**PROJECT REPORT**

**ON**

# **FALSE DATA INJECTION ATTACKS**

Submitted in partial fulfillment of the requirements

**Submitted By**

**P. AISHWARYA 245320748037**

**Under the guidance**

**Of**

Dr. Rajasekaran

**PhD, PMP**



**Department of CSE(AIML)**



NEIL GOGTE INSTITUTE OF TECHNOLOGY

A Unit of Keshav Memorial Technical Education (KMTES)

Approved by AICTE, New Delhi & Affiliated to Osmania University, Hyderabad

# **PROJECT SCHOOL CERTIFICATE**

**Title:** False Data Injection

**Mentor:** Dr. Rajasekaran

**Session Duration** :12Weeks [13thOct2022 - 12th Jan2023]

**Class:** CSM A

**Student Name:** P. AISHWARYA

**Roll No:** 245320748037

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Faculty Signature of student

# **PROJECT OBJECTIVE**

Predictive maintenance techniques are designed to help determine the condition of in-service equipment to estimate when maintenance should be performed. This approach promises cost savings over routine or time- based preventive maintenance because tasks are performed only when warranted. Thus, it is regarded as condition-based maintenance carried out as suggested by estimations of the degradation state of an item.

This project deals with the problem of false data injection. In this application, I am currently using “PDM Dataset”, Using this Dataset I am trying to predict the life cycle of ratio using the PdM method (predictive maintenance). The main objective comes into place when an external unauthorized user makes changes in dataset without knowledge of the person handling with the data, the predictions also get affected.

The project can be divided into following parts: -

* Prediction of the life cycle
* Making Graph of truly identified life cycles using CMAPPS Dataset.
* Injecting False Data into CMAPPS Dataset.
* Making another graph using the same method but using the false CMAPPS dataset.
* Making a website, displaying the graphs of different algorithms.

The above parts may further be classified into different parts.

## **TECHNOLOGY STACK**

The technologies used in this application are:

* Python 3.10.0
* MATLAB
* Django

The modules used in this application are:

* TensorFlow
* scikit-learn
* matplotlib
* pandas
* NumPy
* Keras
* Seaborn

The algorithms used in this application are:

* CNN (Convolutional Neural Networks)
* LSTM (Long Short Term Memory)
* GRU (Gated Recurrent Unit)

## **DEVELOPED SOLUTION**

**PdM DATASET:**

Dataset consists of multiple multivariate time series. Each data set is further divided into training and test subsets. Each time series is from a different engine i.e., the data can be considered to be from a fleet of engines of the same type. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user. This wear and variation are considered normal, i.e., it is not considered a fault condition. There are three operational settings that have a substantial effect on engine performance. These settings are also included in the data. The data is contaminated with sensor noise.

The engine is operating normally at the start of each time series, and develops a fault at some point during the series. In the training set, the fault grows in magnitude until system failure. In the test set, the time series ends some time prior to system failure. The objective of the competition is to predict the number of remaining operational cycles before failure in the test set, i.e., the number of operational cycles after the last cycle that the engine will continue to operate. Also provided a vector of true Remaining Useful Life (RUL) values for the test data.

The data are provided as a zip-compressed text file with 26 columns of numbers, separated by spaces. Each row is a snapshot of data taken during a single operational cycle, each column is a different variable. The columns correspond to:

1. unit number
2. time, in cycles
3. operational setting 1
4. operational setting 2
5. operational setting 3
6. sensor measurement 1
7. sensor measurement 2

…

26) sensor measurement 26

Using these sensors, we are going to predict its lifetime.

**PREDICTIVE MAINTENANCE**:

Predictive maintenance software uses data science and predictive analytics to estimate when a piece of equipment might fail so that corrective maintenance can be scheduled before the point of failure. The goal is to schedule maintenance at the most convenient and most cost-efficient moment, allowing equipment’s lifespan to be optimized to its fullest, but before the equipment has been compromised.

This predictive maintenance can be analyzed using the previous data of the equipment. CMAPPS dataset is a perfect example to make predictive maintenance of the cycle of the plane, using all the sensors.

**MAKING GRAPHS OF PdM USING PDM DATASET:**

I have used “Machine Learning” algorithms for predicting the condition of the cycle. To achieve this there are different steps:

 **DATA PROCESSING:**

This is the one of the most important step in this application. Using data pre-processing techniques my goal was to make labels whether the cycle is in good state or moderate state or worse state.

My 1st step was to remove all the unrequired features which would confuse the ml algorithm. “NumPy”, “Pandas”, are widely used in this process of preprocessing the data.

My next step was to create EOL Labels (End of life cycle Labels) for making this I used a sample formula, The formula for EOL is: “id – cycle – 1”.

Using this EOL values the next step was to create LR (Life Ratio), the formula I have used to make these Life Ratio Labels is “ current cycle/ current EOL of the cycle ” .

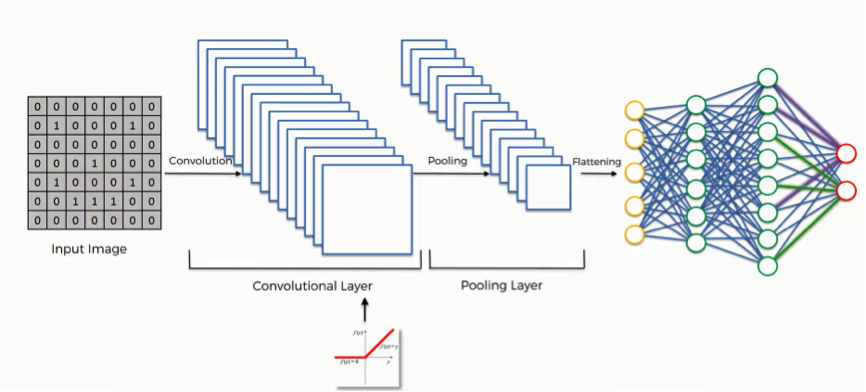
This EOL labels are there after used to label the cycle, the following labels are:

1. If LR<=0.6: Good
2. If LR lies in range of 0.6 and 0.8: Moderate
3. If LR>0.8: Warning

These labels are predicted using the machine learning models.

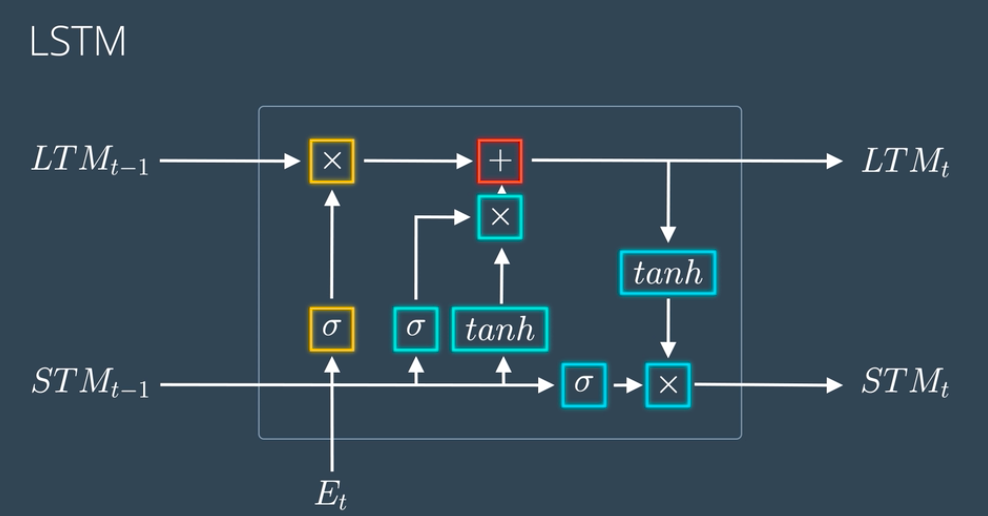
**CNN:**

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.



**LSTM:**

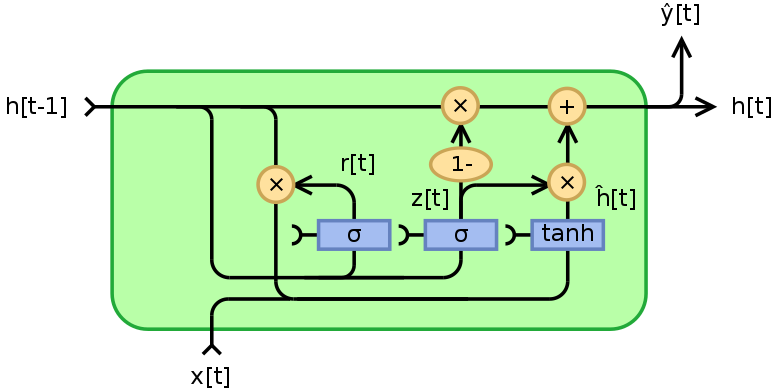
 LSTM stands for long short-term memory networks, used in the field of Deep Learning. It is a variety of RNN’S that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data, apart from single data points such as images. This finds application in speech recognition, machine translation, etc. LSTM is a special kind of RNN, which shows outstanding performance on a large variety of problems.

****

**GRU:**

GRU or Gated recurrent unit is an advancement of the standard RNN i.e recurrent neural network. It was introduced by [Kyunghyun Cho et al](https://arxiv.org/pdf/1406.1078v3.pdf) in the year 2014.

GRUs are very similar to Long Short Term Memory(LSTM). Just like LSTM, GRU uses gates to control the flow of information. They are relatively new as compared to LSTM. This is the reason they offer some improvement over LSTM and have simpler architecture. Another Interesting thing about  GRU is that, unlike LSTM, it does not have a separate cell state (Ct). It only has a hidden state(Ht). Due to the simpler architecture, GRUs are faster to train.

****

**SCREENSHOTS**

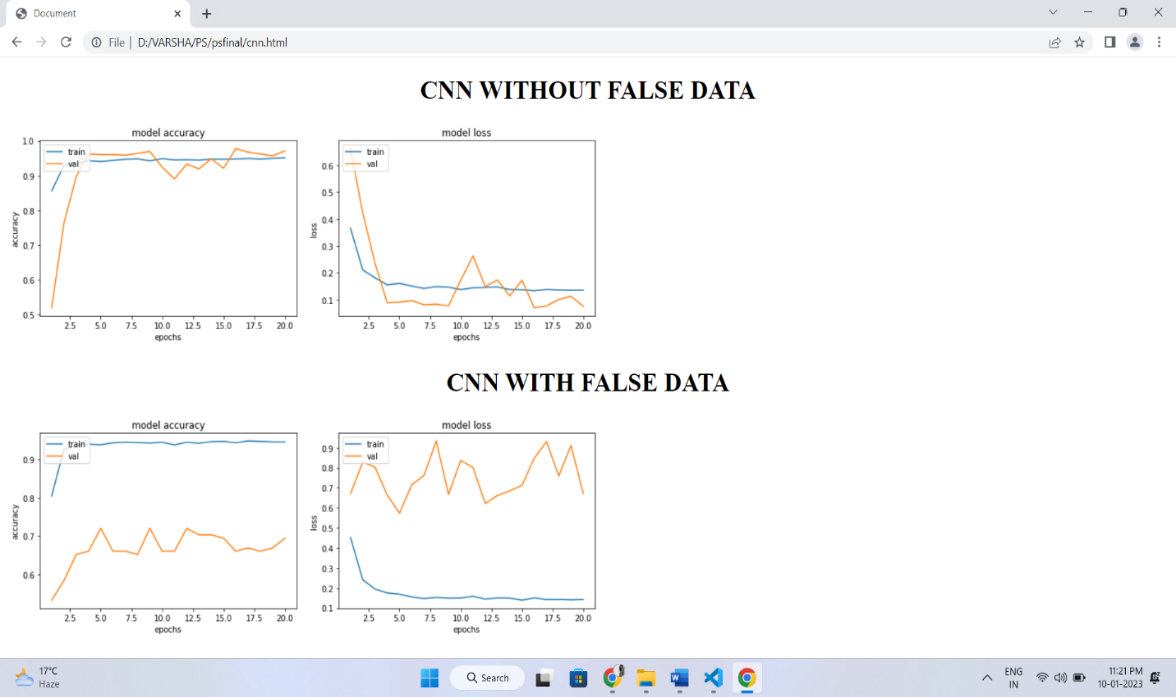


Figure 1: CNN with and without False Data Injection

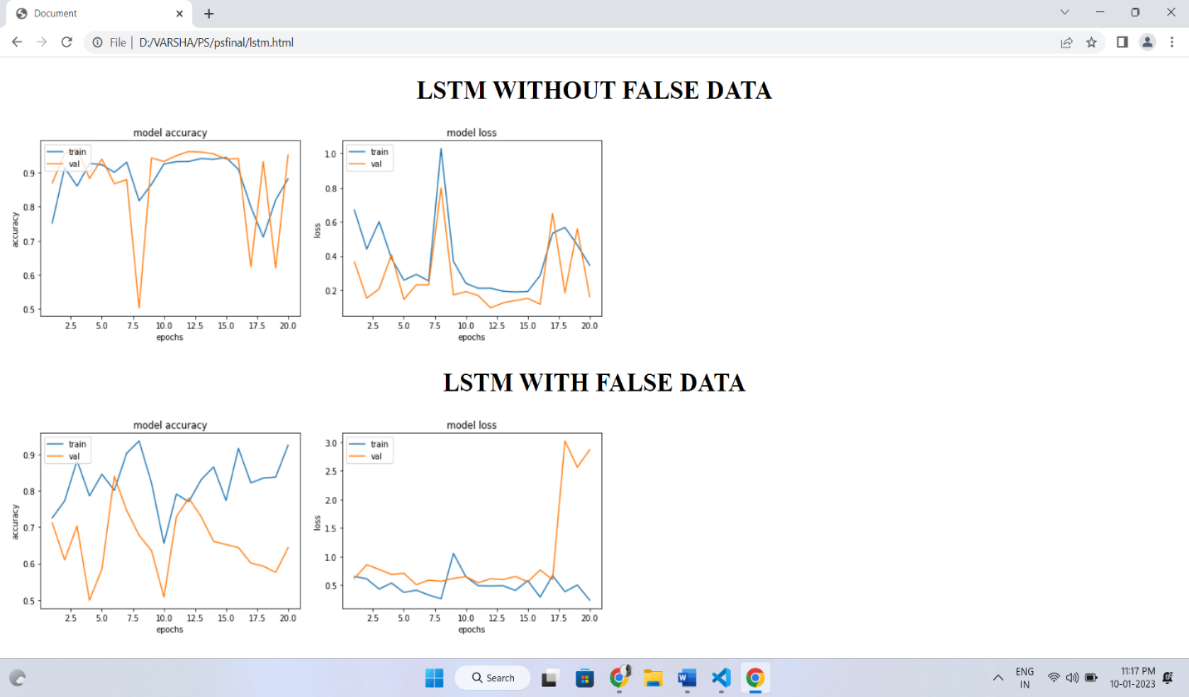


Figure 2: LSTM with and without injection of False Data

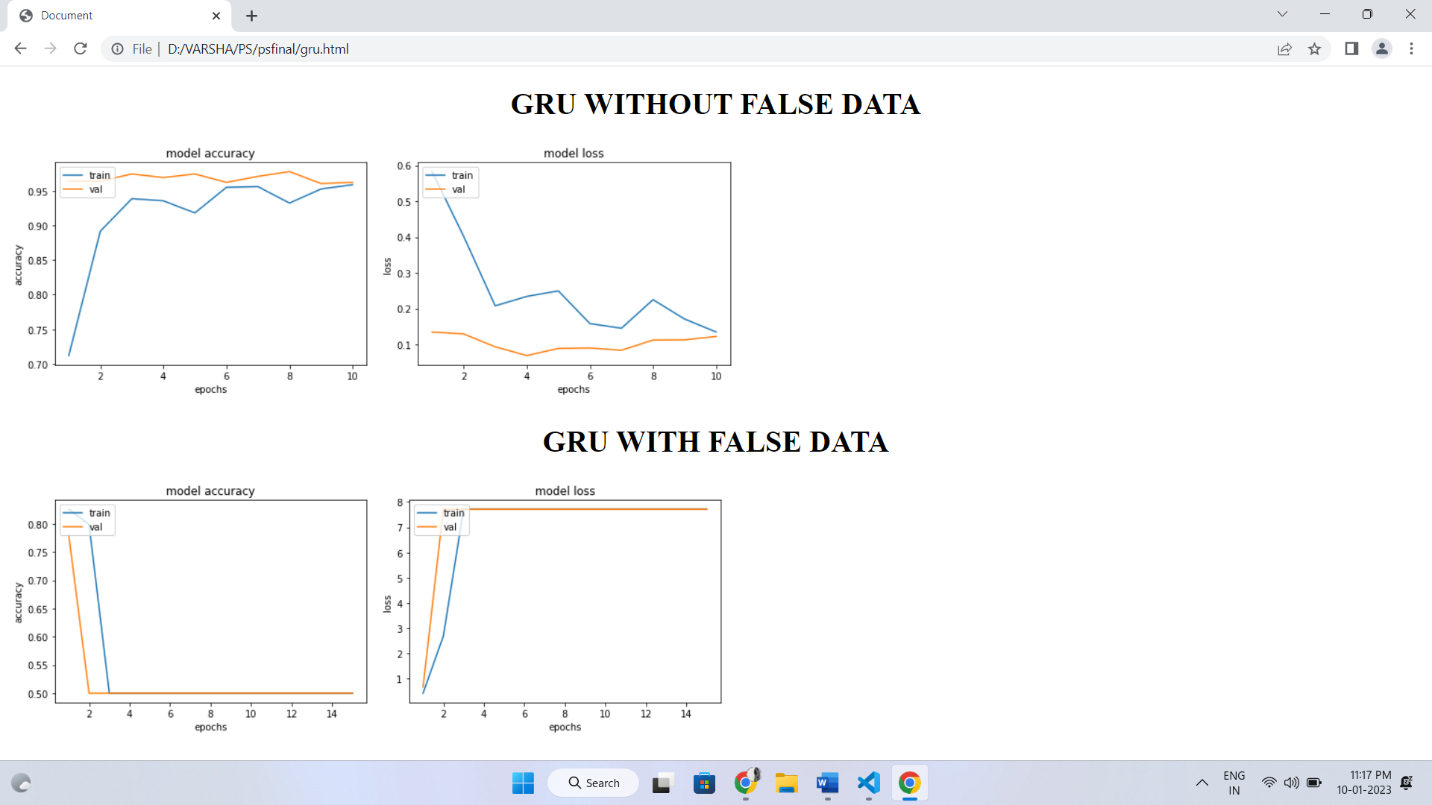
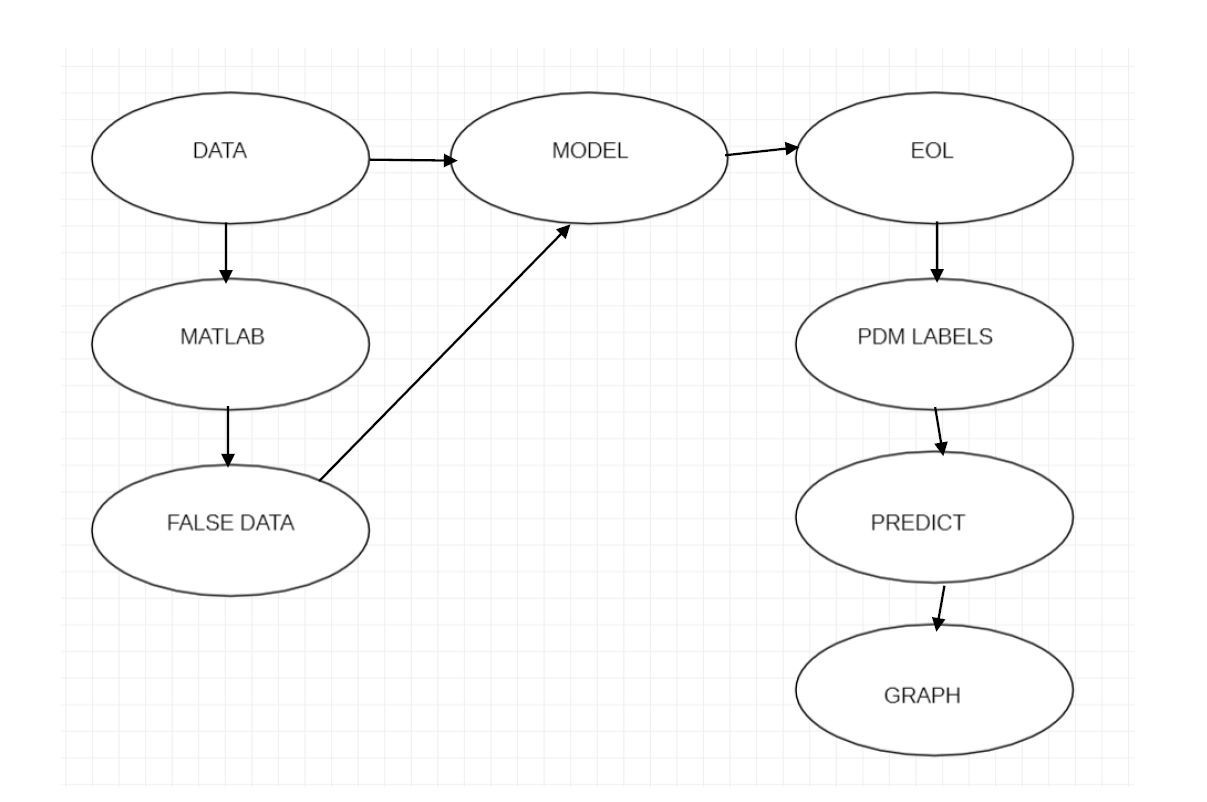


Figure 3: GRU with and without False Data Injection

**SYSTEM WORK FLOW**



**CONCLUSION AND FUTURE SCOPE**

From the above graphs we can see understand that the red dots which are not covered by the green dots are the falsely predicted labels. The Above graphs have the common pattern. In which the True Dataset trained models are having less number of red dots displaying and the false dataset trained models have more number of red dots. This can be understood as the false dataset trained models make a greater number of the false predictions which intern make more number of red dots to appear as the green dots (predicted dots) would not be covering them.

From this application we have understood that the purity of the data is the most important thing to predict the label. This application is not yet completed, there is still room for improvement. For example, we can increase the number algorithms. Also, this application is not real time so we can modify this application to intake data stream directly from the sensors themselves. Also, the number of features used in the current version is less so we can increase the number of features.

**GITHUB LINK:**

https://github.com/AishwaryaPillalamarri/false\_data\_injection\_attacks\_prediction\_using\_DL