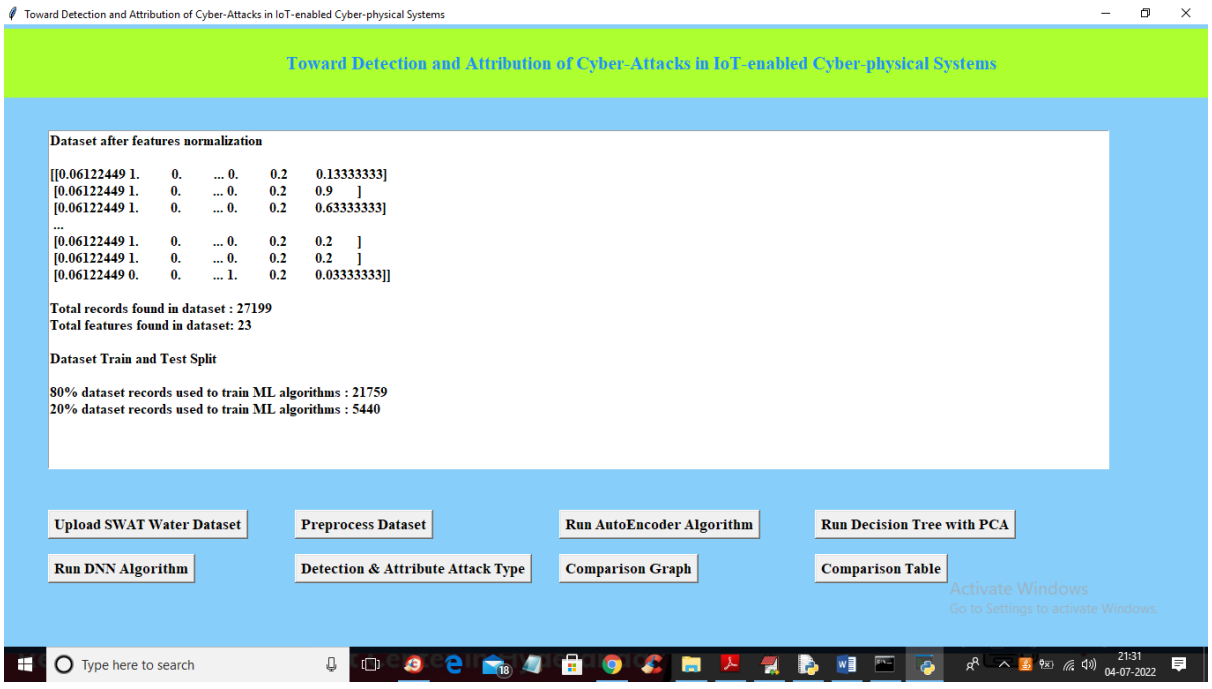
To upload the dataset to the programme, choose the "Upload SWAT Water Dataset" button on the previous page. The outcome is shown below.



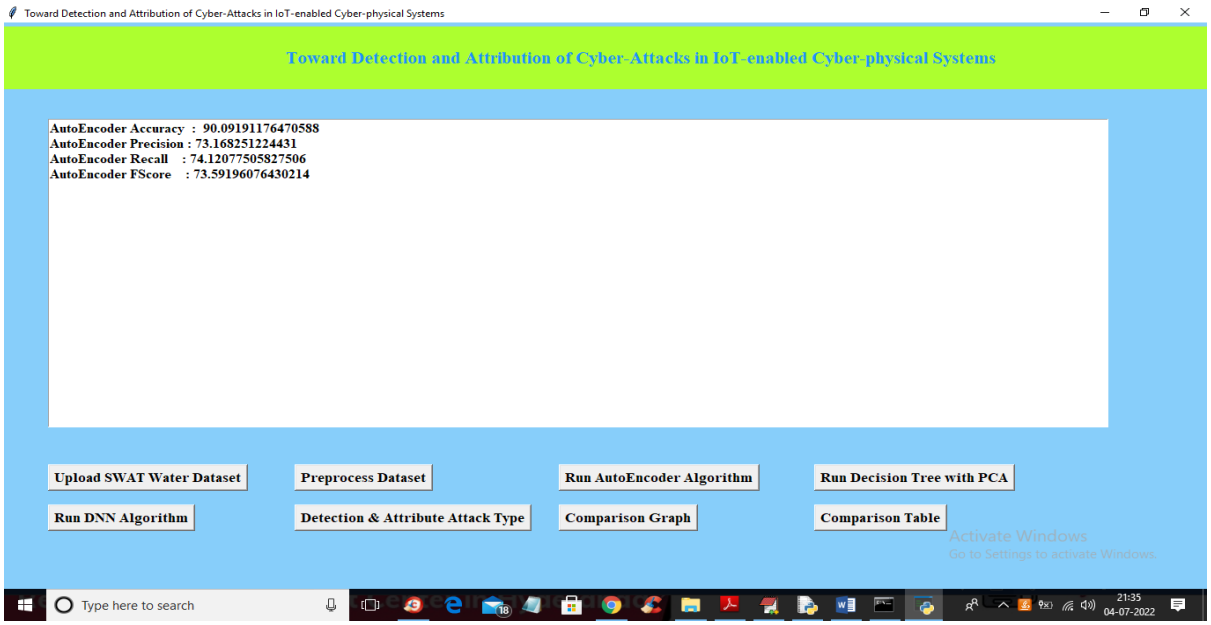
Choose the SWAT dataset file in the aforementioned page, upload it, and then click the "Open" button to load the dataset to produce the result shown below.



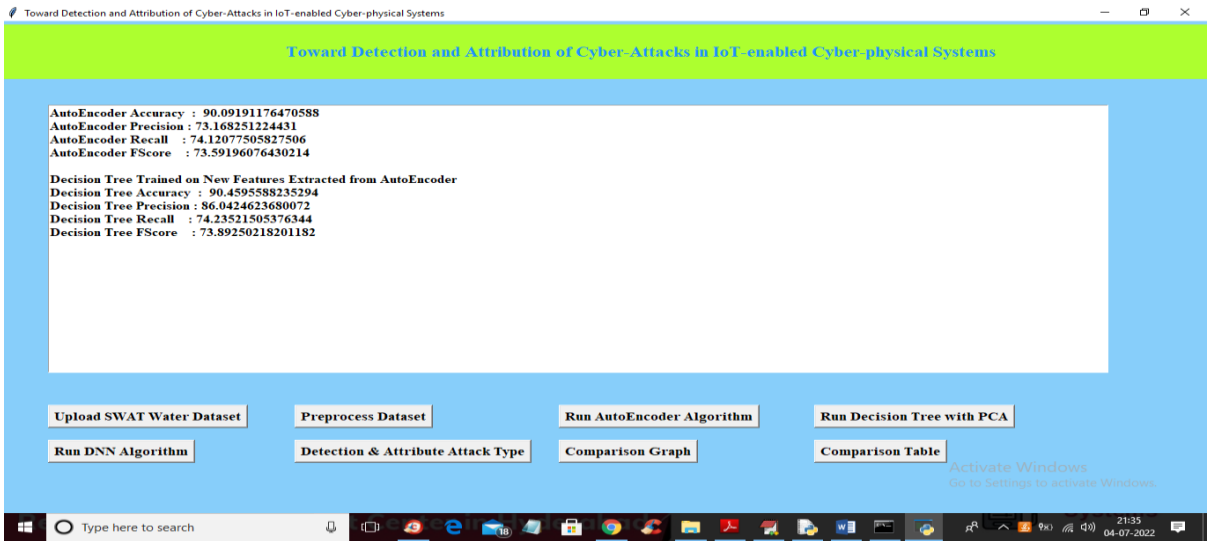
The dataset is loaded in the image above, and the graph's x-axis lists the attack names while The y-axis shows the number of assaults in the dataset. The dataset's 'NORMAL' class has many records, while other attacks have few, causing a data imbalance that the AutoEncoder, Decision Tree, and DNN can rectify. "Preprocess Dataset" removes missing data and normalises values using MIN-MAX.



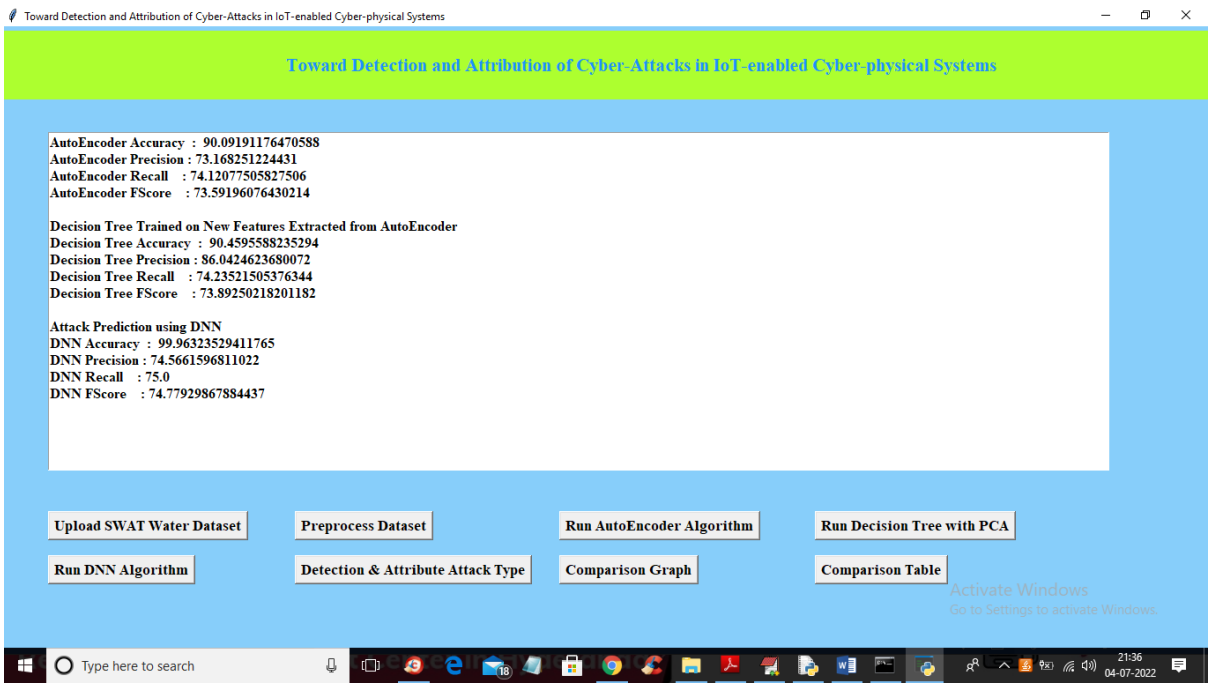
The data on the screen above has been normalised (normalisation is the process of transforming data between 0 and 1), After that, we can see the dataset's and train and test splits' total records. Now that the dataset is prepared, use the "Run AutoEncoder Algorithm" button to train it using the autoencoder and get the accuracy shown below.



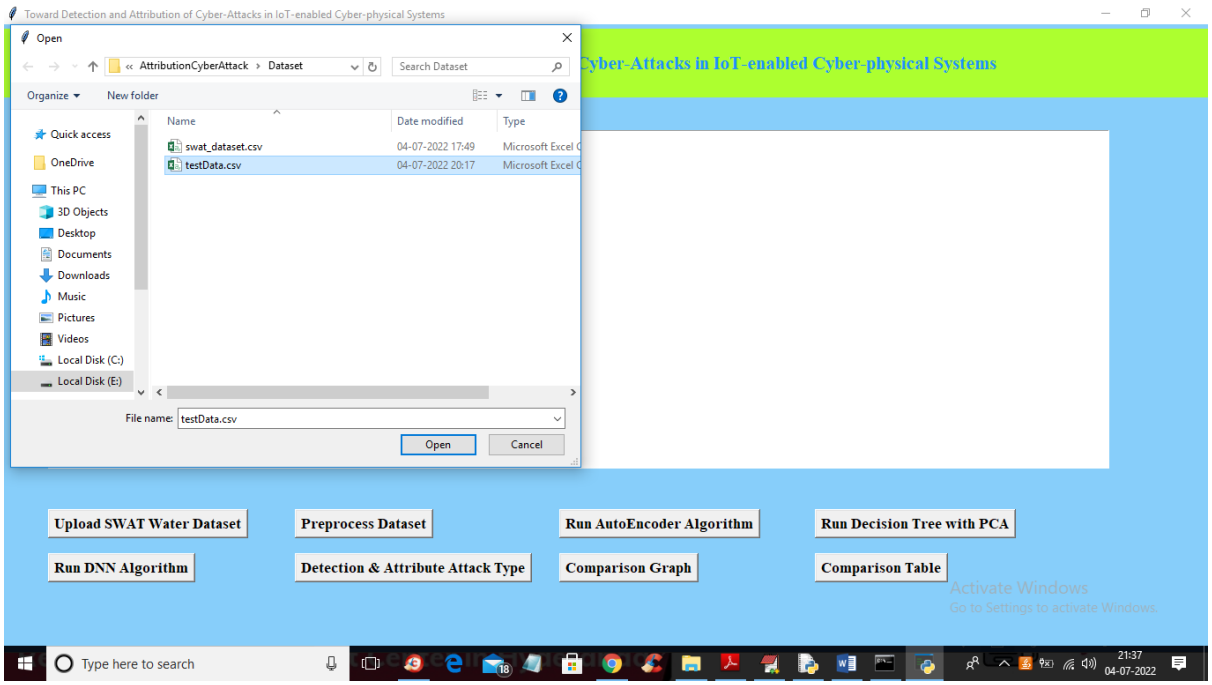
By using the Decision Tree with PCA technique, we were able to get 90% accuracy on the screen above using AutoEncoder, and this accuracy may be increased. Click the "Run Decision Tree with PCA" button to obtain the result shown below.



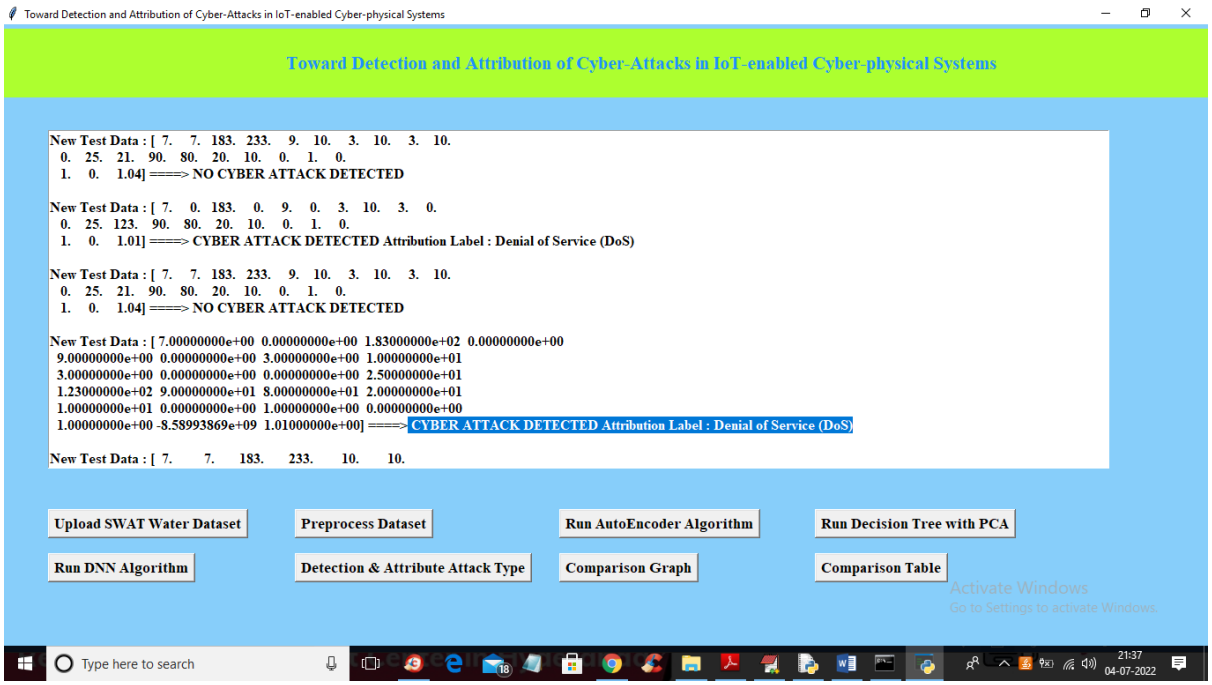
Clicking the "Run DNN Algorithm" button will increase accuracy even further, giving the result shown below. In the screen above, we can see that decision trees have improved accuracy and precision values.



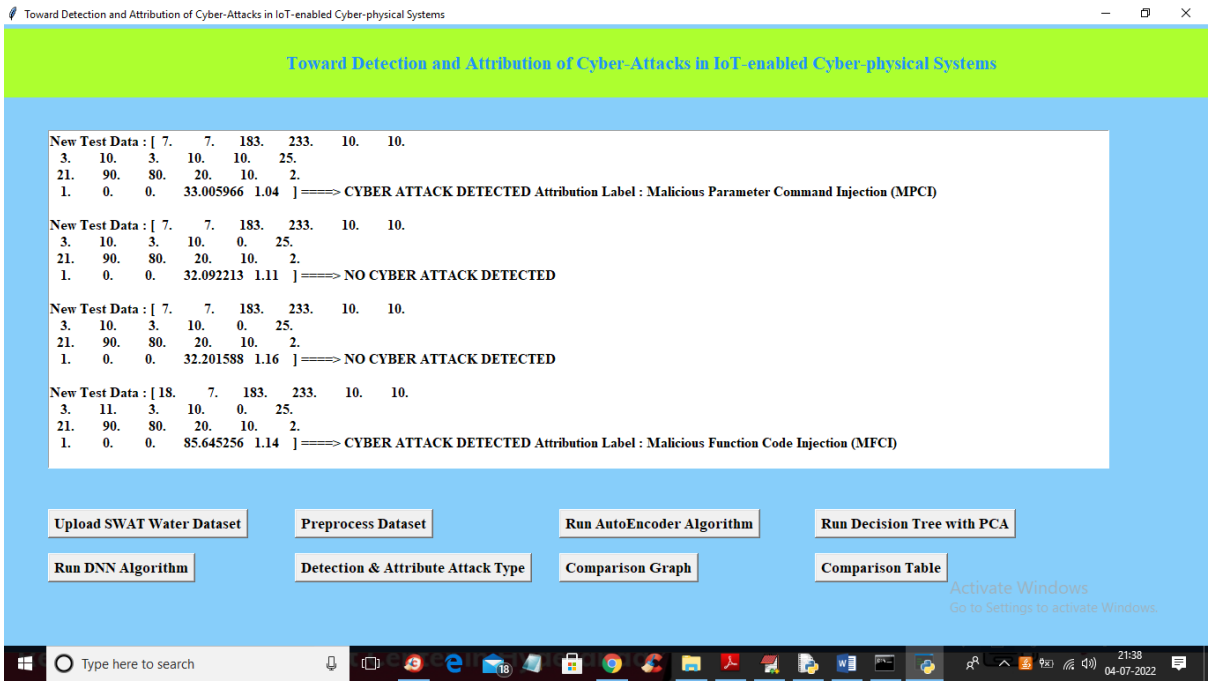
To submit test data and identify attacks, click the "Detection & Attribute Attack Type" buttoncharacteristics after using DNN to achieve 99% accuracy in the screen above.



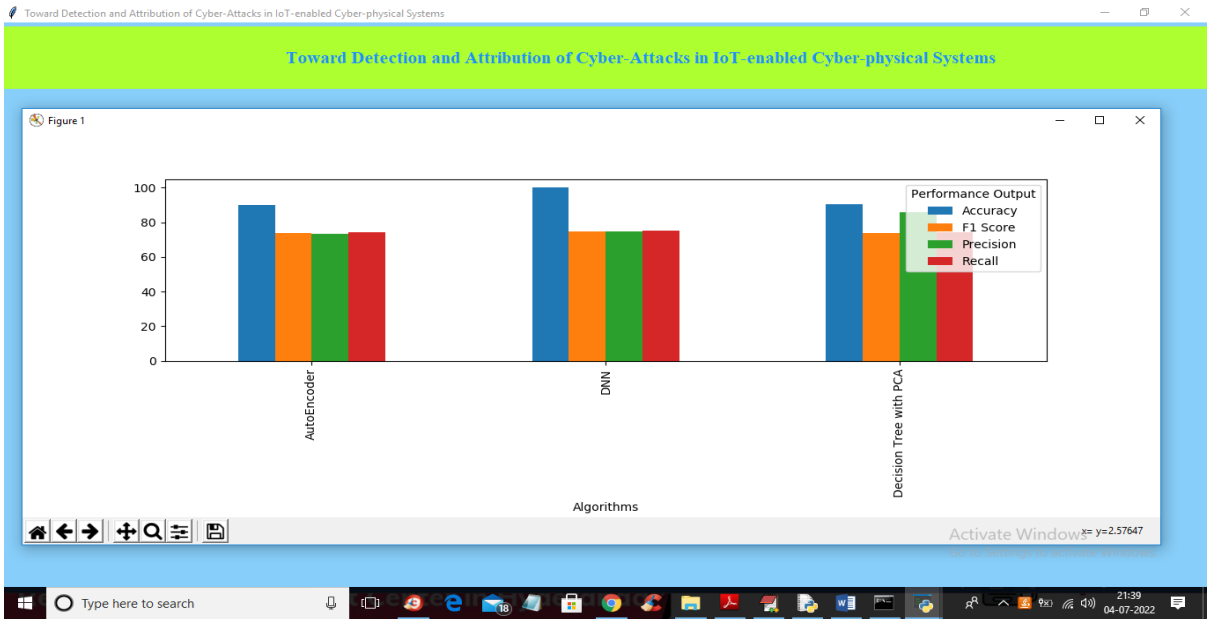
To acquire the results shown below, choose the "TEST DATA" file in the screen above, upload it, and then click the "Open" button.



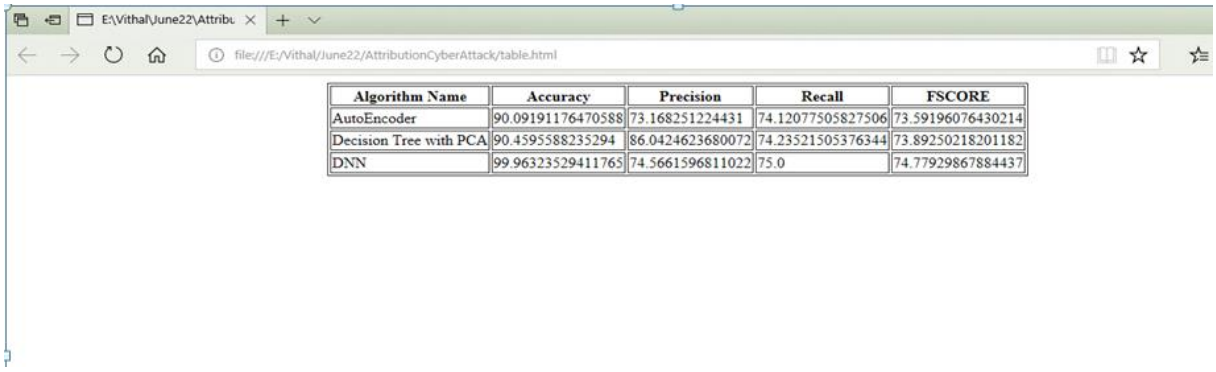
In the screen above, the numbers for the test data are shown in square brackets, and the attack type that was detected is shown after the arrow =To see all detection, scroll down above the text box.



The 'Comparison Graph' button will take you to the graph above after selecting one of the assaults from the 'Detected' section of the screen.



Close the above graph, click "Comparison Table," and you will see a comparison table of all algorithms. The x-axis depicts algorithms, while the y-axis shows precision, recall, accuracy, and FSCORE with coloured bars. All DNN algorithms worked.



Algorithm Name	Accuracy	Precision	Recall	FSCORE
AutoEncoder	90.09191176470588	73.168251224431	74.12077505827506	73.59196076430214
Decision Tree with PCA	90.4595588235294	86.0424623680072	74.23521505376344	73.89250218201182
DNN	99.96323529411765	74.5661596811022	75.0	74.77929867884437

The names of the algorithms and their metrics, such as accuracy and precision, are shown in the table above.

CHAPTER 5

CONCLUSION

For the purpose of detecting and attributing intrusions in imbalanced ICS data, this work developed a unique two-stage ensemble deep learning-based paradigm. After deep representation learning translates the data to a higher-dimensional space, DTs are used to identify attack samples in the attack detection stage. This phase has the ability to detect previously unknown attacks and is resistant to skewed data sets. In the step of attack attribution, each one-vs-all classifier was trained on a distinct attack attribute.As shown, the model develops a complicated DNN with a partly and completely linked component that can effectively recognise intrusions. Despite the complexity of the proposed framework's architecture, the training and testing phases' computational complexity—where n is the number of training samples—is O(n 4) and O(n 2), respectively, comparable to earlier DNN-based methodologies.Furthermore, the suggested system outperforms earlier research in terms of recall and f-measure for timely detection and attribution of data. The construction of a cyber-threat hunting component is a future development that will help with the discovery of anomalies that are hidden from the detection component, for example by creating a typical profile across the whole system and the assets.

5.1 REFERENCES

[1] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, and J. Coble, "Multilayer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System, and Process Data," IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp. 4362–4369, 2019.

[2] R. Ma, P. Cheng, Z. Zhang, W. Liu, Q. Wang, and Q. Wei, "Stealthy Attack Against Redundant Controller Architecture of Industrial CyberPhysical System," IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9783–9793, 2019.

[3] E. Nakashima, "Foreign hackers targeted U.S. water plant in apparent malicious cyber attack, expert says." [Online]. Available: https://www.washingtonpost.com/blogs/checkpointwashington/post/foreign-hackers-broke-into-illinois-water-plant-controlsystem-industry-expert-says/2011/11/18/gIQAgmTZYN blog.html

[4] G. Falco, C. Caldera, and H. Shrobe, "IIoT Cybersecurity Risk Modeling for SCADA Systems," IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4486–4495, 2018.

[5] J. Yang, C. Zhou, S. Yang, H. Xu, and B. Hu, "Anomaly Detection Based on Zone Partition for Security Protection of Industrial Cyber-Physical Systems," IEEE Transactions on Industrial Electronics, vol. 65, no. 5, pp. 4257–4267, 2018.

[6] S. Ponomarev and T. Atkison, "Industrial control system network intrusion detection by telemetry analysis," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 252–260, 2016.

[7] J. F. Clemente, "No cyber security for critical energy infrastructure," Ph.D. dissertation, Naval Postgraduate School, 2018.

[8] C. Bellinger, S. Sharma, and N. Japkowicz, "One-class versus binary classification: Which and when?" in 2012 11th International Conference on Machine Learning and Applications, vol. 2, 2012, pp. 102–106.

[9] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[11] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6822–6834, 2019.

[12] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain, and A. Nawaz, "HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," IEEE Access, vol. 7, pp. 89 507–89 521, 2019.

[13] T. K. Das, S. Adepu, and J. Zhou, "Anomaly detection in industrial control systems using logical analysis of data," Computers & Security, vol. 96, p. 101935, 2020.

[14] J. J. Q. Yu, Y. Hou, and V. O. K. Li, "Online False Data Injection Attack Detection With Wavelet Transform and Deep Neural Networks," IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3271–3280, 2018.

[15] M. M. N. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, and M. Gidlund, "A machine-learning-based technique for false data injection attacks detection in industrial iot," IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8462–8471, 2020.

[16] W. Yan, L. K. Mestha, and M. Abbaszadeh, "Attack detection for securing cyber physical systems," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8471–8481, 2019.

[17] A. Cook, A. Nicholson, H. Janicke, L. Maglaras, and R. Smith, "Attribution of Cyber Attacks on Industrial Control Systems," EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, vol. 3, no. 7, p. 151158, 2016.

[18] L. Maglaras, M. Ferrag, A. Derhab, M. Mukherjee, H. Janicke, and S. Rallis, "Threats, Countermeasures and Attribution of Cyber Attacks on Critical Infrastructures," ICST Transactions on Security and Safety, vol. 5, no. 16, p. 155856, 2018.

[19] M. Alaeiyan, A. Dehghantanha, T. Dargahi, M. Conti, and S. Parsa, "A Multilabel Fuzzy Relevance Clustering System for Malware Attack Attribution in the Edge Layer of Cyber-Physical Networks," ACM Transactions on Cyber-Physical Systems, vol. 4, no. 3, pp. 1–22, 2020.

[20] U. Noor, Z. Anwar, T. Amjad, and K.-K. R. Choo, "A machine learning-based FinTech cyber threat attribution framework using highlevel indicators of compromise," Future Generation Computer Systems, vol. 96, pp. 227–242, 2019.

[21] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," Chemometrics and Intelligent Laboratory Systems, vol. 2, no. 1, pp. 37 – 52, 1987, proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[22] A. N. Jahromi, J. Sakhnini, H. Karimpour, and A. Dehghantanha, "A deep unsupervised representation learning approach for effective cyber-physical attack detection and identification on highly imbalanced data," in Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering, ser. CASCON '19. USA: IBM Corp., 2019, p. 14–23.

[23] T. Morris, Z. Thornton, and I. Tunipseed, "Industrial control system simulation and data logging for intrusion detection system research," in 7th Annual Southeastern Cyber Security Summit, 2015.

[24] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in Critical Information Infrastructures Security, G. Havarneanu, R. Setola, H. Nassopoulos, and S. Wolthusen, Eds. Cham: Springer International Publishing, 2017, pp. 88–99.

[25] S. N. Shirazi, A. Gouglidis, K. N. Syeda, S. Simpson, A. Mauthe, I. M. Stephanakis, and D. Hutchison, "Evaluation of anomaly detection techniques for scada communication resilience," in 2016 Resilience Week (RWS), 2016, pp. 140–145.

[26] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly detection for a water treatment system using unsupervised machine learning," IEEE International Conference on Data Mining Workshops, ICDMW, vol. 2017-November, pp. 1058–1065, 2017.

[27] M. Kravchik and A. Shabtai, "Detecting cyber attacks in industrial control systems using convolutional neural networks," Proceedings of the ACM Conference on Computer and Communications Security, no. 1, pp. 72–83, 2018.

[28] S. D. Anton, A. Hafner, S. Sinha, and H. Schotten, "Anomaly-based intrusion detection in industrial aata with SVM and random forests," in the 27th International Conference on Software, Telecommunicationsand Computer Networks (SoftCOM). IEEE, 2019.

[29] M. Kravchik and A. Shabtai, "Efficient cyber attack detection in industrial control systems using lightweight neural networks and pca," IEEE transactions on dependable and secure computing, pp. 1–1, 2021.

[30] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S. K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11730 LNCS, pp. 703–716, 2019.

[31] Q. Lin, S. Verwer, S. Adepu, and A. Mathur, "TABOR: A graphical model-based approach for anomaly detection in industrial control systems," ASIACCS 2018 - Proceedings of the 2018 ACM Asia Conference on Computer and Communications Security, pp. 525–536, 2018.

[32] C. Feng, T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and lstm networks," in 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2017, pp. 261–272.

[33] M. Macas and W. Chunming, "Enhanced cyber-physical security through deep learning techniques," CEUR Workshop Proceedings, vol. 2457, no. 38, 2019.

[34] C.-t. Chu, S. Kim, Y.-a. Lin, Y. Yu, G. Bradski, K. Olukotun, and A. Ng, "Map-reduce for machine learning on multicore," in Advances in Neural Information Processing Systems, B. Scholkopf, J. Platt, and T. Hoffman, ¨ Eds., vol. 19. MIT Press, 2007, pp. 281–288.

[35] J. Su and H. Zhang, "A fast decision tree learning algorithm," in Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1, ser. AAAI'06. AAAI Press, 2006, p. 500–505.

APPENDIX

Python

* One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

*It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms. Python is a big language with a free, binary-distributed interpreter standard library.

* It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming.

* It supports several different programming paradigms in addition to object-oriented programming, including practical and procedural programming. Python is mighty while maintaining a relatively straightforward syntax. Classes, highly dynamic data types, modules, and exceptions are covered. Python can also be utilised by programmes that require programmable interfaces as an external language.

Python Features:

1) Easy: Because Python is a more accessible and straightforward language, Python programming is easier to learn.

2) Interpreted language: Python is an interpreted language, therefore it can be used to examine the code line by line and provide results.

3) Open Source: Python is a free online programming language since it is open-source.

4) Portable: Python is portable because the same code may be used on several computer standard **libraries:** Python offers a sizable library that we may utilize to create applications quickly.

6) GUI: It stands for GUI (Graphical User Interface)

7) Dynamical typed: Python is a dynamically typed language, therefore the type of the value will be determined at runtime.

Python GUI (Tkinter)

* Python provides a wide range of options for GUI development (Graphical User Interfaces).

* Tkinter, the most widely used GUI technique, is used for all of them.

* The Tk GUI toolkit offered by Python is used with the conventional Python interface.

* Tkinter is the easiest and quickest way to write Python GUI programs.

* Using Tkinter, creating a GUI is simple.

* A part of Python's built-in library is Tkinter. The GUI programs were created.

* Python and Tkinter together give a straightforward and quick way. The Tk GUI toolkit's object-oriented user interface is called Tkinter.

* Making a GUI application is easy using Tkinter. Following are the steps:

1) Install the Tkinter module in place.

2) The GUI applicatioMakeske the primary window

3) Include one or more of the widgets mentioned above in the GUI application.

4) Set up the main event loop such that it reacts to each user-initiated event.

*Although Tkinter is the only GUI framework included in the Python standard library, Python includes a GUI framework. The default library for Python is called Tkinter. Tk is a scripting

language often used in designing, testing, and developing GUIs. Tk is a free, open-source widget toolkit that may be used to build GUI applications in a wide range of computer languages.

Machine Learning

*Artificial intelligence (AI), which includes machine learning, enables computer systems to learn without being explicitly programmed. It has to do with statistics and applied mathematics. Mike Robert's definition of machine learning. As a computer gathers and learns from the data it provides, it may operate more correctly via machine learning.

*For large classes of machine learning, many algorithms are used. We must provide algorithms with more precise data for them to complete certain jobs. In some circumstances, a computer will utilize data to gather information, check its output against the desired outcome, and make necessary corrections.

*For instance, when someone texts on a phone, the phone learns about spelling errors and either autocorrects the offending word or suggests a replacement. For many top organizations, machine learning is a critical component of the creation of new products.

*ML is an important factor in the operations of many companies, like Facebook and Google. Data science uses machine learning in many different ways. Data scientists rely on ML approaches to carry out their modeling. Regression and classification are of utmost relevance in data science; hence, the main tool utilized in ML is to accomplish such objectives.

* ML applies applicable to practically all phases of data science and is most often associated with the data modeling phase. Python has been the primary computer language used for data processing. Several Python packages are used in ML settings. The three sections of Python are huge data, optimizing your code, and data files in memory.

1.6 Types of Machine Learning

There are three fundamental forms of machine learning: -supervising, semi-supervised, and machine learning

a) Supervised Machine Learning

* That method looks for patterns in the labeled data set to obtain results. Data labeling in supervised learning requires human intervention. To train the algorithm with labeled inputs and the intended output, supervised ML requires human participation. ML under supervision is good for a task like;

I. Classify the data using a binary system into two groups.

II. Multi-classification: The division of data into more than two categories,

III. Modeling imaging continuous value using regression.

IV. Assembling: Compiling the estimates from many ML models to provide a precise estimate.

b) Unsupervised Machine Learning

*This method searches for patterns in the data collection without relying on labeled data or human interaction. Data labeling is not necessary for this strategy. ML Unsupervised is effective for tasks like;

I. Dimensionality reduction: Reduce the number of variables in the data collection.

II. Clustering: Grouping the dataset based on similarities.

III. Association mining identifies the item or group of items that commonly appear together in data.

IV. Data point identification for anomaly detection in the data set

c) Semi-supervised Learning:

*For this method, you require labeled data. As a consequence, human interaction is also necessary, but the process still moves forward. In this kind of learning, the algorithm is given a tiny quantity of labeled data by data scientists, and as a result, the algorithm gains knowledge about the data set's dimension, which it may then apply to mother del, unlabeled data.

* There are several contexts in which semi-supervised machine learning (ML) may be used.

I. Machine translation: Language conversion using a learning system.

II. Data labeling: An algorithm trained on modest amounts of data will automatically apply data labels to enormous collections.

1.7 Uses of Machine Learning

*Machine learning is used in many areas nowadays. The most well-known example is the machine learning recommendation engine that drives a book's news feed. This engine makes an effort to reinforce established patterns in a user's online activity inside a certain Facebook group.

*The news is appropriately adjusted if a user alters the design and doesn't read anything from that particular group the following week. Applications of machine learning (ML) include business intelligence, human resource information systems, autonomous vehicles, and virtual assistants.

Advantages:

• ML helps enterprises in comprehending their clients. ML assists in improving goods in response to client demand by gathering the necessary user data and associating it with shifting behavior. Some companies' business models are heavily reliant on machine learning, such as Uber, which uses an algorithm to connect drivers and customers. To surface the advertising in searches, Google employs ML.

Disadvantage:

• ML might be expensive. High wages for machine learning are a result of data emotions command on the project. These initiatives also often demand expensive software infrastructure.

• In addition to that, when an algorithm is trained on a data set, ML bias might develop. That has flaws in it that might provide erroneous results.

Steps to choosing the suitable ML model

The issue is solved by selecting the best ML model, which might take some time. The steps are as follows:

1) For the difficulty with the pure date alignment, the input should be thought about.

2) Gather, label, and prepare the data as appropriate.

3) To put the right algorithms to use and test them to determine how well they perform.

Libraries Used

Pandas:

* Pandas is a Python computer language library for data analysis and manipulation. It offers a specific operation and data format for handling time series and numerical tables. It differs significantly from the release3-clause of the BSD license. It is a well-liked open-source of opinion that is utilized in machine learning and data analysis.

NumPy:

* The NumPy Python library for multi-dimensional, big-scale matrices adds a huge number of high-level mathematical functions. It is possible to modify NumPy by utilizing a Python library. Along with line, algebra, and the Fourier transform operations, it also contains several matrices-related functions.

Matplotlib:

* It is a multi-platform, array-based data visualization framework built to interact with the whole SciPy stack. MATLAB is proposed as an open-source alternative. Matplotlib is a Python extension and a cross-platform toolkit for graphical plotting and visualization.

Scikit-learn:

* The most stable and practical machine learning library for Python is scikit-learn. Regression, dimensionality reduction, classification, and clustering are just a few of the helpful tools it provides through the Python interface for statistical modeling and machine learning. It is an essential part of the Python machine learning toolbox used by JP Morgan. It is frequently used in various machine learning applications, including classification and predictive analysis.

Keras:

* Google's Keras is a cutting-edge deep learning API for creating neural networks. It is created in Python and is designed to simplify the development of neural networks. Additionally, it enables the use of various neural networks for computation. Deep learning models are developed and tested using the free and open-source Python software known as Keras.

h5py:

* The h5py Python module offers an interface for the binary HDF5 data format. Thanks to p5py, the top can quickly halt the vast amount of numerical data and alter it using the NumPy library. It employs common syntax for Python, NumPy, and dictionary arrays.