

ISSN 1989-9572

DOI:10.47750/jett.2021.12.01.030

An Imbalance-Aware Approach for Cyber-Attack Detection in IoT-Enabled Cyber-Physical Systems using Autoencoders and Deep Learning

M. Ramana Kumar,

Journal for Educators, Teachers and Trainers, Vol.12(1)

<https://jett.labosfor.com/>

Date of reception: 10 Dec 2020

Date of revision: 1 Jan 2021

Date of acceptance: 29 Jan 2021

M. Ramana Kumar (2021). An Imbalance-Aware Approach for Cyber-Attack Detection in IoT-Enabled Cyber-Physical Systems using Autoencoders and Deep Learning. Journal for Educators, Teachers and Trainers, Vol.12(1). 247-254.



An Imbalance-Aware Approach for Cyber-Attack Detection in IoT-Enabled Cyber-Physical Systems using Autoencoders and Deep Learning

M. Ramana Kumar,

Assistant Professor, Dept. of CSE, Shadan Women's College of Engineering & Technology, Hyderabad, Telangana. dmrkumar2020@rediffmail.com

ABSTRACT

Cyber-physical systems enabled by the Internet of Things (IoT) facilitate the transmission and reception of data over the internet, encompassing industrial equipment and operational IT. This equipment will be equipped with sensors to monitor its condition and transmit data to a centralized server via an internet connection. Malicious users may occasionally target sensors, compromising their integrity by altering data. This manipulated information can then be transmitted to a centralized server, resulting in erroneous actions being executed. Many countries' equipment and production systems have failed due to inaccurate data. Consequently, numerous algorithms have been developed to detect attacks. However, these algorithms often encounter issues related to data imbalance, where one class may contain a significantly larger number of records (for instance, NORMAL records) compared to the other class, such as attacks, which may have only a few records. This imbalance can result in detection algorithms failing to make accurate predictions. To address data imbalance, current algorithms utilize OVER and UNDER sampling techniques, which generate new records exclusively for the minority class. To address this issue, we are introducing a novel technique that does not utilize any under or oversampling algorithms. The proposed technique is comprised of two components. An autoencoder will be trained on an imbalanced dataset to extract features. These extracted features will subsequently be utilized to train a decision tree algorithm for predicting labels for both known and unknown attacks. The decision tree is trained using a reduced set of features derived from the principal component analysis (PCA) algorithm. Deep Neural Networks (DNNs) are trained on both known and unknown attacks. If any records contain an attack signature, DNN will identify the corresponding attack label or class and attribute them accordingly.

Keywords: Cyber-attacks, Internet of Things, Water Subsystem, Machine Learning, Principal Component Analysis, Deep Neural Networks.

1. INTRODUCTION

Sensors are utilized in a wide array of applications, including the measurement of body parameters and the facilitation of automated driving. Sensors are integral to executing detection and vision-related tasks across contemporary applications in science, engineering, and technology, particularly in areas where computer vision is prevalent. An emerging domain that utilizes smart sensors is the Internet of Things (IoT), which involves wireless networks and distributed sensors that collect data in real time and generate specific outcomes of interest through appropriate processing methods. In devices utilizing the Internet of Things (IoT), sensors and artificial intelligence (AI) serve as the critical components that enable these devices to exhibit sensory and intelligent capabilities. The integration of AI enables sensors to function as smart sensors, optimizing their application across various domains. These include general environmental monitoring, the assessment of multiple environmental factors, weather forecasting, satellite imaging, remote sensing applications, monitoring of hazard events such as landslide detection, autonomous vehicles, and healthcare, among others. In relation to this latter sector, there has been a significant increase in the utilization of smart devices within hospitals and diagnostic centers for the evaluation and monitoring of various health conditions in affected patients, both remotely and physically [2].

In practical terms, no field of science or research operates efficiently without the utilization of modern sensors. The extensive application and necessity of sensors, along with the integration of IoT in remote sensing, environmental monitoring, and human health assessment, contribute to the intelligence of these applications. Over the past ten years, agricultural applications have incorporated the use of various sensors to monitor and control a range of environmental parameters, including temperature, humidity, soil quality, pollution, air quality, water contamination, and radiation. This document aims to emphasize the application of sensors and IoT technologies in remote sensing and agricultural applications through comprehensive discussion and review.

In recent years, the structural health monitoring (SHM) of civil structures has emerged as a significant area of research. Structural Health Monitoring (SHM) is utilized to identify damage within a structure and offers early warnings regarding the safety status of the structure for usage. Civil infrastructure, such as bridges, experiences deterioration over time due to factors including heavy vehicle loads, environmental changes, and dynamic forces like seismic activity. Changes of this nature primarily take place in existing structures that were built in the past, and a range of methods is available to identify such damage. The strategy of Structural Health Monitoring (SHM) entails the observation of the structure over a defined period to assess its condition. Periodic data measurements will be collected, and features will be extracted from these computational results. The analysis process can then utilize the extracted features to determine the current health status of the structure. The data gathered from the process can be periodically updated to assess the structure. Based on the information obtained through monitoring, the structure can be reinforced and repaired, and rehabilitation and maintenance activities can be executed [5].

2. LITERATURE SURVEY

Ullo et. al [6] focused on an extensive study of the advances in smart sensors and IoT, employed in remote sensing and agriculture applications such as the assessment of weather conditions and soil quality; the crop monitoring; the use of robots for harvesting and weeding; the employment of drones. The emphasis has been given to specific types of sensors and sensor technologies by presenting an extensive study, review, comparison and recommendation for advancements in IoT that would help researchers, agriculturists, remote sensing scientists and policy makers in their research and implementations.

Sivasuriyan et. al [7] provides a detailed understanding of bridge monitoring, and it focuses on sensors utilized and all kinds of damage detection (strain, displacement, acceleration, and temperature) according to bridge nature (scour, suspender failure, disconnection of bolt and cables, etc.) and environmental degradation under static and dynamic loading. This paper presents information about various methods, approaches, case studies, advanced technologies, real-time experiments, stimulated models, data acquisition, and predictive analysis. Future scope and research also discussed the implementation of SHM in bridges. The main aim of this research is to assist researchers in better understanding the monitoring mechanism in bridges.

Dazhe Zhao et. al [8] proposed an easy-fabricated and compact untethered triboelectric patch with Polytetrafluoroethylene (PTFE) as triboelectric layer and human body as conductor. We find that the conductive characteristic of human body has negligible influence on the outputs, and the untethered triboelectric patch has good output ability and robustness. The proposed untethered triboelectric patches can work as sensor patches and energy harvester patches. Three typical applications are demonstrated, which are machine learning assisted objects distinguishing with accuracy up to 93.09–94.91 %, wireless communication for sending typical words to a cellphone, and human motions energy harvesting for directly powering electronics or charging an energy storage device.

Bacco et. al [9] described, both analytically and empirically, a real testbed implementing IEEE 802.15.4-based communications between an UAV and fixed ground sensors. In our scenario, we found that aerial mobility limits the actual IEEE 802.15.4 transmission range among the UAV and the ground nodes to approximately 1/3 of the nominal one. We also provide considerations to design the deployment of sensors in precision agriculture scenarios.

Verma et. al [10] discussed the existing state-of-the-art practices of improved intelligent features, controlling parameters and Internet of things (IoT) infrastructure required for smart building. The main focus is on sensing, controlling the IoT infrastructure which enables the cloud clients to use a virtual sensing infrastructure using communication protocols. The following are some of the intelligent features that usually make building smart such as privacy and security, network architecture, health services, sensors for sensing, safety, and overall management in smart buildings. As we know, the Internet of Things (IoT) describes the ability to connect and control the appliances through the network in smart buildings. The development of sensing technology, control techniques, and IoT infrastructure give rise to a smart building more efficient. Therefore, the new and problematic innovation of smart buildings in the context of IoT is to a great extent and scattered. The conducted review organized in a scientific manner for future research direction which presents the existing challenges, and drawbacks.

Hu et. al [11] presented a real-time, fine-grained, and power-efficient air quality monitor system based on aerial and ground sensing. The architecture of this system consists of the sensing layer to collect data, the transmission

layer to enable bidirectional communications, the processing layer to analyze and process the data, and the presentation layer to provide a graphic interface for users. Three major techniques are investigated in our implementation for data processing, deployment strategy, and power control. For data processing, spatial fitting and short-term prediction are performed to eliminate the influences of incomplete measurement and the latency of data uploading. The deployment strategies of ground sensing and aerial sensing are investigated to improve the quality of the collected data. Power control is further considered to balance between power consumption and data accuracy. Our implementation has been deployed in Peking University and Xidian University since February 2018, and has collected almost 100,000 effective values thus far.

Famila et. al [12] proposed an Improved Artificial Bee colony optimization based ClusTering(IABCOCT) algorithm by utilizing the merits of Grenade Explosion Method (GEM) and Cauchy Operator. This incorporation of GEM and Cauchy operator prevents the Artificial Bee Colony (ABC) algorithm from stuck into local optima and improves the convergence rate. The benefits of GEM and Cauchy operator are embedded into the Onlooker Bee and scout bee phase for phenomenal improvement in the degree of exploitation and exploration during the process of CH selection. The simulation results reported that the IABCOCT algorithm outperforms the state of art methods like Hierarchical Clustering-based CH Election (HCCHE), Enhanced Particle Swarm Optimization Technique (EPSOCT) and Competitive Clustering Technique (CCT) in-terms of different measures such as throughput, packet loss, delay, energy consumption and network lifetime.

3. PROPOSED SYSTEM

Figure 4.1 shows the proposed system model with detailed design. The steps involved in working with the SWAT (Soil and Water Assessment Tool) water IoT dataset for cyber-physical attack prediction, including dataset preprocessing, PCA (Principal Component Analysis) feature extraction, using a Deep Neural Network (DNN) classifier, and performance evaluation. The detailed operation illustrated as follows:

Step 1: SWAT Water IoT Dataset: Start with obtaining and understanding the SWAT water IoT dataset, which contains data related to water quality, environmental parameters, and IoT sensor readings.

Step 2: Dataset Preprocessing:

Data Cleaning: Handle missing data points by either imputation or removal, depending on the extent of missingness and domain knowledge. Address outliers, if present, by applying appropriate techniques (e.g., winsorization or outlier removal).

Feature Engineering: Engineer relevant features, such as aggregating IoT sensor data over time intervals, calculating statistics, or creating new variables based on domain expertise. Convert categorical variables into numerical format (e.g., one-hot encoding).

Data Splitting: Split the dataset into training, validation, and testing sets to facilitate model training, hyperparameter tuning, and evaluation.

Normalization or Standardization: Normalize or standardize numerical features to ensure they have a similar scale, which can improve the convergence and performance of the DNN.

Step 3: PCA Feature Extraction: Apply PCA to reduce the dimensionality of the dataset while retaining the most important information. Select the appropriate number of principal components based on the explained variance ratio or cross-validation.

Step 4: Deep Neural Network (DNN) Classifier:

Model Architecture: Design the architecture of a DNN classifier. The architecture may include input layers, hidden layers, activation functions, and output layers tailored to the dataset and the problem.

Hyperparameter Tuning: Tune hyperparameters of the DNN model, such as the number of hidden layers, units per layer, learning rate, batch size, and dropout rates, using techniques like grid search or random search.

Training: Train the DNN model on the preprocessed and PCA-transformed training dataset using an appropriate optimization algorithm (Adam). Monitor training progress using metrics like loss and accuracy on the validation set and implement early stopping if needed.

Step 5: Cyber-Physical Attack Prediction: Apply the trained DNN model to predict cyber-physical attacks on the testing dataset.

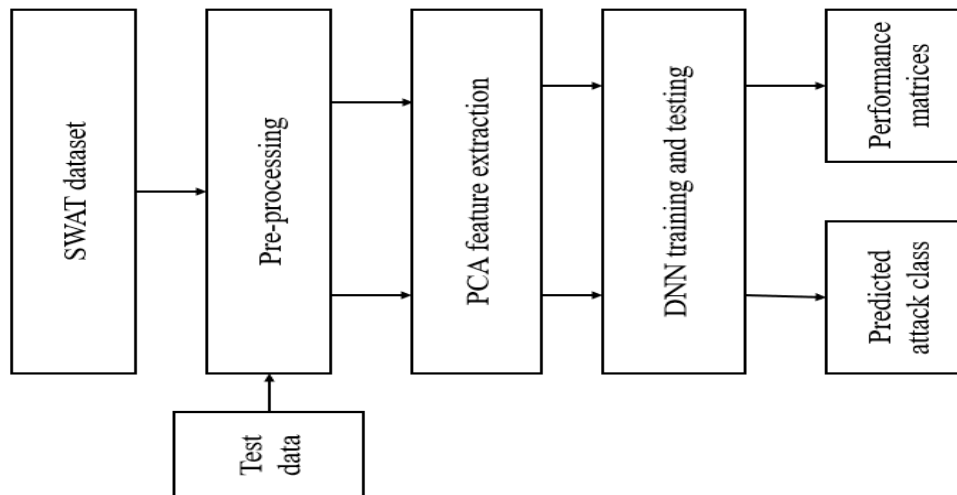


Fig.1: Block diagram of proposed system.

Data Preprocessing

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

DNN

Although today the Perceptron is widely recognized as an algorithm, it was initially intended as an image recognition machine. It gets its name from performing the human-like function of perception, seeing, and recognizing images. Interest has been centered on the idea of a machine which would be capable of conceptualizing inputs impinging directly from the physical environment of light, sound, temperature, etc. — the “phenomenal world” with which we are all familiar — rather than requiring the intervention of a human agent to digest and code the necessary information. Rosenblatt’s perceptron machine relied on a basic unit of computation, the neuron. Just like in previous models, each neuron has a cell that receives a series of pairs of inputs and weights. The major difference in Rosenblatt’s model is that inputs are combined in a weighted sum and, if the weighted sum exceeds a predefined threshold, the neuron fires and produces an output.

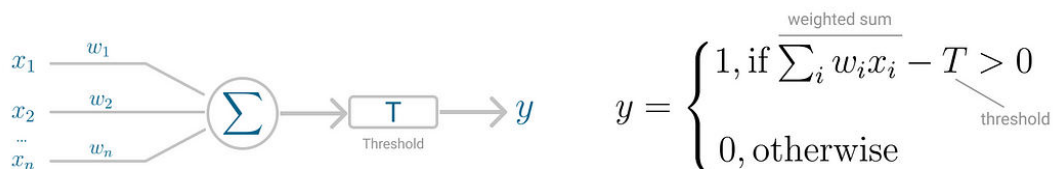


Fig. 2: Perceptron neuron model (left) and threshold logic (right).

Threshold T represents the activation function. If the weighted sum of the inputs is greater than zero the neuron outputs the value 1, otherwise the output value is zero.

4. RESULTS AND DISCUSSION

Figure 3 displays a portion of the dataset used in this study. It provides an actual glimpse into the data, showcasing several rows and columns. This figure serves as a representative sample, illustrating the format and structure of the dataset under analysis.

Figure 4 showcases the dataset after undergoing preprocessing steps. It provides an overview of how the data has been divided into training and testing subsets. This figure includes information about the number of samples allocated for training and testing purposes, highlighting the preparation of the dataset for subsequent analysis. The total records found in the dataset is 27199, and total features found in the dataset is 23. Further, 80% of dataset is used for training, which contains the 21799 number of records. In addition, 20% of dataset is used for testing, which contains the 5440 number of records.

| A1 | | | command_address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|---------|----------|-----------------|----------|---------|----------|------|-----|------|-----|------|------|------|-------|-----|------|---------|------|--------|----|---|----------|------|---------|---|---------|----|------|-----|------|---------|------|------|
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | | | | | | | | | | |
| 1 | command | response | command | response | command | response | comm | res | comm | wri | resp | read | resp | write | sub | func | command | resp | length | HH | H | L | LL | control | m | control | sc | pump | crc | rate | measure | time | 1 |
| 2 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.75896 | 1.07 | | | | | | | | | | |
| 3 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.67368 | 1.07 | | | | | | | | | | |
| 4 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.61683 | 1.16 | | | | | | | | | | |
| 5 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.55988 | 1.1 | | | | | | | | | | |
| 6 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.4747 | 1.15 | | | | | | | | | | |
| 7 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.38942 | 1.29 | | | | | | | | | | |
| 8 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.33257 | 1.27 | | | | | | | | | | |
| 9 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.21888 | 1.22 | | | | | | | | | | |
| 10 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.10517 | 1.26 | | | | | | | | | | |
| 11 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 2 | 1 | 0 | 1 | 85.07675 | 1.04 | | | | | | | | | | |
| 12 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 13 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 14 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.19 |
| 15 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 16 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.01 |
| 17 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.15 |
| 18 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.01 |
| 19 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 20 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.22 |
| 21 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.01 |
| 22 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.25 |
| 23 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.01 |
| 24 | 7 | 7 | 183 | 233 | 9 | 10 | 3 | 10 | 3 | 10 | 0 | 25 | 21 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.28 |
| 25 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1.01 |
| 26 | 7 | 0 | 183 | 0 | 9 | 0 | 3 | 10 | 3 | 0 | 0 | 25 | 123 | 90 | 80 | 20 | 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| < > | | | swat_dataset + | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 3: Sample dataset.

```
[[0.06122449 1. 0. ... 0. 0.2 0.73333333]
[0.06122449 1. 0. ... 0. 0.2 0.76666667]
[0.06122449 0. 0. ... 1. 0.2 0.03333333]
...
[0.06122449 1. 0. ... 0. 0.2 0.26666667]
[0.06122449 0. 0. ... 1. 0.2 0. ]
[0.06122449 1. 0. ... 0. 0.2 0.46666667]]
```

Total records found in dataset : 27199
Total features found in dataset: 23

Dataset Train and Test Split

80% dataset records used to train ML algorithms : 21759
20% dataset records used to train ML algorithms : 5440

Figure.4: Preprocessed dataset with training and testing count.

Figure 5 is focused on estimating the performance of the cyber-attack detection model. It likely presents metrics and results related to the model's accuracy, precision, recall, F1-score, or other relevant evaluation measures. This figure helps assess the effectiveness of the detection system. Here, the proposed DNN resulted in superior performance as compared to existing DTC and DAE classifiers

Figure 6 demonstrates the outcomes of the cyber-attack detection process. It likely showcases a comparison between actual attack instances and the model's predictions. This figure aids in visualizing the model's success in identifying and classifying cyber-attacks based on the predicted data.

```

AutoEncoder Accuracy : 89.63235294117648
AutoEncoder Precision : 73.40893703452946
AutoEncoder Recall : 73.86363636363636
AutoEncoder FScore : 73.56900233735225

Decision Tree Trained on New Features Extracted from AutoEncoder
Decision Tree Accuracy : 90.36764705882354
Decision Tree Precision : 73.52111086023933
Decision Tree Recall : 74.64285714285714
Decision Tree FScore : 74.03290345468014

Attack Prediction using DNN
DNN Accuracy : 100.0
DNN Precision : 100.0
DNN Recall : 100.0
DNN FScore : 100.0

```

Figure.5: Performance estimation

```
New Test Data : [ 7. 7. 183. 233. 9. 10. 3. 10. 3. 10.
0. 25. 21. 90. 80. 20. 10. 0. 1. 0.
1. 0. 1.04]====> NO CYBER ATTACK DETECTED

New Test Data : [ 7. 0. 183. 0. 9. 0. 3. 10. 3. 0.
0. 25. 123. 90. 80. 20. 10. 0. 1. 0.
1. 0. 1.01]====> CYBER ATTACK DETECTED & Identified As : Denial of Service (DoS)

New Test Data : [ 7. 7. 183. 233. 9. 10. 3. 10. 3. 10.
0. 25. 21. 90. 80. 20. 10. 0. 1. 0.
1. 0. 1.04]====> NO CYBER ATTACK DETECTED

New Test Data : [ 7.00000000e+00 0.00000000e+00 1.83000000e+02 0.00000000e+00
9.00000000e+00 0.00000000e+00 3.00000000e+01 1.00000000e+01
3.00000000e+00 0.00000000e+00 0.00000000e+00 2.50000000e+01
1.23000000e+02 9.00000000e+01 8.00000000e+01 2.00000000e+01
1.00000000e+01 0.00000000e+00 1.00000000e+00 0.00000000e+00
1.00000000e+00 -8.58993869e+09 1.01000000e+00]====> CYBER ATTACK DETECTED & Identified As : Denial of Service (DoS)
```

Figure.6: Cyber-attack detection from predicted data.

Table 10.1 presents a performance comparison of different algorithms for cyber-attack detection. The table includes the following metrics for each algorithm:

Accuracy: The accuracy metric represents the overall correctness of the algorithm's predictions. It is the percentage of correctly classified instances out of the total instances. In this table:

- AutoEncoder achieved an accuracy of 89.63%.
- Decision reached an accuracy of 90.36%.
- DNN (Deep Neural Network) achieved a perfect accuracy of 100.0%, indicating that it correctly classified all instances.

Precision: Precision measures the ratio of true positive predictions (correctly identified cyber-attacks) to all positive predictions (instances classified as attacks). In this table:

- AutoEncoder achieved a precision of 73.40%.
- Decision reached a precision of 73.52%.
- DNN achieved a perfect precision of 100.0%, indicating that when it predicted an attack, it was always correct.

Recall: Recall, also known as sensitivity or true positive rate, measures the ratio of true positive predictions to all actual positive instances. In this table:

- AutoEncoder achieved a recall of 73.86%.
- Decision reached a recall of 74.64%.
- DNN achieved a perfect recall of 100.0%, meaning it correctly identified all actual attacks.

F1-SCORE: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. In this table:

- AutoEncoder achieved an F1-score of approximately 73.569.
- Decision reached an F1-score of approximately 74.032.
- DNN achieved a perfect F1-score of 100.0%, indicating an ideal balance between precision and recall.

Table1. Performance comparison.

| Algorithm Name | Accuracy | Precision | Recall | F1-SCORE |
|----------------|----------|-----------|--------|----------|
| AutoEncoder | 89.63 | 73.40 | 73.86 | 73.569 |
| Decision | 90.36 | 73.52 | 74.64 | 74.032 |
| DNN | 100.0 | 100.0 | 100.0 | 100.0 |

5. CONCLUSION

Internet of Things enabled cyber physical systems such as Industrial equipment's and operational IT to send and receive data over internet. This equipment's will have sensors to sense equipment condition and report to centralized server using internet connection. Sometime some malicious users may attack or hack such sensors and then alter their data and this false data will be report to centralized server and false action will be taken. Due to false data many countries equipment and production system got failed and many algorithms was developed to detect attack, but all these algorithms suffer from data imbalance (one class my contains huge records (for example Normal records and other class like attack may contains few records which lead to imbalance problem and detection algorithms may failed to predict accurately). To deal with data imbalance existing algorithms were using over and under sampling which will generate new records for fewer class, but this technique improve accuracy but not up to the mark. Therefore, to overcome from this issue, this project introduced an efficient deep learning model without using any under or oversampling algorithms with the usage of auto encoder, decision

tree with PCA, and DNN for identifying the attack and classify the type of attack. In addition, the performance evaluation of three models also compared and proven that proposed DNN obtained enhanced accuracy 99.98%.

REFERENCES

- [1] Kayad, A.; Paraforos, D.; Marinello, F.; Fountas, S. Latest advances in sensor applications in agriculture. *Agriculture* 2020, 10, 362.
- [2] Elahi, H.; Munir, K.; Eugeni, M.; Atek, S.; Gaudenzi, P. Energy harvesting towards self-powered IoT devices. *Energies* 2020, 13, 5528.
- [3] Ullo, S.L.; Sinha, G.R. Advances in smart environment monitoring systems using IoT and sensors. *Sensors* 2020, 20, 3113.
- [4] Carminati, M.; Sinha, G.R.; Mohdiwale, S.; Ullo, S.L. Miniaturized pervasive sensors for indoor health monitoring in smart cities. *Smart Cities* 2021, 4, 146–155.
- [5] Ullo, S.L.; Addabbo, P.; Di Martire, D.; Sica, S.; Fiscante, N.; Cicala, L.; Angelino, C.V. Application of DInSAR technique to high coherence Sentinel-1 images for dam monitoring and result validation through in situ measurements. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* 2019, 12, 875–890.
- [6] Ullo, S.L. and Sinha, G.R., 2021. Advances in IoT and smart sensors for remote sensing and agriculture applications. *Remote Sensing*, 13(13), p.2585.
- [7] Sivasuriyan, A., Vijayan, D.S., LeemaRose, A., Revathy, J., Gayathri Monicka, S., Adithya, U.R. and Jebasingh Daniel, J., 2021. Development of smart sensing technology approaches in structural health monitoring of bridge structures. *Advances in Materials Science and Engineering*, 2021.
- [8] Dazhe Zhao, Kaijun Zhang, Yan Meng, Zhaoyang Li, Yucong Pi, Yujun Shi, Jiacheng You, Renkun Wang, Ziyi Dai, Bingpu Zhou, Junwen Zhong, Untethered triboelectric patch for wearable smart sensing and energy harvesting, *Nano Energy*, Volume 100, 2022, 107500, ISSN 2211-2855, <https://doi.org/10.1016/j.nanoen.2022.107500>.
- [9] M. Bacco, A. Berton, A. Gotta and L. Caviglione, "IEEE 802.15.4 Air-Ground UAV Communications in Smart Farming Scenarios," in *IEEE Communications Letters*, vol. 22, no. 9, pp. 1910-1913, Sept. 2018, doi: 10.1109/LCOMM.2018.2855211.
- [10] A. Verma, S. Prakash, V. Srivastava, A. Kumar and S. C. Mukhopadhyay, "Sensing, Controlling, and IoT Infrastructure in Smart Building: A Review," in *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9036-9046, 15 Oct.15, 2019, doi: 10.1109/JSEN.2019.2922409.
- [11] Z. Hu, Z. Bai, Y. Yang, Z. Zheng, K. Bian and L. Song, "UAV Aided Aerial-Ground IoT for Air Quality Sensing in Smart City: Architecture, Technologies, and Implementation," in *IEEE Network*, vol. 33, no. 2, pp. 14-22, March/April 2019, doi: 10.1109/MNET.2019.1800214.
- [12] Famila, S., Jawahar, A., Sariga, A. et al. Improved artificial bee colony optimization- based clustering algorithm for SMART sensor environments. *Peer-to-Peer Netw. Appl.* 13, 1071–1079 (2020). <https://doi.org/10.1007/s12083-019-00805-4>