ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

IDENTIFYING AND PROTECTING CYBER-PHYSICAL SYSTEMS' INFLUENTIAL DEVICES FOR SUSTAINABLE CYBER SECURITY

Mr G Mukesh
Assistant Professor, Department of Computer Science and Engineering (Cyber Security), Sphoorthy Engineering College, Nadergul,501510
g.mukesh@sphoorthyengg.ac.in

D. Veneela Reddy,
Computer Science and Engineering (Cyber Security), Sphoorthy Engineering College, Nadergul, Hyderabad - 501510, Telangana, India,
daidaveneelareddy@gmail.com

P. Likhil Reddy,
Computer Science and Engineering (Cyber Security), Sphoorthy Engineering College, Nadergul, Hyderabad - 501510, Telangana, India,
reddylikhil03@gmail.com

M. Akash,
Computer Science and Engineering (Cyber Security), Sphoorthy Engineering College, Nadergul, Hyderabad - 501510, Telangana, India,
Bittuakash906@gmail.com

P. Pavan,
Computer Science and Engineering (Cyber Security), Sphoorthy Engineering College, Nadergul, Hyderabad - 501510, Telangana, India,
Pogakupavan16@gmail.com

Abstract:

Cyber-Physical Systems (CPS) are characterized by a wide range of complex multi-tasking components with close interaction that leads to integrating cyber sections into the physical world. Considering the significant growth of cyber-physical systems and due to the widespread use of smart features and communication tools, new challenges have emerged. In this regard, a new generation of CPSs such as the smart grid are facing different vulnerabilities and many threats and attacks. Therefore, the most important challenges for these systems are security and privacy. Anomaly detection is an important data analysis task as one of the approaches for CPSs security. As different anomaly detection methods are presented, it is difficult to compare the advantages and disadvantages of these techniques. Finally, in this chapter Machine Learning (ML) methods for detection of anomalies are presented through a case study which demonstrates the effectiveness of machine learning techniques at classifying False Data Injection (FDI) attacks.

Keywords: Cyber-Physical Systems, Cybersecurity, Anomaly Detection, Machine Learning, False Data Injection, Random Forest, Decision Tree, XGBoost.

1. INTRODUCTION

1.1 Motivation

The amount of application service that is streamed to their users has increased explosively. This type of service requires minimal installations and computing power on the user terminal because the applications are operating at the service carrier's cloud servers instead of the local terminal; all the inputs and outputs are streamed to the users via the internet. Seeing the obvious advantage of providing high-end service to customers, who are not able to access high-end devices, many corporations have started to develop their streaming services. For instance, entertaining service such as Google Stadia makes high-end gaming, which is typically hardware demanding, now possible on any portable devices with good internet connectivity. The game is processed and rendered at Google's cloud server with user's inputs in real-time, then the video is streamed back to the user's terminal via the internet. However, the extensive data exchange at the network between the cloud servers and local user terminals also expand the attack surface for intrusions. Malicious hackers may deploy various types of attacks, such as Distributed Denial-of-Service (DDoS), Port Scan and Infiltration attack to hijack valuable data or make servers unavailable to users. To stop these cyberattacks from happening, the development of a reliable and effective Intrusion Detection System (IDS) for cybersecurity.

The motivation for identifying and protecting cyber-physical systems influential devices for a sustainable cybersecurity project based on ML algorithms like Random Forest, Decision Tree, and XGBoost lies in addressing the growing cybersecurity challenges posed by complex, interconnected systems, and leveraging advanced analytical techniques to enhance resilience, adaptability, and compliance with regulatory mandates.

1.2 Objectives

The main objectives of our project are:

- To detect CPS effectively.
- To implement the machine learning using Random Forest, Decision and Xgboost.
- To enhance the overall performance analysis.

1.3 Problem Statement

A more common approach for detecting main problem in detecting slow DDoS attacks is the inability to prevent them, since the determination process is based on the study of existing traffic without the possibility of predicting it depending on users' activity.

1.4 Scope of the Project

DoS or Denial-of-Service attack is an attack targeting the availability of web applications. Unlike other kinds of attacks, the primary goal of a DoS attack is not to steal information but to slow or take down a web site. Prevention can be used to perform Distributed Denial-of-Service (DDoS) attacks, steal data, send spam, and allow the attacker to access the device and its connection. A Denial-of-Service (DoS) attack is an attack meant to shut down a Deep or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash.

2. LITERATURE SURVEY

[1] Title: Cloud security architecture based on user authentication and symmetric key cryptographic techniques, 2020

- **Author:** Abdul Raoof
- **Technologies and Algorithm Used:** The study is implemented on the Structure for cloud security with efficient security in communication system and AES based file encryption system. This security architecture can be easily applied on PaaS, IaaS and SaaS and one time password provides extra security in the authenticating users.
- **Advantages:** Performance time and accuracy.
- **Disadvantages:** Training model prediction on Time is High. It is based on Low Accuracy.

[2] Title: Analysis and Countermeasures for Security and Privacy Issues in Cloud Computing, 2019

- **Author:** Q. P. Rana, Nitin Pandey
- **Technologies and Algorithm Used:** The cloud computing environment is adopted by a large number of organizations so the rapid transition toward the clouds has fuelled concerns about security perspective. There are numbers of risks and challenges that have emerged due to use of cloud computing. The aim of this paper is to identify security issues in cloud computing which will be helpful to both cloud service providers and users to resolve those issues. As a result, this paper will access cloud security by recognizing security requirements and attempt to present the feasible solution that can reduce these potential threats.
- **Advantages:** More effective and efficient.
- **Disadvantages:** Not give accurate prediction result.

[3] Title: Detecting Distributed Denial of Service Attacks Using Data Mining Techniques, 2018

- **Author:** Linga
- **Technologies and Algorithm Used:** In this study, we DDoS (Distributed Denial of Service) attack has affected many IoT networks in recent past that has resulted in huge losses. We have proposed deep learning models and evaluated those using latest CICIDS2017 datasets for DDoS attack detection which has provided highest accuracy as 97.16% also proposed models are compared with machine learning algorithms.
- **Advantages:** The proposed solution can successfully detect network intrusions and DDOS communication with high precision. More Reliable.
- **Disadvantages:** It is less in efficiency and not give perfect result. This finding is disadvantageous to the organization experiencing such attack. The difficulty in identifying all articles that are related to this study.

3. EXISTING SYSTEM

In the current system, DDoS attacks are first detected, and then specific characteristics are forwarded to classifiers such as support vector machine, decision tree, naïve Bayes, and multilayer perceptron to ascertain the type of attack. The experimental study utilizes publicly accessible datasets like KDD Cup 99. The simulation results indicate that GOIDS combined with a decision tree demonstrates superior detection capabilities and accuracy, while maintaining a minimal false-positive rate. For example, employing denoising techniques as feature extractors could potentially enhance performance, particularly in environments with significant noise levels.

3.1 Disadvantages

- Doesn't Efficient for handling large volume of data.
- Theoretical Limits.
- Incorrect Classification Results.
- Less Prediction Accuracy.

4. PROPOSED SYSTEM

The proposed Intrusion Detection System (IDS) model detects network intrusions by categorizing network packet traffic as either benign or malicious. DDOS attacks from the KDD Cup 99 dataset have been utilized for training and validation purposes. The Random Forest, Decision Tree, and Xgboost models are employed for classification. The testing dataset is used to classify attacks or normal behavior in the anomaly detection model. This approach is more effective for performance analysis.

4.1 Advantages

- High performance.
- Provide accurate prediction results.
- It avoid sparsity problems.
- Reduces the information Loss and the bias of the inference due to the multiple estimates.

5. SYSTEM REQUIREMENTS

5.1 Software Requirements

- **O/S:** Windows 7
- **Language:** Python
- **Front End:** Anaconda Navigator – Spyder Notebook

5.2 Hardware Requirements

- **System:** Pentium IV 2.4 GHz
- **Hard Disk:** 200 GB
- **Mouse:** Logitech
- **Keyboard:** 110 keys enhanced
- **Ram:** 4GB

5.3 Software Description: Python

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

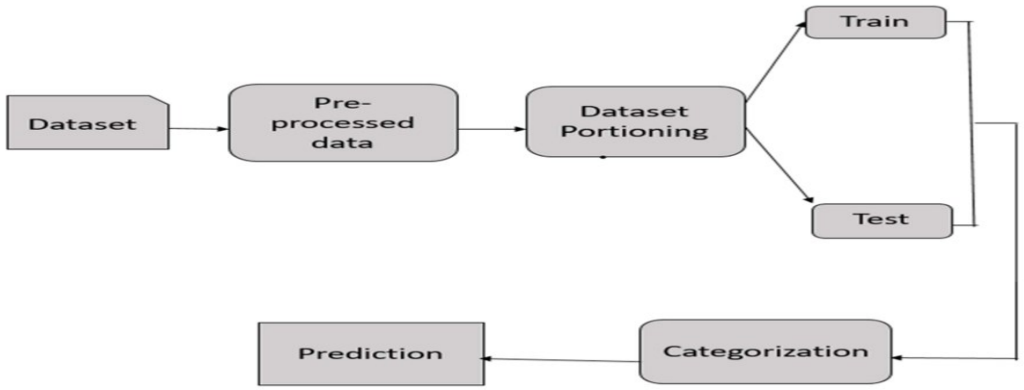
INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

6. MODULES

- **Choose and Import Data:** Determining the appropriate data type and source, as well as suitable instruments to collect data. The KDD Cup 99 dataset is used.
- **Data Preprocessing:** Handling irrelevant and missing parts, including missing data (ignoring tuples, filling values) and encoding categorical data (Count Vectorizer).
- **Dataset Portioning:** Splitting available data into training and testing sets for cross-validation.
- **Categorizing:** Identifying to which of a set of categories a new observation belongs using Random Forest, Decision Trees, and XGBoost.
    - **Random Forest:** Ensemble learning method constructing multiple decision trees.
    - **Decision Trees:** Supervised learning where data is continuously split according to a certain parameter.
- **XG Boost Model:** Scalable, distributed gradient-boosted decision tree (GBDT) machine learning library.
- **Prediction:** Using trained models to predict outcomes. Evaluation metrics include Accuracy, Precision, Recall, ROC, and Confusion Matrix.
- **Generating Results:** Final results based on overall classification and prediction.

7. SYSTEM DESIGN

7.System Architecture



.

Fig-1

8. ALGORITHMS

Machine learning algorithms like Random Forest, Decision Tree, and XGBoost are used.

- **Random Forest Algorithm:**
    - **Description:** Ensemble learning method constructing multiple decision trees. Outputs mode of classes (classification) or mean prediction (regression).
    - **Application:** Identifying influential devices by analyzing features like network

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT
ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

traffic, system configurations, access logs.
    - **Benefits:** Robust against overfitting, handles high-dimensional data, provides feature importance.
- **Decision Tree Algorithm:**
    - **Description:** Supervised learning, partitions data into subsets based on attribute values, creates a tree structure.
    - **Application:** Classifying devices as influential or non-influential based on attributes like network behavior, system configurations.
    - **Benefits:** Easy to interpret and visualize.
- **XGBoost Algorithm:**
    - **Description:** Efficient, scalable gradient boosting. Builds decision trees sequentially, each correcting errors of the previous.
    - **Application:** Identifying influential devices, leveraging performance on large datasets and complex relationships.
    - **Benefits:** High predictive accuracy and speed, handles missing values, built-in regularization.

Project Steps using these algorithms:

1. **Data Collection:** Gather CPS data (network traffic, logs, configurations).
2. **Data Preprocessing:** Cleanse data, handle missing values, transform features.
3. **Feature Engineering:** Extract relevant features.
4. **Model Training:** Use Random Forest, Decision Tree, XGBoost.
5. **Model Evaluation:** Use metrics like accuracy, precision, recall, F1-score.

9. IMPLEMENTATION

(The document includes sample Python code snippets for importing libraries, data selection, preprocessing, feature selection (KMeans for SOM visualization), data splitting, and classification using Random Forest, Decision Tree, and XGBoost, including evaluation with confusion matrices and ROC curves, and a simple prediction interface using easygui and tkinter.)

Key Implementation Steps:

1. **Import Libraries:** numpy, pandas, sklearn (for train_test_split, accuracy_score, metrics, preprocessing, RandomForestClassifier, DecisionTreeClassifier), matplotlib.pyplot, seaborn, xgboost.XGBClassifier, easygui, tkinter.

2. **Data Loading and Initial Inspection:** Load "CPSdataset.csv".
3. **Preprocessing:**
   - Handle missing values (e.g., fillna(0)).
   - Label encode categorical features (proto, service, state).
4. **Feature Selection/Visualization:** Use KMeans for Self-Organizing Map (SOM) visualization of data clusters.
5. **Data Splitting:** Split data into training (80%) and testing (20%) sets.
6. **Model Training and Evaluation:**
   - **Random Forest:** Train RandomForestClassifier, predict, calculate accuracy, classification report, confusion matrix, ROC curve.
   - **Decision Tree:** Train DecisionTreeClassifier, predict, calculate accuracy, classification report, confusion matrix, ROC curve.
   - **XGBoost:** Train XGBClassifier, predict, calculate accuracy, classification report, confusion matrix, ROC curve.
7. **Prediction Interface:** A simple GUI using easygui to input a CPS ID and tkinter to display if it's an "ANOMALY" or "NON ANOMALY".

10. TESTING OF PRODUCT

System testing ensures the system works accurately and efficiently.

- **Unit Testing:** Testing individual modules.
- **Integration Testing:** Testing combined modules (Top-down, Bottom-up).
- **White Box Testing:** Uses control structure for test cases.
- **Black Box Testing:** Finds incorrect/missing functions, interface errors, performance errors.
- **Validation Testing:** Ensures software functions as expected by the customer.
- **User Acceptance Testing:** Key for system success.
- **Output Testing:** Ensures output is in the required format.

Test Cases Example:
(A table of test cases is provided in the original document, including TC-ID, Description, Expected Outcome, Actual Outcome, Pass/Fail for verifying influential device identification, metadata availability, vulnerability assessment, security measure implementation, and real-time anomaly detection.)

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025

11. OUTPUT SCREENSHOTS

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
4	3	1.62313	tcp	-	FIN	8	16	364	13186	14.1702	62	252	1572.27	60929.2	1
5	4	1.68164	tcp	ftp	FIN	12	12	628	770	13.6771	62	252	2740.18	3358.62	1
6	5	0.44945	tcp	-	FIN	10	6	534	268	33.3738	254	252	8561.5	3987.06	2
7	6	0.38054	tcp	-	FIN	10	6	534	268	39.418	254	252	10112	4709.13	2
8	7	0.63711	tcp	-	FIN	10	8	534	354	26.683	254	252	6039.78	3892.58	2
9	8	0.52158	tcp	-	FIN	10	8	534	354	32.593	254	252	7377.53	4754.75	2
10	9	0.54291	tcp	-	FIN	10	8	534	354	31.313	254	252	7087.8	4568.02	2
11	10	0.25869	tcp	-	FIN	10	6	534	268	57.9851	254	252	14875.1	6927.29	2
12	11	0.30485	tcp	-	FIN	12	6	4142	268	55.7646	254	252	99641.5	5878.24	3
13	12	2.09309	tcp	smtp	FIN	62	28	56329	2212	42.521	62	252	211825	8152.56	28
14	13	0.41695	tcp	-	FIN	10	6	534	268	35.9754	254	252	9228.88	4297.86	2
15	14	0.99622	tcp	-	FIN	10	8	564	354	17.0645	254	252	4079.42	2489.41	2
16	15	0.57676	tcp	-	FIN	10	8	534	354	29.4753	254	252	6671.81	4299.92	2
17	16	2E-06	udp	snmp	INT	2	0	138	0	500000	254	0	2.8E+08	0	0
18	17	0.72825	tcp	-	FIN	10	6	534	268	20.5973	254	252	5283.89	2460.69	2
19	18	0.39356	tcp	http	FIN	10	8	860	1096	43.1959	62	252	15733.5	19494	2
20	19	0.38785	tcp	-	FIN	10	6	534	268	38.6745	254	252	9921.31	4620.32	2
21	20	0.53784	tcp	-	FIN	10	8	534	354	31.6079	254	252	7154.54	4611.04	2
22	21	0.23372	tcp	-	FIN	10	6	534	268	64.1794	254	252	16464.1	7667.29	2
23	22	0.33802	tcp	http	FIN	10	6	998	268	44.3765	254	252	21277	5301.51	2
24	23	0.96466	tcp	ftp	CON	14	12	690	950	25.916	62	252	5315.88	7223.3	5
25	24	0.4497	tcp	-	FIN	10	6	738	268	33.3553	254	252	11830	3984.85	2
26	25	0.92103	tcp	-	FIN	10	6	534	268	16.2862	254	252	4177.95	1945.66	2
27	26	0.60098	tcp	-	FIN	10	8	534	354	28.287	254	252	6402.85	4126.58	2
28	27	0.56895	tcp	-	FIN	10	6	534	268	26.3646	254	252	6763.4	3149.69	2

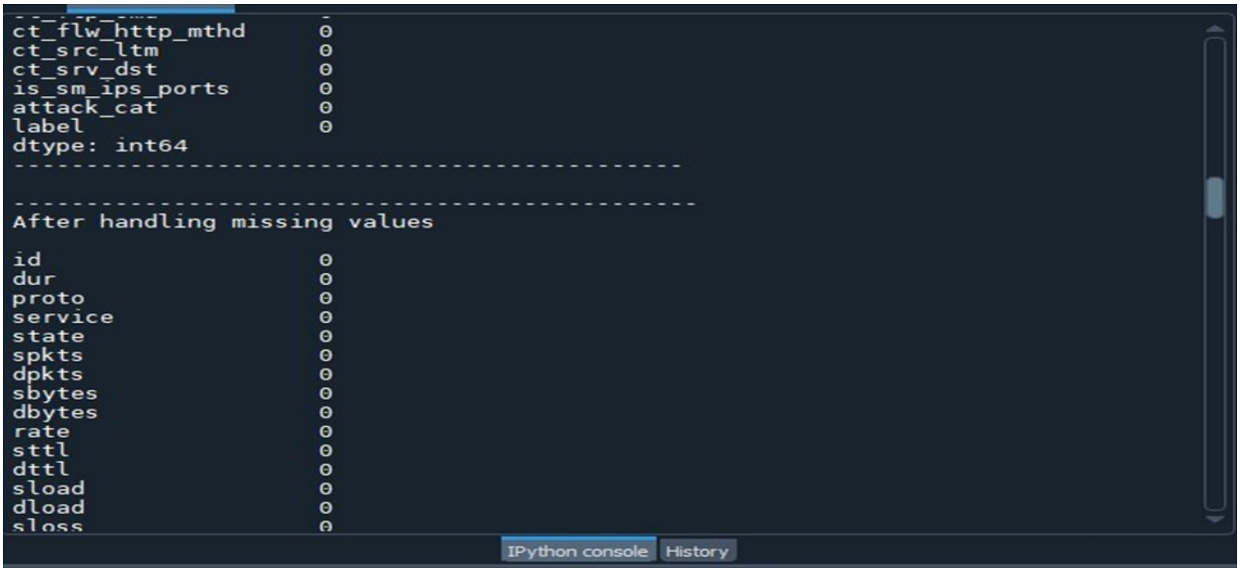
CPSdataset

Screenshot 1: Sample Data Set (Excel view)

```
Data Selection
Samples of our input data
   id       dur proto service  ... ct_srv_dst  is_sm_ips_ports  attack_cat
label
0   1  0.121478   tcp       -  ...          1                0      Normal
0
1   2  0.649902   tcp       -  ...          6                0      Normal
0
2   3  1.623129   tcp       -  ...          6                0      Normal
0
3   4  1.681642   tcp     ftp  ...          1                0      Normal
0
4   5  0.449454   tcp       -  ...         39                0      Normal
0
5   6  0.380537   tcp       -  ...         39                0      Normal
0
6   7  0.637109   tcp       -  ...         39                0      Normal
0
7   8  0.521584   tcp       -  ...         39                0      Normal
0
8   9  0.542905   tcp       -  ...         39                0      Normal
0
9  10  0.258687   tcp       -  ...         39                0      Normal
0

[10 rows x 45 columns]
```

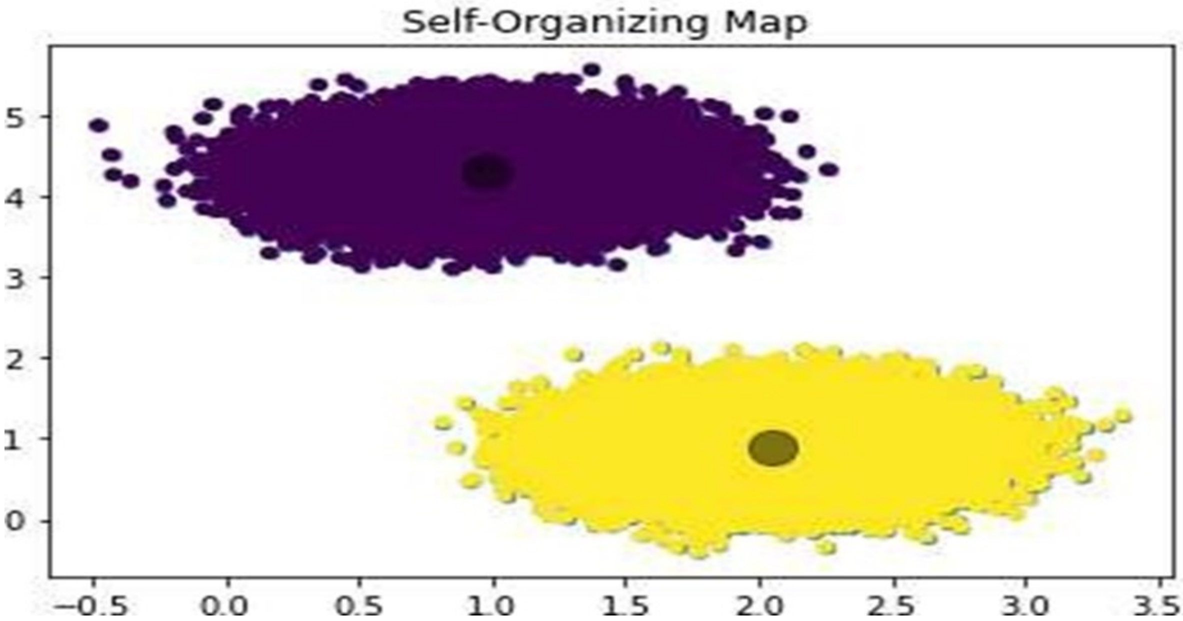
Screenshot 2 : Data Selection (console output)

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025



```
ct_flw_http_mthd      0
ct_src_ltm            0
ct_srv_dst            0
is_sm_ips_ports       0
attack_cat            0
label                 0
dtype: int64
----------------------------------------------

----------------------------------------------
After handling missing values

id                    0
dur                   0
proto                 0
service               0
state                 0
spkts                 0
dpkts                 0
sbytes                0
dbytes                0
rate                  0
sttl                  0
dttl                  0
sload                 0
dload                 0
sloss                 0
```


Screenshot 3: preprocessing steps (console output)



Screenshot 4: Self- Organizing Map (plot) Organizing Map (plot)

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

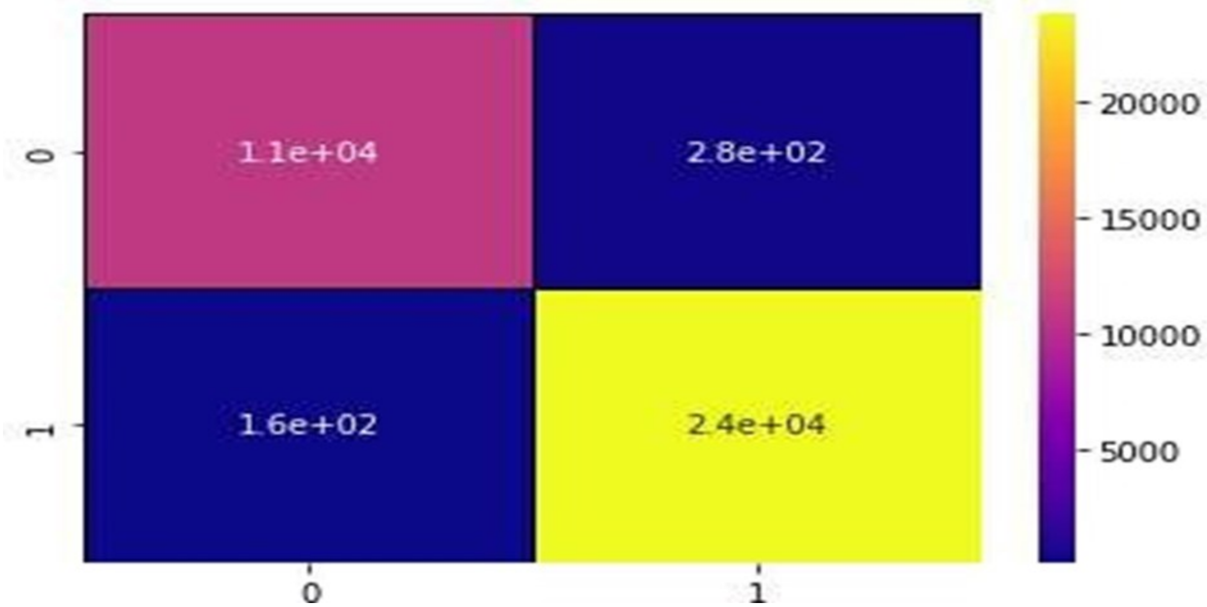
```
==================================================
 Preprocessing
==================================================
3.Data feature Selection
==================================================

                           Warning

     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.
 4.Data Splitting
==================================================
X_train Shapes  (140272, 20)
y_train Shapes  (140272,)
x_test Shapes  (35069, 20)
y_test Shapes  (35069,)
4.Data Classification --Unsupervised Machine Learning
----------------------------------------------------------------------
4.Data Classification --1.Random Forest Algorithm
==================================================

----------------------------------------------------------------------
```

Screenshot 5: Data feature Selection and Data Splitting (console output)

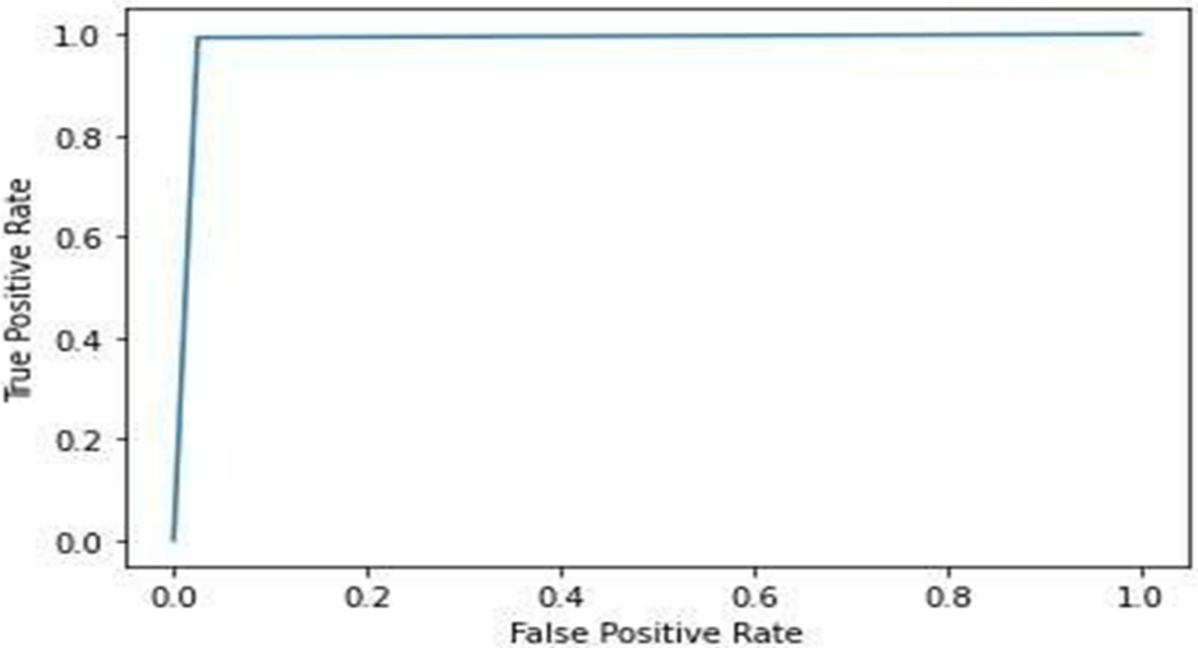
```
==================================================

----------------------------------------------------------------------
Random Forest

              precision    recall  f1-score   support

           0       0.99      0.98      0.98     11169
           1       0.99      0.99      0.99     23900

    accuracy                           0.99     35069
   macro avg       0.99      0.98      0.99     35069
weighted avg       0.99      0.99      0.99     35069


Random Forest Accuracy is: 98.73962759131997 %

Random Forest Confusion Matrix:
[[10891   278]
 [  164 23736]]
-------------------------------------------------------
```

Screenshot 6: Data Classification: Random Forest Algorithm (console output - precision, recall, F1-score, accuracy, confusion matrix)

ISSN 2454-9940
www.ijasem.org
Vol 19, Issue 2, 2025



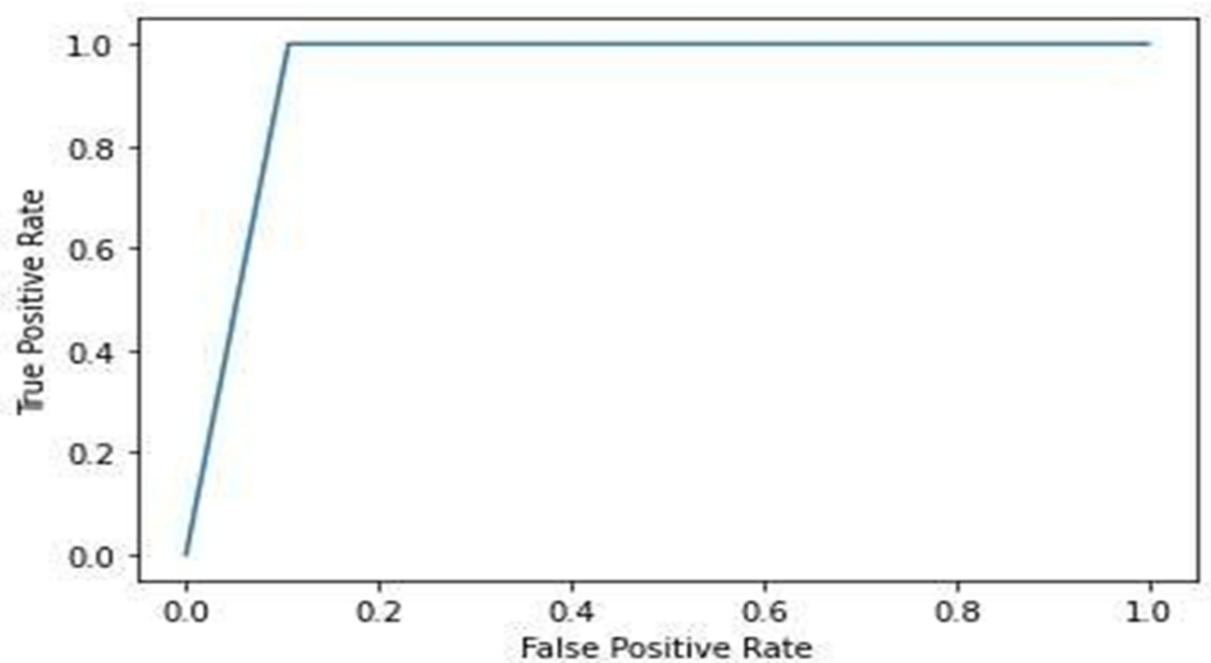
Screenshot 7: Confusion Matrix of Random Forest Algorithm (heatmap plot)



Screenshot 8: ROC Curve of Random Forest Algorithm (plot)

```
-------------------------------------------------------

4.Data Classification --2.Decision tree  Algorithm
==================================================


----------------------------------------------------------------------
Decision Tree

              precision    recall  f1-score   support

           0       1.00      0.89      0.94     11169
           1       0.95      1.00      0.98     23900

    accuracy                           0.97     35069
   macro avg       0.98      0.95      0.96     35069
weighted avg       0.97      0.97      0.97     35069


DT Accuracy is: 96.60098662636517 %
```

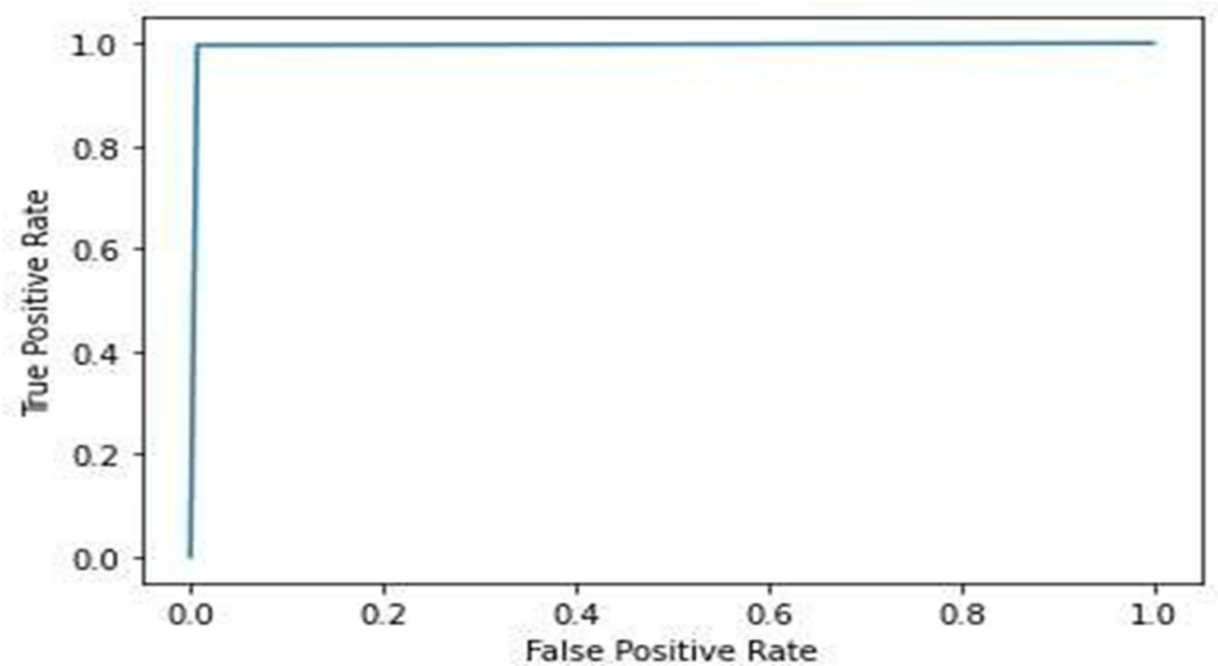
Screenshot 9: Data Classification: Decision Algorithm (console output)



Screenshot 10: ROC Curve of Decision Algorithm (plot)

```
----------------------------------------------------------------------
Xgboost Algorithm

              precision    recall  f1-score   support

           0       0.99      0.99      0.99     11169
           1       1.00      1.00      1.00     23900

    accuracy                           1.00     35069
   macro avg       0.99      0.99      0.99     35069
weighted avg       1.00      1.00      1.00     35069


Xgboost Algorithm  Accuracy is: 99.51524138127691 %

Xgboost Algorithm  Confusion Matrix:
[[11091    78]
 [   92 23808]]
-------------------------------------------------------
```

Screenshot 11: Data Classification: XGBoost Algorithm (console output)



Screenshot 12: ROC Curve of XGBoost Algorithm (plot)

Summary of Results from Screenshots (Approximate):

- **Random Forest Accuracy:** 98.73%
- **Decision Tree Accuracy:** 96.60%
- **XGBoost Accuracy:** 99.51%

XGBoost appears to perform the best in terms of accuracy.

12. CONCLUSION

The challenges for Cyber-Physical Systems include security and privacy. Anomaly detection is a crucial data analysis task for ensuring security in CPSs. This project introduced Machine Learning (ML) techniques for detecting anomalies, showcasing their effectiveness through a case study on classifying False Data Injection (FDI) attacks using Random Forest, Decision Tree, and XGBoost algorithms. The results demonstrate that these ML techniques, particularly XGBoost, can effectively classify attacks with high accuracy.

13. FUTURE ENHANCEMENT

- **Ensemble Learning Techniques:** Explore stacking or boosting to combine predictions from multiple ML algorithms for improved accuracy and robustness.
- **Deep Learning Architectures:** Investigate CNNs or RNNs to handle complex patterns in CPS data.
- **Online Learning and Adaptive Systems:** Develop mechanisms for continuous model updates based on real-time data streams.
- **Privacy-Preserving ML:** Implement federated learning or differential privacy.
- **Adversarial Robustness:** Incorporate adversarial training methods to enhance model resilience against attacks.
- **Dynamic Threat Intelligence Integration:** Integrate threat intelligence feeds for adaptive threat response.

14. REFERENCES

[1] Pathan A-SK, Azad S, Khan R, et al. Security mechanisms and data access protocols in innovative wireless networks. London: Sage; 2018.

[2] Yong-xiong Z, Liang-ming W, Lu-xia Y. A network attack discovery algorithm based on unbalanced sampling vehicle evolution strategy for intrusion detection. Int J Comput Appl. 2017: 1–9.

[3] Zargar ST, Joshi J, Tipper D. A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks. IEEE Commun Surv Tutorials. 2013;15(4):2046–2069.

[4] Toledo AL, Wang X. Robust detection of MAC layer denial-of-service attacks in CSMA/CA wireless networks. IEEE Trans Inf Forensics Secure. 2008;3(3):347–358.

[5] Guo Y, Ten CW, Hu S, et al. Modeling distributed denial of service attack in advanced metering infrastructure. 2015 IEEE power & energy society innovative smart grid technologies conference (ISGT); 2015. p. 1–5.

[6] A. A. Khan, M. H. Rehmani, and M. Reisslein, ''Cognitive radio for smart grids: Survey of architectures, spectrum sensing mechanisms, and networking protocols,'' IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 860–898, 1st Quart., 2016.

[7] Y. Lin, X. Zhu, Z. Zheng, Z. Dou, and R. Zhou, ''The individual identification method of wireless device based on dimensionality reduction,'' J. Supercomput., vol. 75, no. 6, pp. 3010– 3027, Jun. 2019.

[8] T. Liu, Y. Guan, and Y. Lin, ''Research on modulation recognition with ensemble learning,'' EURASIP J. Wireless Commun. Netw., vol. 2017, no. 1, p. 179, 2017.

[9] Y. Tu, Y. Lin, J. Wang, and J.-U. Kim, ''Semi-supervised learning with generative adversarial networks on digital signal modulation classification,'' Comput. Mater. Continua, vol. 55, no. 2, pp. 243–254, 2018.