*A Project Report on*

# ENERGY THEFT DETECTION USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES

*In partial fulfillment of the requirements for the award of degree of*

**Bachelor of Technology In**

**Electrical and Electronics Engineering**

*Submitted by*

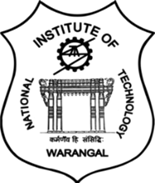
**E. Kruthi Pranada (202218)**

**Raghav Rath (202249)**

*Under the esteemed guidance of* **Dr. Sailaja Kumari M.**

**Professor**

**EEE Department, NIT Warangal**



**Department of Electrical Engineering**

**NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL-506004 (Telangana), May-2024**

**DEPARTMENT OF ELECTRICAL ENGINEERING NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL**



Certificate

This is to certify that the project work titled “**Energy Theft Detection using Machine Learning and Deep Learning Techniques**” is a bonafide record of work carried out by **Ms. E. Kruthi Pranada (202218)** and **Mr. Raghav Rath (202249),** submitted to the faculty of Department of Electrical Engineering in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Electrical and Electronics Engineering at National Institute of Technology, Warangal for the academic year 2023-2024.

Project Guide: Head of the Department:

## Dr. Sailaja Kumari M Dr. B.L. Narasimharaju

Professor Professor

Dept. of Electrical Engineering Dept. of Electrical Engineering National Institute of National Institute of Technology Warangal Technology Warangal

APPROVAL SHEET

The project work titled “**ENERGY THEFT DETECTION USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES**” submitted by **Ms. E. Kruthi Pranada (202218), Mr. Raghav Rath (202249)** are approved for the degree of Bachelor of Technology in Electrical and Electronics Engineering from National Institute of Technology, Warangal.

###### Examiners:

SUPERVISOR:

##### Dr. (Mrs.) Sailaja Kumari M.

Professor, EED, NITW

HEAD OF THE DEPARTMENT:

##### Dr. B.L. Narasimharaju

Professor, EED, NITW

CHAIRMAN:

##### Dr. M. Udaya Bhasker

Associate Professor, EED, NITW

Date:

Place:

# DECLARATION

We declare that this written submission represents our ideas in our own words and where other’s ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

**E. Kruthi Pranada (202218)**

**Raghav Rath (202249)**

# ACKNOWLEDGEMENT

We wholeheartedly thank our guide **Dr. Sailaja Kumari M., Professor,** Department of Electrical Engineering, for being a source of motivation, constant supervision and invaluable encouragement, and for guiding us with perfection in doing work.

We are grateful to **Dr. B.L. Narasimharaju, Professor and Head of Electrical Engineering Department**, for providing us with all the facilities to carry out this project work. We are immensely thankful to him for his moral support during the total span of the project.

We are also very thankful to the project evaluation committee, for their strenuous efforts to evaluate our project.

We consider it a great privilege to express our deep gratitude to many respected personalities who guided, inspired and helped us in the successful completion of the project.

We are also thankful to all our friends who have given valuable suggestions and help in all the stages of the development of the project.

Finally, we would like to dedicate this work to our parents who have provided support and encouragement during every part of life.

**E. Kruthi Pranada (202218)**

**Raghav Rath (202249)**

# ABSTRACT

Electricity is indispensable to human life, driving industrial revolutions and sustaining economies, particularly in developing nations reliant on consistent and affordable power supplies. However, power theft poses a significant threat to electric companies, resulting in catastrophic losses for distributors and financial damage. Manual inspections of customers offer a solution, but they are labor-intensive, costly, and susceptible to corruption.

With advancement of technology, smart meters are becoming more popular as it not only records the electricity consumption, but also communicates the real-time data with the utility companies. This data can be leveraged to gain insights about load patterns which can be used to categorize various consumers and further implement demand response strategies.

The detection of energy theft and identification of fraudulent customers are pivotal for utility companies to maintain operational efficiency and uphold fair billing practices. This project aims to develop a robust Machine Learning or Deep Learning model using Python to tackle these challenges effectively. Through an exploration of various ML techniques and methodologies, including supervised and unsupervised learning methods, the agenda is to provide utility companies with powerful tools for detecting and preventing energy theft, ultimately benefiting both providers and consumers.

Multiple Machine Learning or Deep Learning classifiers such as Random Forest (RF), XGBoost and k-NN (k-Nearest Neighbours), and RNN (Recurrent Neural Networks), CNN (Convolutional Neural Networks), and LSTM (Long Short Term Memory) have been developed to detect energy theft and identify the fraudulent customers. The best results have been obtained with RF model with an accuracy of 85%. However, RF misclassified one specific theft case (5% precision) which XGBoost identifies better (21% precision) with an overall accuracy of 84%.

# CONTENTS

[**CERTIFICATE**](#_gjdgxs) **ii**

**APPROVAL SHEET iii**

[**DECLARATION**](#_gjdgxs) **iv**

[**ACKNOWLEDGEMENT**](#_30j0zll) **v**

**ABSTRACT vi**

[**CONTENTS**](#_1fob9te) v[**ii**](#_1fob9te)

[**LIST OF FIGURES**](#_3znysh7) **ix**

[**LIST OF TABLES** **x**](#_2et92p0)**i**

[**INTRODUCTION** **1**](#_tyjcwt)

* 1. [Problem Statement 1](#_3dy6vkm)
  2. [Objective](#_1t3h5sf) 1
  3. [Scope of Work](#_4d34og8) 1
  4. [Outline of Chapters 2](#_2s8eyo1)

[**LITERATURE REVIEW** **3**](#_17dp8vu)

* 1. Smart Meter Data Analytics3
  2. Energy Theft Detection [6](#_lnxbz9)

[**MACHINE LEARNING AND DEEP LEARNING MODELS**](#_35nkun2) **8**

* 1. [Machine Learning Algorithms](#_1ksv4uv) 8
  2. Regression9
  3. Classification [10](#_z337ya)
  4. Machine Learning Classifiers [12](#_3j2qqm3)
     1. [Random Forest](#_1ksv4uv) 12
     2. [XGBoost](#_1y810tw) 13
     3. [k-NN](#_4i7ojhp) [1](#_4k668n3)5
  5. [Deep Learning Classifiers 16](#_2xcytpi)
     1. [RNN](#_1ci93xb) 16
     2. [CNN](#_3whwml4) 20
     3. [LSTM](#_3as4poj) 21

[**METHODOLOGY**](#_1pxezwc) **24**

* 1. Dataset24
  2. Data Statistics/Summary25
  3. Theft Generator Algorithm27
  4. [Data Analysis](#_3o7alnk)28

**RESULTS** **29**

* 1. [Long Short Term Memory](#_ihv636) 29
     1. [Confusion Matrix](#_32hioqz) 29
     2. [Classification Report](#_32hioqz) 30
  2. [Convolutional Neural Network 3](#_41mghml)0
     1. [Confusion Matrix 3](#_2grqrue)0
     2. [Classification Report 3](#_vx1227)1
  3. [Recurrent Neural Network 3](#_3fwokq0)1
     1. [Confusion Matrix](#_1v1yuxt) 31
     2. [Classification Report](#_1v1yuxt) 32
  4. [kNN](#_4i7ojhp) 32
     1. [Confusion Matrix](#_19c6y18) 32
     2. [Classification Report](#_19c6y18) 33
  5. [XGB](#_3tbugp1)oost 33
     1. [Confusion Matrix](#_28h4qwu) 33
     2. [Classification Report](#_28h4qwu) 34
  6. [Random Forest](#_nmf14n) 34
     1. [Confusion Matrix](#_37m2jsg) 34
     2. [Classification Report](#_37m2jsg) 35
  7. [Comparison of Models](#_1mrcu09) 35

[**CONCLUSION & FUTURE SCOPE** 36](#_46r0co2)

* 1. [CONCLUSION 36](#_2lwamvv)
  2. [FUTURESCOPE 37](#_111kx3o)

[**REFERENCES** 3](#_3l18frh)8

[APPENDIX 43](#_206ipza)

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIGURE NUMBER | NAME OF THE FIGURE | PAGE NUMBER |
| 2.1 | Difference between normal and smart meter | 4 |
| 2.2 | Smart meter data analytics | 5 |
| 3.1 | Standardized steps in machine learning | 9 |
| 3.2 | Linear regression | 10 |
| 3.3 | Difference between binary and multiclass classification | 11 |
| 3.4 | An example of classification | 11 |
| 3.5 | Random forest classifier | 13 |
| 3.6 | Basic architecture of XGBOOST | 14 |
| 3.7 | Basic understanding of RNN | 18 |
| 3.8 | Basic architecture of RNN | 19 |
| 3.9 | Basic architecture of CNN network | 21 |
| 3.10 | Basic architecture of LSTM | 23 |
| 4.1 | Data set | 24 |
| 4.2 | Code for consumers | 25 |
| 4.3 | Code for features and thefts | 26 |
| 4.4 | Percentage of normal and theft entries in the data | 28 |
| 4.5 | Percentage of each theft in data set | 28 |
| 5.1 | Confusion matrix for CNN | 29 |
| 5.2 | Prediction on data set | 30 |
| 5.3 | Confusion matrix for RNN | 30 |
| 5.4 | Prediction on data set | 31 |
| 5.5 | Confusion matrix for KNN | 31 |
| 5.6 | Prediction on data set | 32 |
| 5.7 | Confusion matrix for XGBOOST | 32 |
| 5.8 | Prediction on test data set | 33 |
| 5.9 | Confusion matrix for Random forest | 33 |
| 5.10 | Prediction on test data set | 34 |
| 5.11 | Confusion matrix for LSTM | 34 |
| 5.12 | Prediction on test data set | 35 |

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| TABLE NUMBER | NAME OF THE TABLE | PAGE NUMBER |
| 3.1 | An overview of pros and cons of KNN | 16 |
| 5.1 | An overview of precision and accuracy of all models | 35 |

**CHAPTER 1**

# INTRODUCTION

## Problem Statement

Build a suitable Machine Learning or Deep Learning Model to detect energy theft and identify the fraudulent customers using Python coding.

## Objective

The main objective of the project work is to develop an effective and accurate Machine Learning or Deep Learning model using Python, which can be utilized by utility companies to detect energy theft and identify fraudulent customers. The model will be capable of analyzing consumption patterns, identifying irregularities, and distinguishing between normal usage and potential fraudulent activities. This will enable companies to mitigate losses, enhance operational efficiency, and ensure fair billing, benefitting both providers and consumers alike.

## Scope of Work

* + - The project presents a model to detect energy theft and identify fraudulent customers based on the load data and the available energy resources of the consumer. energy resources of the consumer.
    - The residential load dataset comprises 17 different consumer types.
    - The model is formulated for these loads incorporating these consumers and their electricity consumption through various electrical equipment and appliances.
    - The optimized solution for the model is found out using the Particle swarm optimization algorithm for the Time of Use tariff rates.
    - The model is simulated with data and results for different cases are analyzed.

## Outline of Chapters

Chapter 2: This chapter is the literature review that briefly describes various smart meter data analytics usage, energy theft in recent times, and techniques to detect energy theft.

Chapter 3: Describes the architecture and working of the various ML or DL models used.

Chapter 4: Explains the data considered and algorithm

Chapter 5: Results of the different ML and DL models are presented in this chapter.

Chapter 6: This chapter presents the conclusion. It also contains steps in the direction of modification and improvement of current work.

# CHAPTER 2

# LITERATURE REVIEW

## Smart Data Analytics

A smart meter is a digital device that replaces traditional analog meters to record energy consumption in homes. It communicates with the utility company via the internet, providing real-time data on energy usage. Smart meters enable consumers to monitor and manage their electricity consumption more effectively, similar to a prepaid mobile connection. They help reduce billing errors, allow power companies to analyze power quality in real-time, reduce peak power purchase costs, and aid in recovering receivables. Smart meters also contribute to environmental sustainability by promoting energy efficiency. Despite concerns about tampering and cyber attacks, smart meters offer benefits such as remote monitoring, outage detection, and improved billing accuracy, making them a valuable tool for efficient energy management in daily life [1].

In the realm of smart meter data analytics, energy theft detection, load profiling, bad data detection, and the importance of addressing data security and privacy concerns are key areas of focus. “ENERGY THEFT DETECTION” is particularly critical, with supervised methods involving feature extraction and classification to identify abnormal energy consumption patterns, and unsupervised methods like clustering algorithms detecting anomalies without labeled data [1].



Fig 2.1 Difference between normal and smart energy meter

Load profiling, which classifies consumers based on their electricity consumption behaviors, is also explored, along with the significance of real-time detection for applications like short-term load forecasting. The importance of spatiotemporal correlation in identifying outliers and the need for robust methods to handle cyber attacks or bad data without prior detection are underscored [1].

Analyzing electricity consumption patterns and load profiles recorded by smart meters involves several key steps, with load analysis being a crucial aspect. This process aims at improving energy efficiency, grid reliability, and customer satisfaction while enabling utilities to optimize their operations and resources, thereby empowering both utilities and consumers to make informed decisions about energy usage and management. Anomaly detection, with a focus on “BAD DATA DETECTION” and “NON TECHNICAL LOSS DETECTION” is central to load analysis. Bad data refers to missing data or unusual patterns caused by unplanned events or failures in data collection, communication, or entry [1].

## 

Fig 2.2 Smart Meter Data Analytics [1]

Various modeling methods are employed to address these challenges, including time-series-based approaches, low-rank matrix techniques, and machine learning algorithms like support vector machines (SVM) [1]. Energy theft, a significant concern for utilities, stems from issues such as meter tampering, illegal connections, meter malfunctions, billing irregularities, and unpaid bills. Detecting energy theft using only smart meter data is a multifaceted task, categorized into supervised and unsupervised learning methods [2]. By employing sophisticated analytics and detection methods, utilities can mitigate losses, enhance operational efficiency, and ensure fair billing practices, ultimately benefiting both providers and consumers alike. This overview thus provides a holistic perspective of the current state of smart meter data analytics.

## Energy Theft Detection

Smart meters play a crucial role in smart grid systems, providing vital energy consumption data essential for detecting theft. However, their vulnerability to tampering and cyber threats poses significant challenges, leading to non-technical losses (NTLs) through electricity theft. Addressing this issue is paramount for smart grid operators, prompting extensive research into various machine learning (ML) techniques such as k-nearest neighbors, decision trees, random forest, bagging ensemble, and artificial neural networks. Among these, the random forest model stands out, boasting an accuracy exceeding 94% across diverse scenarios in theft detection [2].

Moreover, data-driven approaches to NTL detection have been explored, including methods to balance imbalanced datasets and model realistic theft behaviors. The choice of ML algorithms and feature engineering significantly impacts detection accuracy [6]. Although integrating hardware-based and software-based methods enhances theft detection capabilities, it comes with heightened deployment and maintenance costs. Nonetheless, detecting energy theft in smart grids remains pivotal for ensuring system integrity and efficiency. Various ML approaches have been utilized, including gradient boosting classifiers (GBCs) and feature engineering-based preprocessing techniques [3].

An innovative approach, the gradient boosting theft detector (GBTD), leverages cutting-edge GBCs such as extreme gradient boosting (XGBoost), categorical boosting (CatBoost), and light gradient boosting method (LightGBM) [6]. Unlike traditional ML algorithms, the GBTD approach prioritizes feature engineering-based preprocessing to boost detection performance while reducing time-complexity. Stochastic features like standard deviation, mean, minimum, and maximum value of daily electricity usage substantially enhance both detection rate and false positive rate (FPR) of the GBCs [3].

Furthermore, ongoing advancements in smart meter technology, coupled with evolving cyber defense mechanisms, contribute to a dynamic landscape in theft detection and prevention. Collaborative efforts among stakeholders, including utilities, regulatory bodies, and technology providers, are essential for developing robust strategies to combat electricity theft effectively. By leveraging comprehensive data analytics and innovative ML techniques, smart grid operators can stay ahead of emerging threats, ensuring the reliability and sustainability of energy distribution systems for years to come [4].

Concurrently, strategies for data preprocessing like interpolation, the three sigma rule, and normalization are applied to proficiently handle missing values and outliers. Additionally, an Adasyn algorithm is employed to address class imbalance, thereby ensuring equitable treatment of minority class samples by the model. Feature extraction is performed utilizing a Visual Geometry Group (VGG-16) module to identify irregular consumption patterns, succeeded by classification employing a Firefly Algorithm-based Extreme Gradient Boosting (FA-XGBoost) method. This holistic methodology produces noteworthy outcomes, characterized by elevated F1-scores, precision, and recall rates, outperforming established approaches [4].

Moreover, the study critically reviews prior methodologies for electricity theft detection, categorizing them into state-based solutions, game theory, and machine learning approaches. Notably, machine learning techniques, including unsupervised clustering and supervised classification, are explored for their application to unlabelled datasets to differentiate between fraudulent and legitimate consumers [5]. This study represents a significant advancement in the field of electricity theft detection, offering utility companies an effective tool for identifying and curbing energy theft, thereby promoting energy conservation and financial savings [3].

# CHAPTER 3

# MACHINE LEARNING AND DEEP LEARNING MODELS

## Machine learning algorithms

1. Supervised Learning Algorithms: Supervised learning involves training the machine using labeled datasets where the input data is mapped to the output data.
2. Unsupervised Learning Algorithms: Unsupervised learning does not require labeled data for training; the machine learns from unlabeled datasets.
3. Reinforcement Learning Algorithms: Reinforcement learning involves an agent interacting with an environment, learning from feedback in the form of rewards or punishments.

Supervised learning algorithms can be further divided into two main categories:

* Classification algorithms: Used for predicting categorical output variables.
* Regression algorithms: Used for predicting continuous output variables.

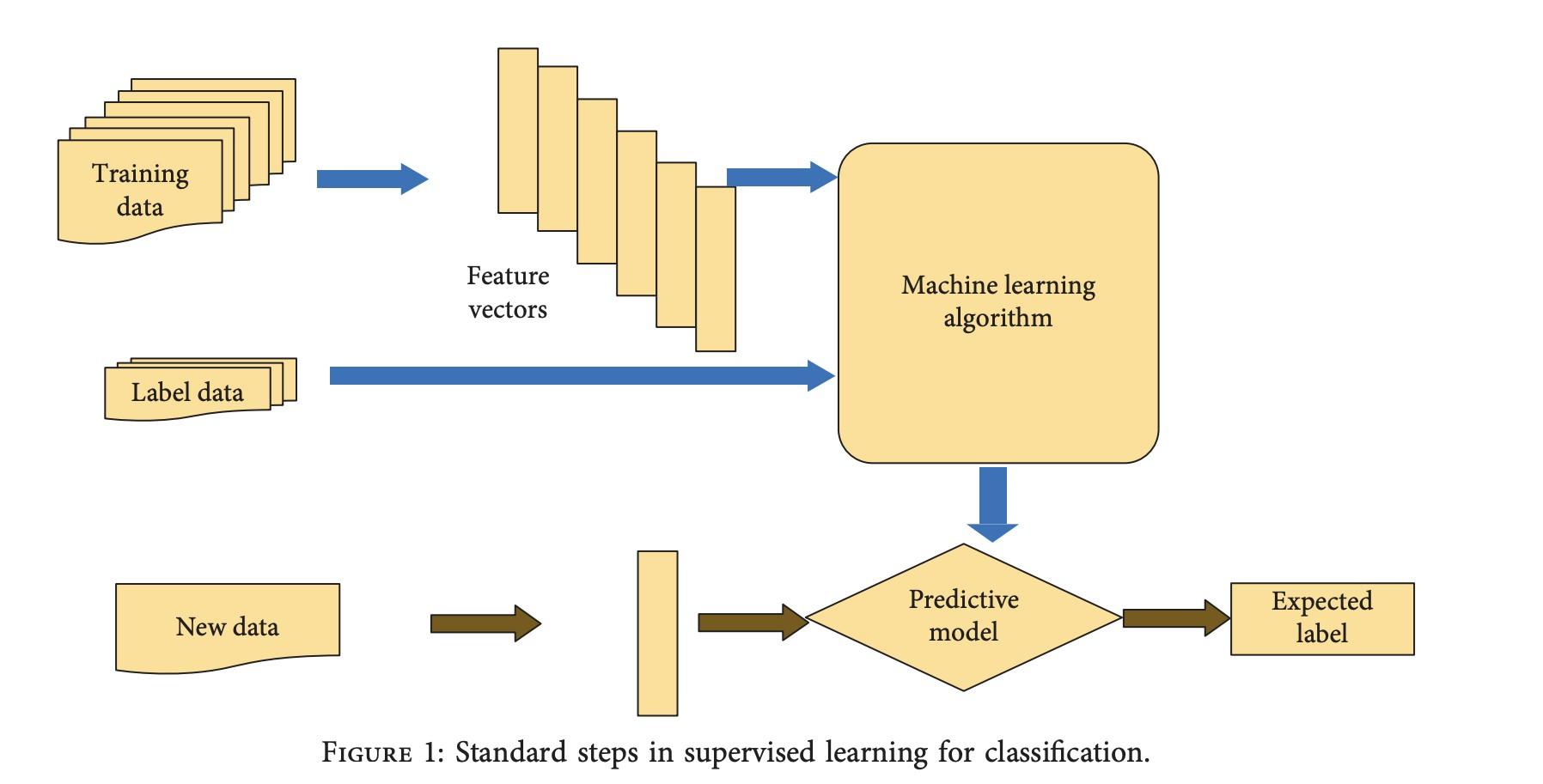


FIG-3.1 Standard steps in supervised learning for classification

## Regression

Regression is a foundational method in machine learning that aims to model the connections between independent and dependent variables, ultimately focusing on forecasting results. It involves training algorithms to identify patterns that characterize the distribution of data points, enabling accurate predictions for new data points or input values. Regression encompasses various types, with linear regression focusing on fitting data points along a clear line, while logistic regression categorizes data points above or below a line for distinct classifications like fraud detection or spam filtering. Regression is crucial for predictive analytics, offering insights into key business issues such as customer churn and price elasticity. Linear regression is the simplest model, while nonlinear regression, including logistic regression and neural networks, provides more flexibility but may sacrifice explainability.

# 

FIG-3.2 LINEAR REGRESSION

## Classification

Classification in machine learning is a fundamental task that involves categorizing data into predefined classes or categories based on their features. It is a supervised learning technique where an algorithm is trained on a labeled dataset to predict the class or category of new, unseen data. The main aim of classification machine learning is to create a model that can accurately give a label or category to a new observation based on its features. There are two main types of classification in machine learning:

1. **Binary classification:** In binary classification, the main task is to categorize the input into two classes or categories. For example, determining whether a person has a certain disease based on their health conditions.
2. **Multiclass classification:** In multiclass classification, the task is to classify the input into one of many classes or categories. For example, identifying different species of flowers based on their characteristics.

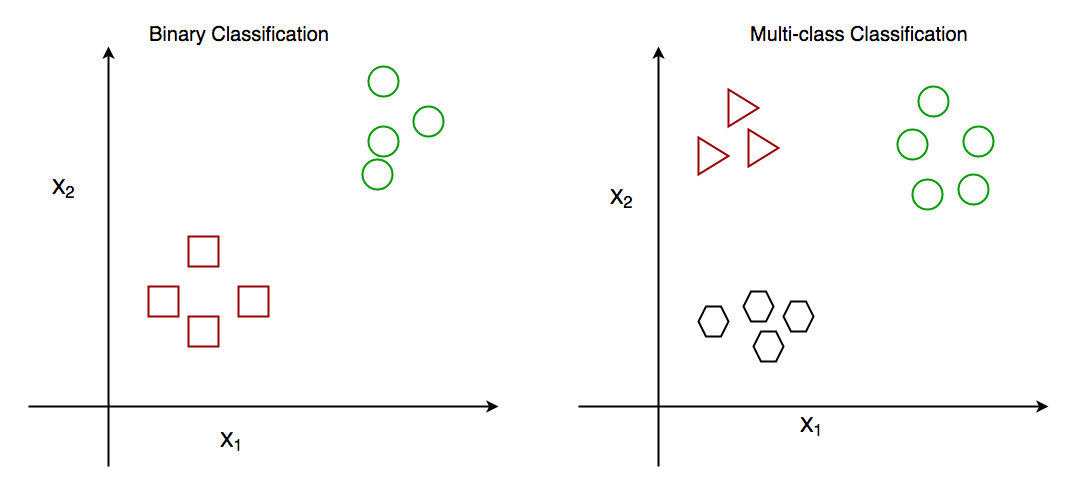


FIG-3.3 Difference between Binary classification and Multi-class Classification

Classification algorithms play an important role in various applications, such like spam detection in emails, image recognition, sentiment analysis, and medical diagnosis. By leveraging labeled data, these algorithms learn patterns and relationships within the data to make accurate predictions about the class or category of new observations.

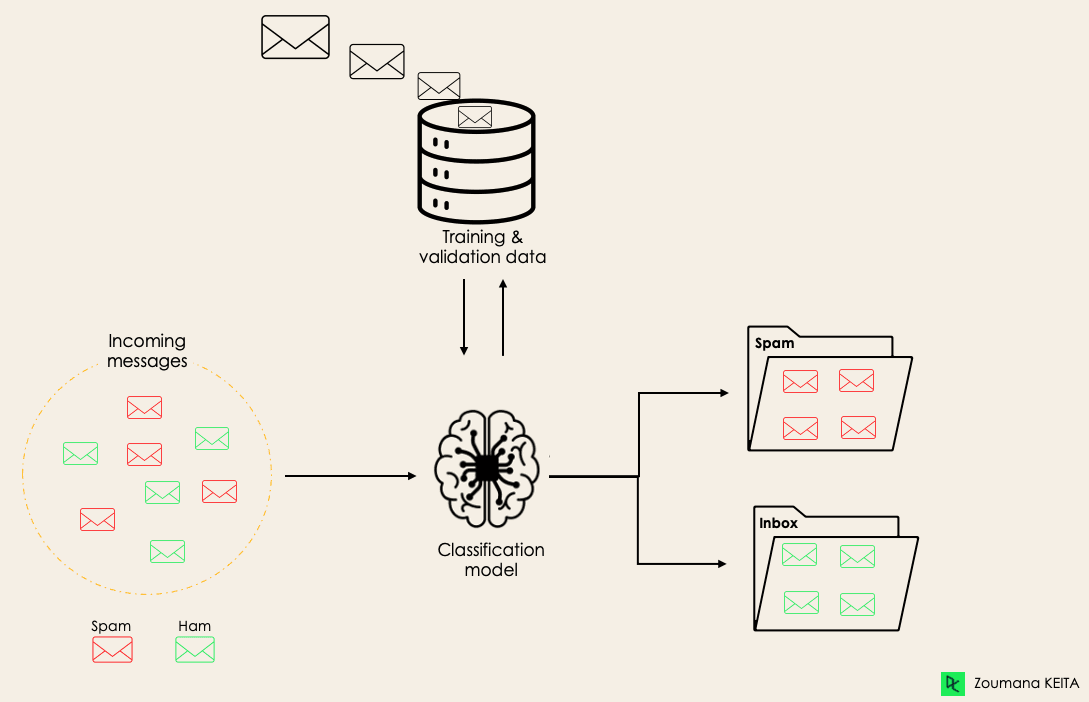


FIG- 3.4 An example of classification

## Machine Learning Classifiers

* + 1. **Random Forest**

A Random Forest Classifier is an ensemble learning technique that mixes multiple decision trees to increase the accuracy and stability of the predictions. Random Forest Classifier is a powerful and widely-used machine learning algorithm that uses the strengths of ensemble learning and decision trees to make reliable predictions, even in the presence of complex and noisy datasets . General architecture of a Random Forest Classifier involves:

* Ensemble of Decision Trees: The Random Forest algorithm builds an ensemble of decision trees, where every tree is trained on a random subset of the training data.
* Random Feature Selection: During the training of every decision tree, the algorithm chooses a random subset of features to consider for splitting the nodes. This introduces diversity among the trees, as each one focuses on different aspects of the data.
* Robustness and Reduced Overfitting: The ensemble nature of the Random Forest Classifier, combined with the random feature selection and bagging, helps to decrease overfitting and improve the overall robustness and accuracy of the model. The diversity among the decision trees ensures that the model is less sensitive to the peculiarities of the training data.
* Bootstrap Aggregating (Bagging): The Random Forest algorithm uses the technique of bootstrap aggregating (bagging) to create the diverse subsets of training data for each decision tree. It randomly samples the training data with replacement, allowing some instances to be repeated while others are left out.
* Voting/Averaging: For classification tasks, the Random Forest Classifier makes predictions by taking a vote among the predictions of the individual decision trees. For regression tasks, it averages predictions of trees.

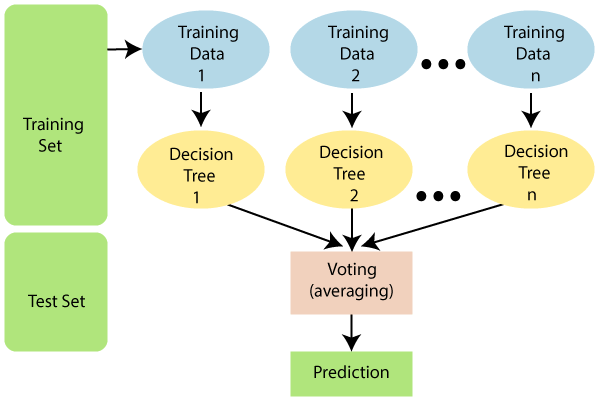


FIG- 3.5 Random forest classifier

## XGBoost (Extreme Gradient Boosting)

* XGBoost is a machine learning algorithm that constructs an ensemble of decision trees. Each subsequent tree is trained to correct errors made by the previous trees in the ensemble.
* XGBoost has gained widespread popularity for its ability to outperform other algorithms in terms of both speed and accuracy.
* XGBoost is an optimized and scalable implementation of the gradient boosting algorithm, a powerful ensemble learning technique.

**Basic architecture of XGBoost:**

* **Ensemble of Decision Trees:** XGBoost builds a sequence of decision trees, where each new tree corrects errors from previous trees.
* **Gradient Boosting:** XGBoost uses the gradient boosting algorithm to iteratively add new models to the ensemble, minimizing residual errors.
* **Regularization:** XGBoost incorporates techniques like L1 and L2 regularization to prevent overfitting and improve generalization.
* **Sparse Data Handling:** XGBoost uses a specialized split-finding algorithm to effectively handle different types of sparse data.
* **Out-of-Core Computation:** XGBoost can handle large datasets that don't fit in memory by optimizing disk usage and minimizing disk I/O.
* **Parallelization:** XGBoost is designed for parallel training, using a level-wise strategy to evaluate split quality efficiently.
* **Feature Importance:** XGBoost provides insights into the relative importance of features, aiding in feature selection.
* **Scalability:** XGBoost is scalable and can be used on a single machine or in distributed processing frameworks.

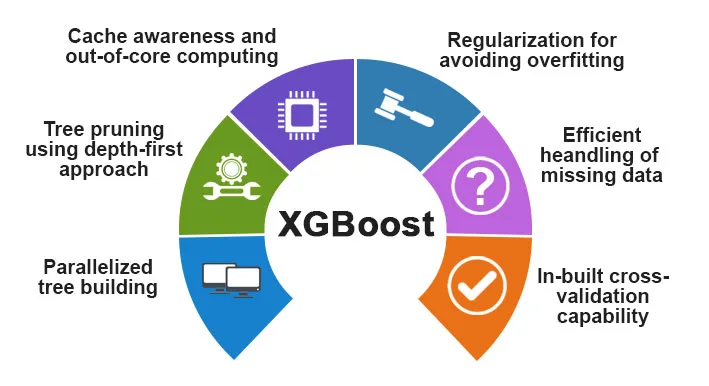
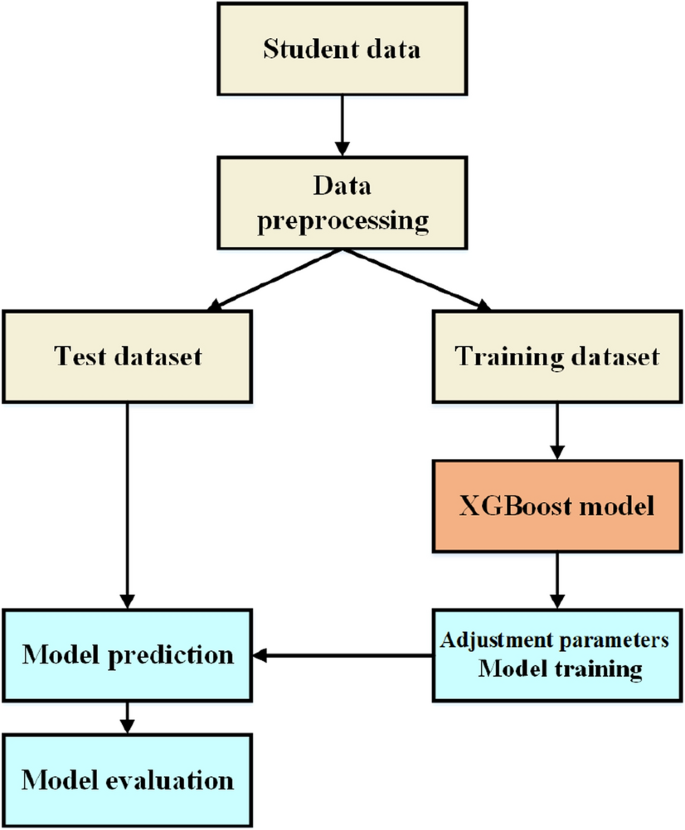
****

FIG-3.6 Basic architecture of XGBOOST

## K-NN (k-Nearest Neighbours)

K-Nearest Neighbors (KNN) is a straightforward yet highly efficient supervised machine learning algorithm utilized for classification and regression purposes. It functions based on the concept of similarity, where it predicts the label or value of a new data point by considering the labels or values of its closest neighbors. How KNN works:

* Nearest Neighbor Identification: KNN finds the K nearest neighbors to a given data point based on a distance metric, typically using Euclidean distance.
* Classification: For classification tasks, KNN determines the class of the data point based on the majority vote of its K nearest neighbors. The class with the highest frequency among the neighbors is assigned to the data point.
* Regression: In regression tasks, KNN predicts the value of the data point based on the average (or median) value of its K nearest neighbors.
* Parameter Selection: Choosing the right value of K is crucial for KNN. Typically, an odd value of K is selected to avoid ties in classification. The optimal K value can be determined through parameter tuning and cross-validation methods.

Limitations of K-Nearest Neighbors (KNN) Algorithm:

* Slow Speed: The algorithm's performance can be slow, particularly when dealing with a large number of training records and high-dimensional data.
* Memory and Storage Issues: KNN may face challenges with memory and storage for large datasets, impacting its efficiency and scalability.
* Sensitivity to 'K' Selection: The selection of the optimal number of neighbors (K) in the KNN algorithm is crucial, as it can substantially influence the model's performance. Choosing a small value for K may cause the model to overfit the training data, while selecting a large value for K can lead to underfitting, where the model fails to capture the underlying patterns in the data accurately.
* ​​Imbalanced Classes and Missing Values: KNN can struggle with imbalanced classes, where some classes have significantly more instances than others, leading to bias. Additionally, missing values in the data can affect the distance calculation and classification reliability.
* Lack of Explanation: KNN does not provide explanations or confidence levels for its predictions, limiting its interpretability and utility in certain applications.
* Outlier Sensitivity: KNN can be sensitive to outliers, potentially impacting the accuracy of predictions, especially when dealing with noisy data.

| Pros of KNN Algorithm | Cons of KNN Algorithm |
| --- | --- |
| Known for simplicity, comprehensibility, and scalability | Expensive in determining K if the dataset is large |
| High predictive power leading to effectiveness and efficiency | Unclear parameter selection for K and choice of distance metric |
| Effective for large training sets | Sensitivity to data scale and irrelevant features |
| Versatile for both classification and regression tasks | High computation cost due to distance calculations for each training example |
| Suitable for nonlinear data with no assumptions about the data | Dependency on accurate distance computations for algorithm accuracy |
| Less calculation time | Prediction phase is slow for larger datasets |

TABLE -3.1 An overview of Pros and Cons of KNN

## Deep Learning Classifiers

## RNN (Recurrent Neural Network)

The Recurrent Neural Network (RNN) is a variant of artificial neural networks tailored for handling sequential data processing and generation. In contrast to conventional feed-forward neural networks, RNNs possess internal memory, facilitating the processing of input sequences. At each time step, the output is contingent on both the present input and the preceding hidden state. This attribute empowers RNNs to grasp temporal relationships within the data, rendering them effective for applications involving sequential data analysis, such as natural language processing, speech recognition, and time series forecasting.

* RNNs incorporate a recurrent link that facilitates the transmission of information from one time step to the subsequent, granting them the capability to retain past inputs.
* They employ a consistent set of weights across all time steps, thereby minimizing the parameter count and enhancing the efficiency of RNNs.
* With the ability to process input and output sequences of varying lengths, RNNs exhibit adaptability, rendering them suitable for a diverse array of applications.

**How RNNs work:**

* RNNs sequentially handle inputs, where the output at each step relies on the current input and the preceding hidden state.
* This functionality enables RNNs to grasp temporal relationships within the data, positioning them favorably for tasks concerning sequential data analysis like natural language processing, speech recognition, and time series forecasting.

**Applications of RNNs:**

* RNNs are widely used in natural language processing tasks, such as language modeling, machine translation, and text generation.
* They are also used in speech recognition, handwriting recognition, and time series forecasting.
* RNNs can be combined with other neural network architectures, such as convolutional neural networks (CNNs), to tackle complex problems.

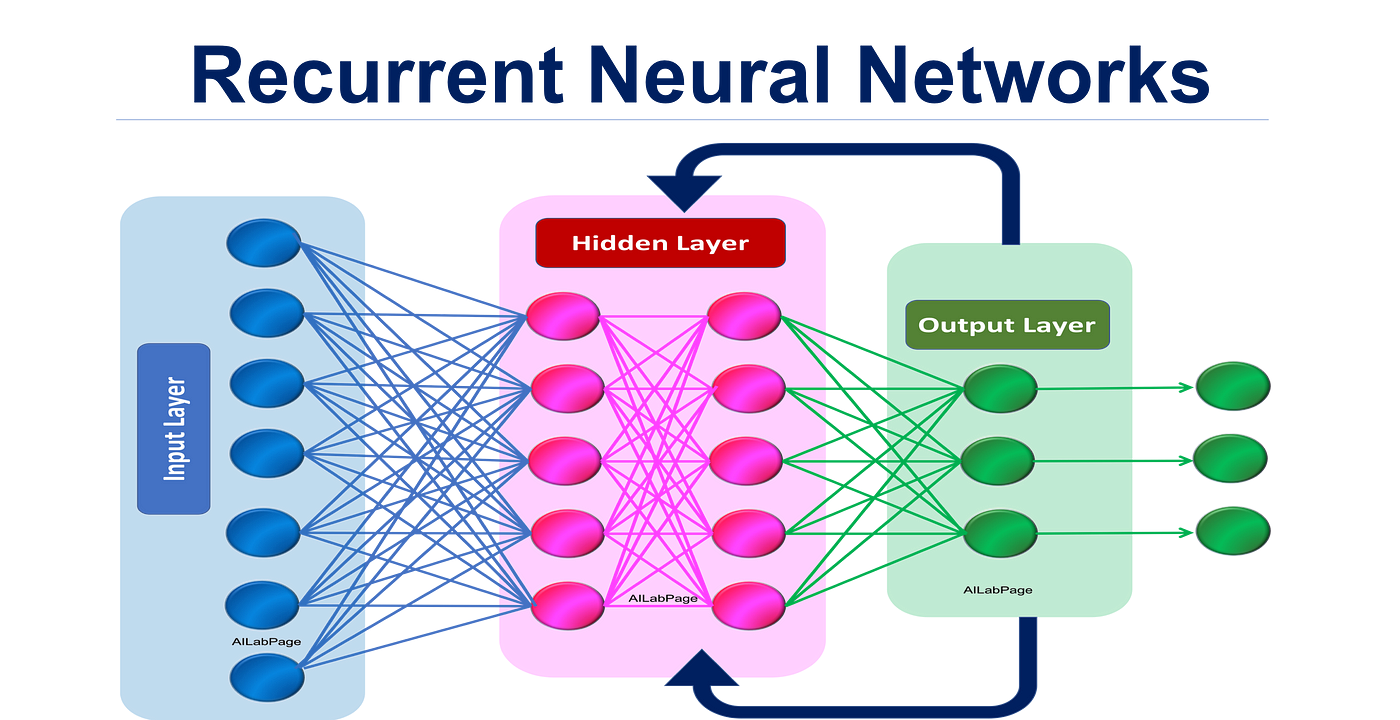
****

FIG 3.7 -Basic understanding of RNN network

Recurrent Neural Network (RNN) Basic architecture of a Recurrent Neural Network (RNN) involves the following key components and processes based on the provided sources:

1. Fundamental Feature: An RNN contains at least one feedback connection, enabling activations to flow in a loop. This loop structure allows RNNs to perform temporal processing, learn sequences, and handle tasks like sequence recognition, reproduction, association, and prediction.
2. Network Structure: RNN architectures can vary, with some consisting of a standard Multi-Layer Perceptron (MLP) with added loops to exploit non-linear mapping capabilities and incorporate memory. Others may have more uniform structures with every neuron connected to all others, potentially using stochastic activation functions.
3. Sequence Processing: RNNs excel in tasks where present input data depends on previous inputs. They aim to find the relationship between the current input and previous inputs, theoretically capable of utilizing information sequences of any length but often limited to looking back only a few steps.
4. Unfolding RNN: Unfolding an RNN involves repeating the same layer structure for the complete sequence, with each time step represented by an input (Xt), a hidden state (A), and a memory calculation based on the previous hidden state and current input.
5. Memory and Temporal Dependencies: RNNs are valued for their ability to connect previous information to the present task, allowing them to handle long-term dependencies and leverage past information for current predictions.

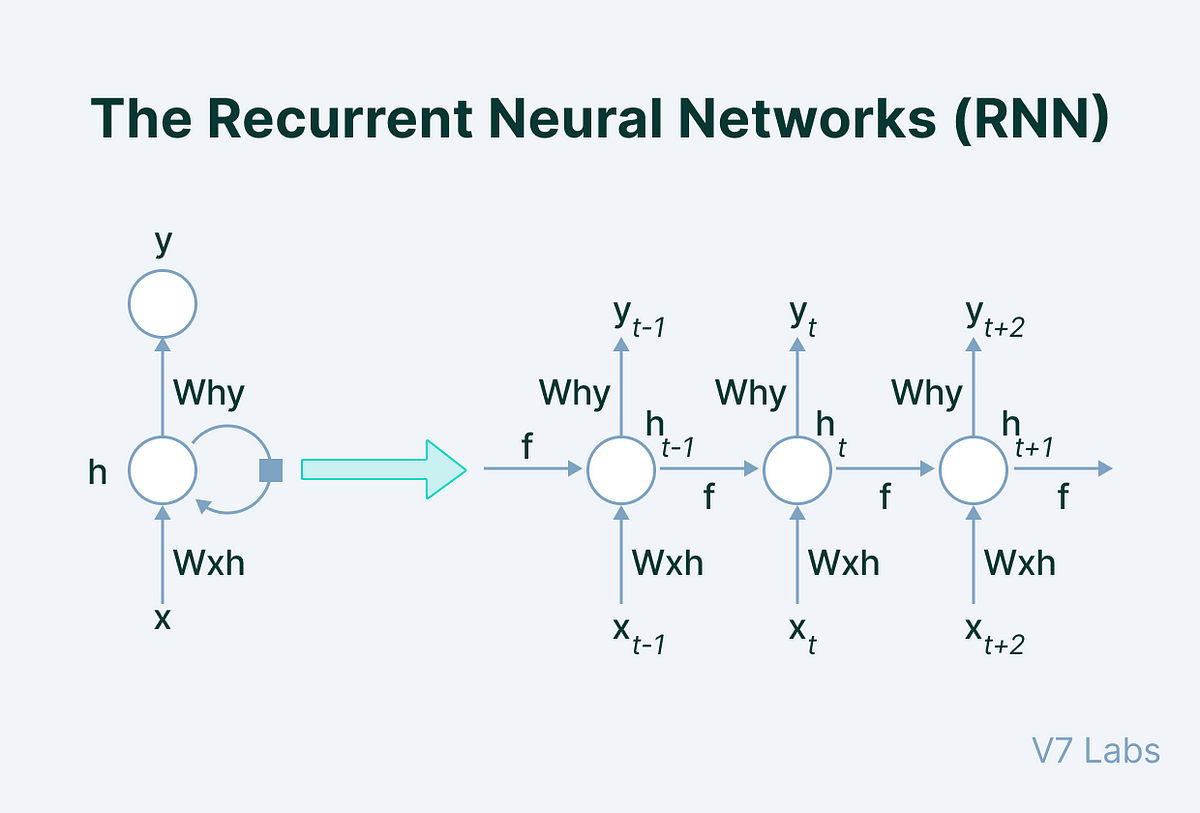


FIG 3.8 -Basic architecture of RNN network

## CNN (Convolutional Neural Network)

A Convolutional Neural Network (CNN or ConvNet) represents a specialized deep learning neural network structure adept at handling and scrutinizing visual data, including images and videos. Below are the essential aspects regarding CNNs:

* The fundamental elements of a CNN encompass the convolutional layers, tasked with implementing filters on the input image to extract features such as edges, textures, and shapes. Subsequently, these features undergo processing through pooling layers to diminish dimensionality, culminating in the final prediction or classification via fully connected layers.
* CNNs excel in tasks such as image classification, object detection, image segmentation, and image generation. They possess the capability to autonomously acquire hierarchical feature representations from the data, rendering them ideal for scenarios where spatial relationships and patterns hold significant importance.
* CNNs find widespread applications in medical image analysis, self-driving cars, facial recognition, and visual search in e-commerce. They have emerged as a potent asset within the realm of computer vision.

The basic architecture of a Convolutional Neural Network (CNN) includes:

* Convolutional Layer: It applies filters to extract features from the input image and involves sliding filters across the image and computing dot products.
* Activation Layer: Introduces non-linearity using functions like ReLU.
* Pooling Layer: It reduces spatial size of feature maps, performs common operations like max pooling and average pooling.
* Fully Connected Layer: Produces final classification or prediction and learns non-linear combinations of high-level features.

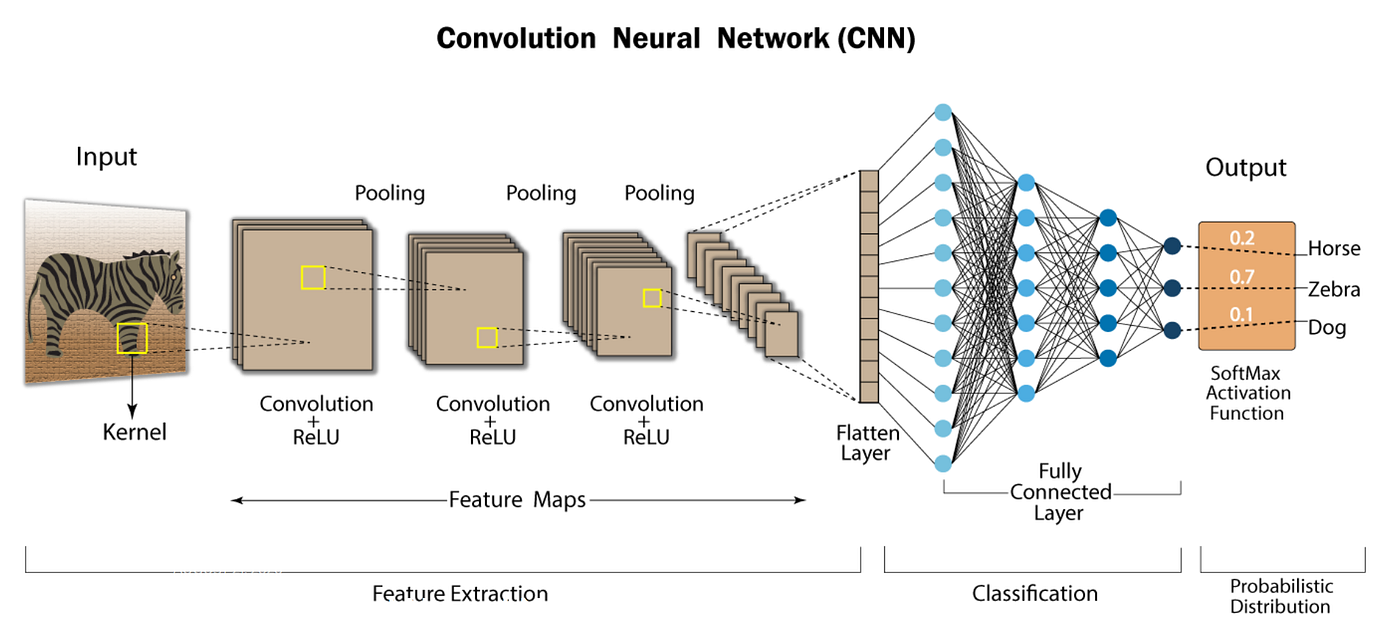


FIG- 3.9 Basic Architecture of CNN network

## 

## 

## LSTM (Long Short Term Memory)

Long Short-Term Memory (LSTM) represents a subtype of Recurrent Neural Network (RNN) specially adept at capturing extended dependencies within sequential data. LSTMs are engineered to surmount the challenge of the vanishing gradient, a hindrance frequently encountered by conventional RNNs, thereby enabling them to process entire sequences while preserving valuable insights from prior data points to facilitate the analysis of fresh data. Employing an array of gates—including input, output, and forget gates—LSTMs regulate the information flow within the network, allowing for the selective retention or dismissal of data as necessary. Renowned for their versatility, LSTM networks find application across diverse domains such as natural language processing, speech recognition, time series forecasting, video analysis, and beyond including:

1. Forecast: LSTMs are useful for predicting values ​​or outcomes based on past observations, such as stock prices, weather, or traffic. They can model and forecast time series data, use nonlinear and nonstationary data, and predict stock prices or weather conditions.
2. Speech Recognition: LSTMs can model the temporal and acoustic features of speech signals, dealing with noisy and incomplete inputs. They can recognize spoken words, synthesize speech from text, and improve speech recognition by combining with other neural network layers.
3. Video Analysis: LSTMs can capture the spatial and temporal features of video frames, learning from complex and dynamic scenes. They can classify activities or emotions in a video, generate captions or summaries for a video, and analyze video data.
4. Handwriting Recognition: LSTMs can be applied to handwriting recognition tasks, where they can learn to recognize handwritten characters or words from sequential data.

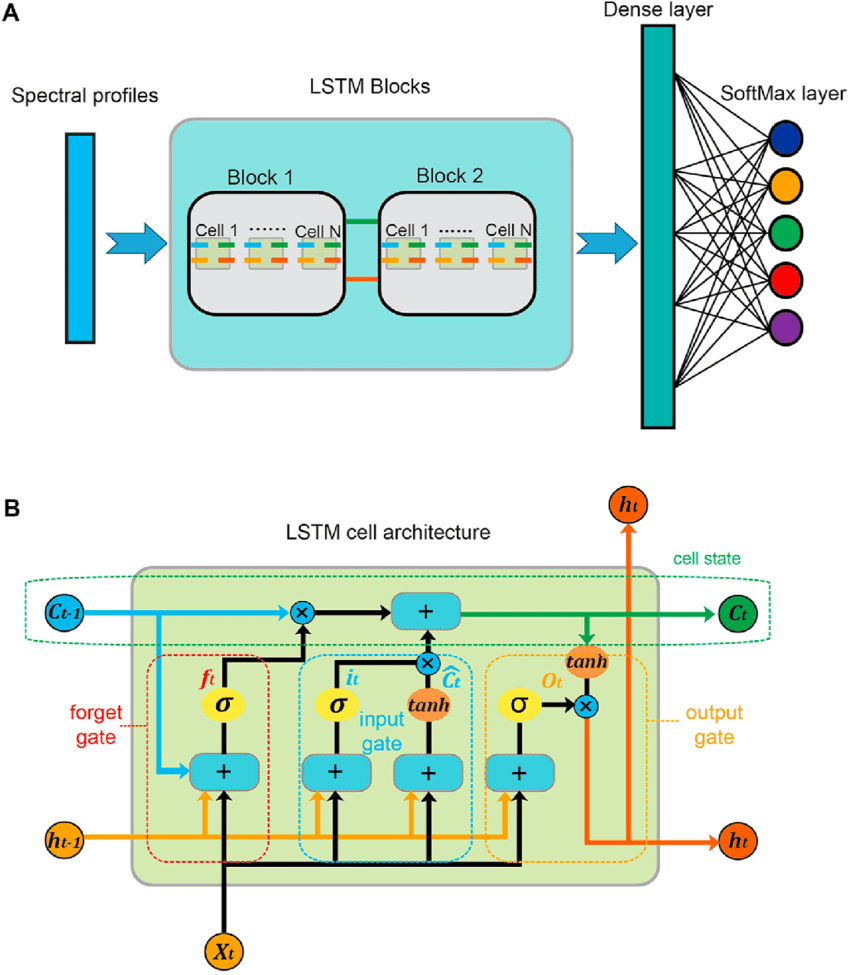


FIG-3.10- Basic architecture of LSTM(Long term short memory)

# CHAPTER 4

# METHODOLOGY

## Dataset

The data used in this work from the Open Energy Data Initiative (OEDI) platform. It is a centralized repository of high-value energy research datasets aggregated from the U.S. Department of Energy’s Programs, Offices, and National Laboratories (Leite and Mantovani, 2016).

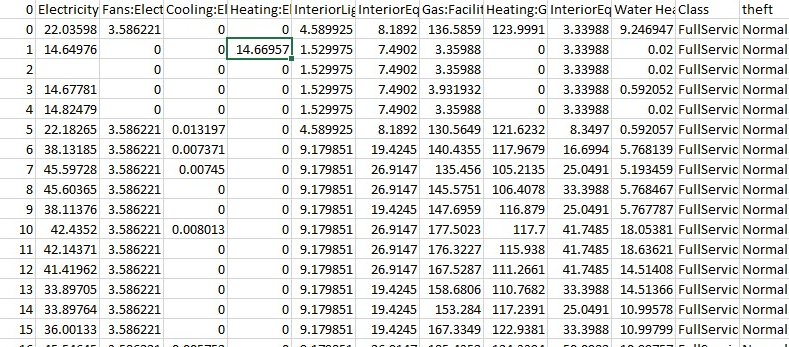


Fig 4.1 Dataset

## Data Statistics/Summary

The dataset has the following features -

* No of consumer types = 17
* No of records: 5,60,655
* No of features: 13



Fig 4.2 Code for Consumers

* Customer list:
  1. FullServiceRestaurant
  2. Hospital
  3. LargeHotel
  4. LargeOffice
  5. MediumOffice
  6. MidriseApartment
  7. OutPatient
  8. PrimarySchool
  9. QuickServiceRestaurant
  10. SecondarySchool
  11. SmallHotel
  12. SmallOffice
  13. Stand-aloneRetail
  14. StripMall
  15. SuperMarket
  16. Warehouse
  17. ‘0’ - Fraud customer
* Features (Columns):
  1. Electricity:Facility [kW](Hourly)
  2. Fans:Electricity [kW](Hourly)
  3. Cooling:Electricity [kW](Hourly)
  4. 'Heating:Electricity [kW](Hourly)
  5. InteriorLights:Electricity [kW](Hourly)
  6. InteriorEquipment:Electricity [kW](Hourly)
  7. Gas:Facility [kW](Hourly)
  8. Heating:Gas [kW](Hourly)Gas [kW](Hourly)
  9. Water Heater:WaterSystems:Gas [kW](Hourly)
  10. Class
  11. theft

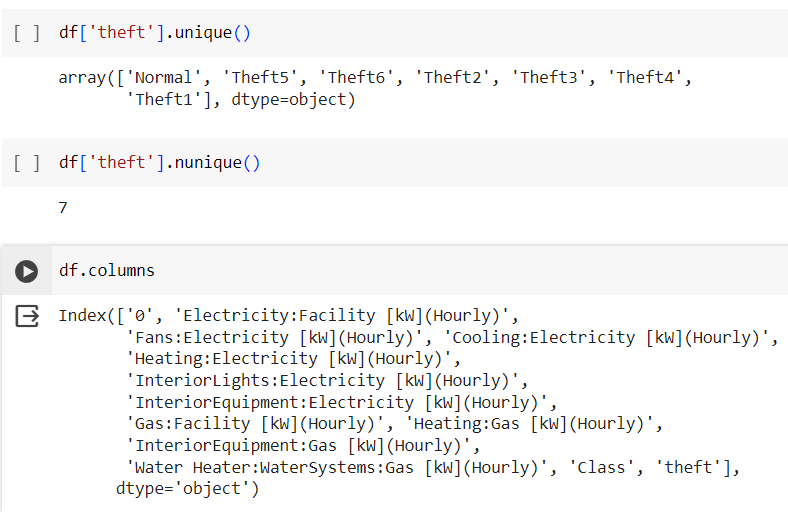


Fig 4.3 Code for features and thefts

## Theft Generator Algorithm

The proposed algorithm to generate the 6 types of electricity thefts can be formally stated as follows: Suppose the daily electricity consumption (X) vector is given as -

X = {x₁; x₂; x₃;...............;x₂₄}

where xᵢ represents the hourly consumption of electricity for i = 1, 2, 3,....24, then the thefts can be generated as follows -

Input: X, Output: TheftN, where N = 1, 2, 3,...,6

Begin:

Theft1(xi) = ɑ\*xᵢ, ɑ = random(.1, 0.8)

Theft2(xi) = βᵢ\*xᵢ,

βᵢ = 0 for tstart < i < tend

tstart = 0.25\*random (0, 23 - toff)

duration = random (toff , 24)

tend = tstart + duration

toff ≥ 4

Theft3(xi) = γi\*xi; γi = random(0.1, 0.8)

Theft4(xi) = γi\*mean(xᵢ), γi = random(0.1, 0.8)

Theft5(xi) = mean(x)

Theft6(xi) = X24-t

## Data Analysis

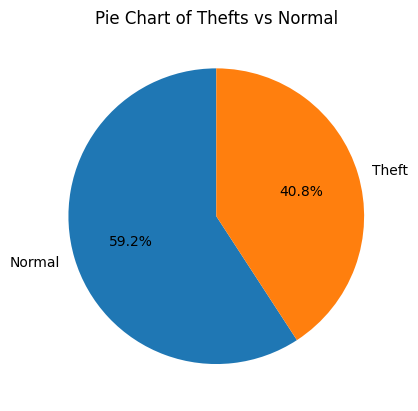
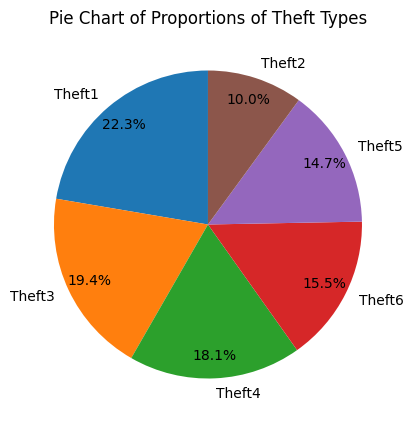


FIG-4.4 Percentage of normal and theft entries in the data set



## FIG-4.5 Percentage of each theft in the data set

# CHAPTER 5 RESULTS

## LSTM (Long Short Term Memory)

## Confusion Matrix:

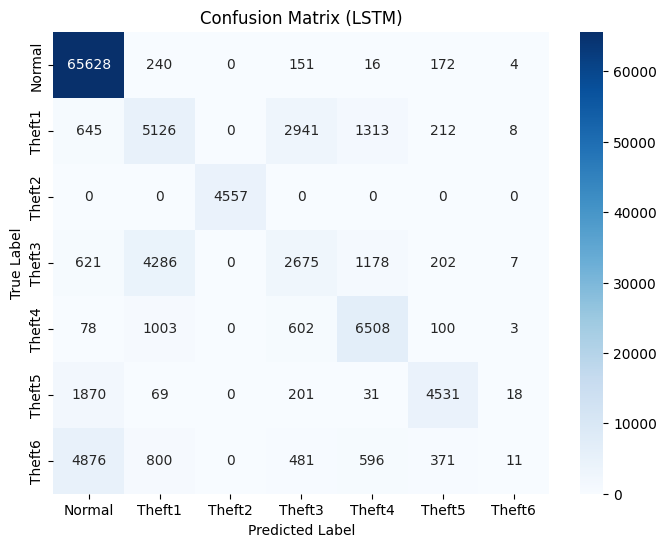
****

FIG-5.1 Confusion matrix for LSTM

* + 1. **Classification Report:**

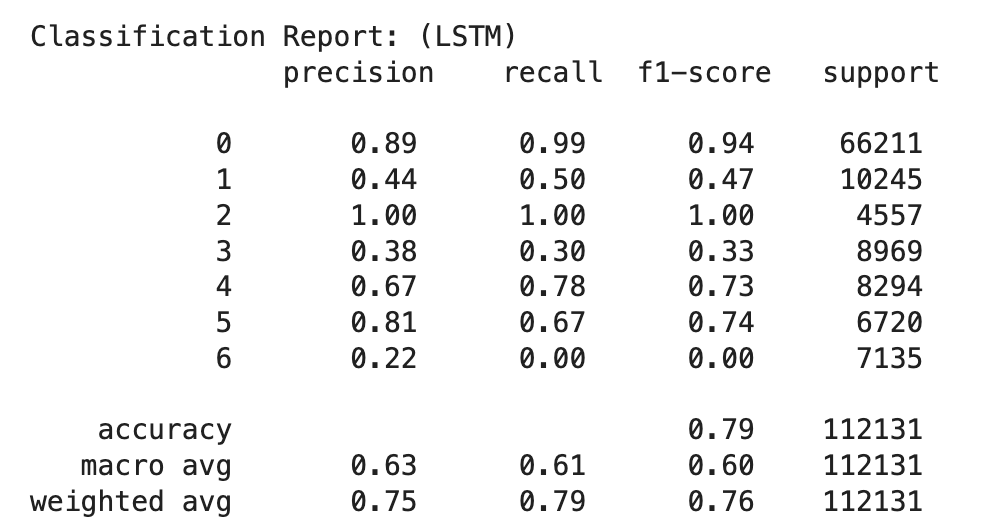


FIG- 5.2 Classification report for LSTM

## 

## Convolutional Neural Networks (CNN)

## Confusion Matrix:

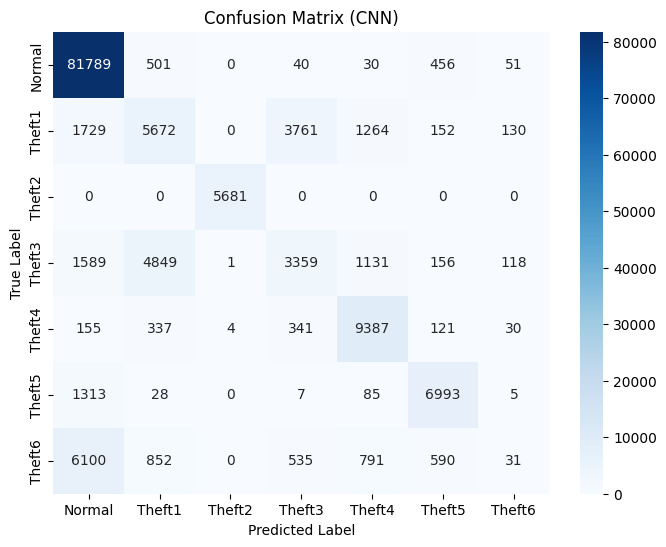
****

Fig 5.3 Confusion matrix for CNN

## Classification Report:

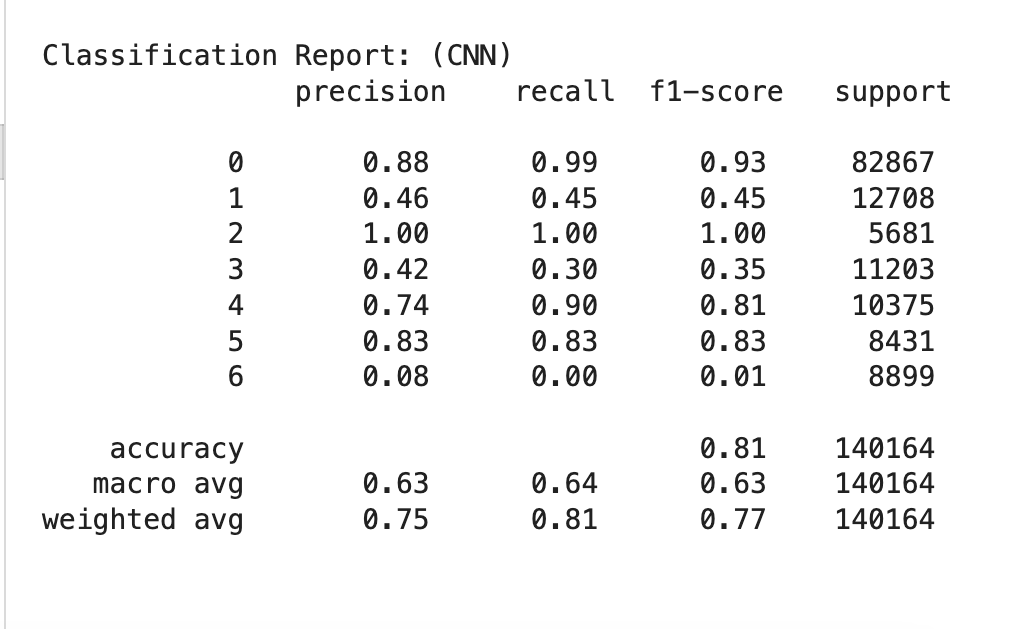


FIG -5.4 Classification report for CNN

## Recurrent Neural Networks (RNN)

## Confusion Matrix:

# 

Fig 5.5 Confusion matrix for RNN

* + 1. **Classification Report:**

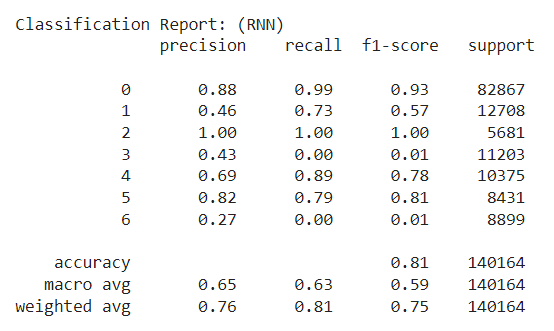


FIG- 5.6 Classification report for RNN

## k-NN (k-Nearest Neighbours)

## Confusion Matrix:

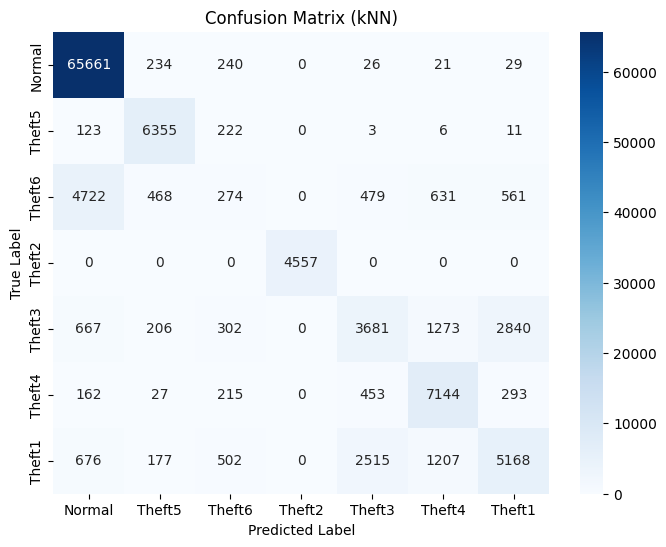


FIG-5.7 Confusion matrix for KNN

* + 1. **Classification Report:**

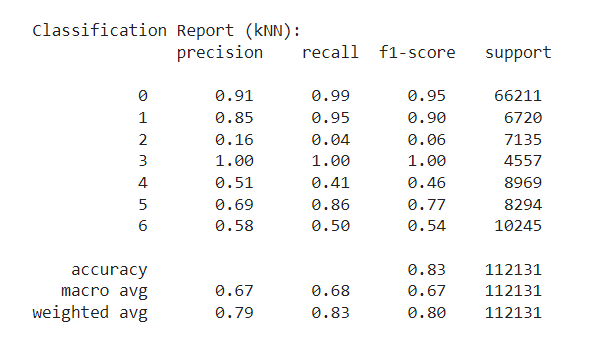


FIG-5.8 Classification report for kNN

## XGBoost (Extreme Gradient Boosting)

## Confusion Matrix:

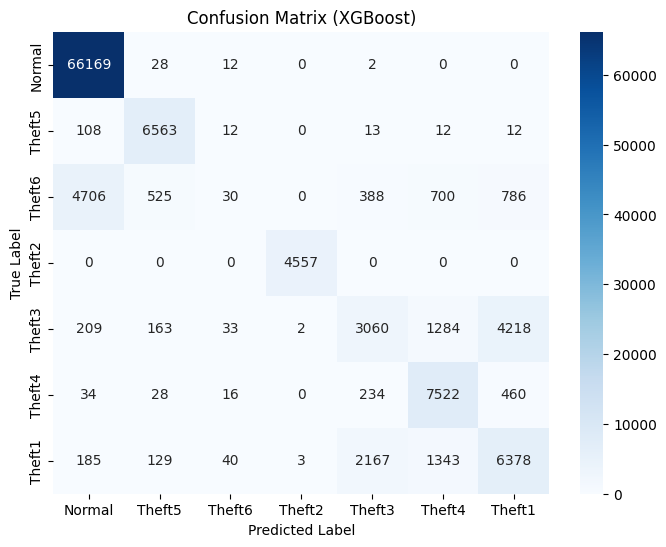


FIG-5.9 Confusion matrix for XGBoost

* + 1. **Classification Report:**

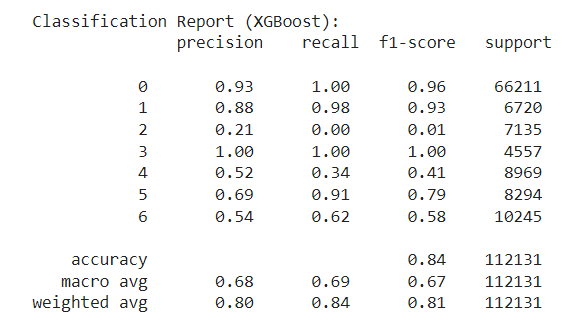


FIG-5.10 Classification report for XGBoost

## Random Forest

## Confusion Matrix:

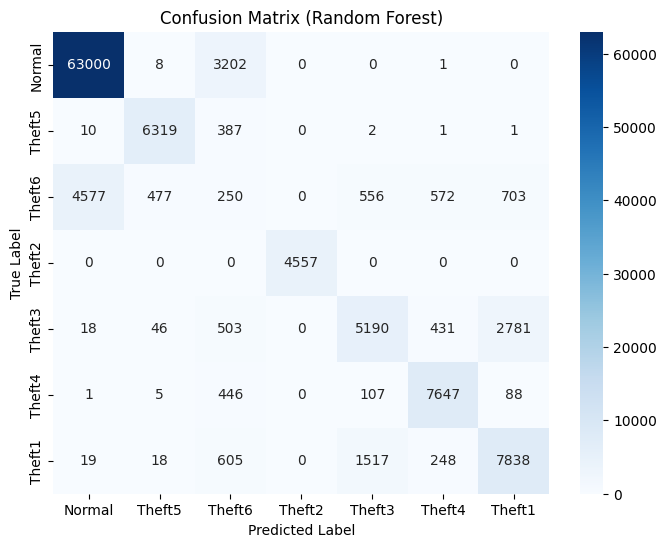


FIG-5.11 Confusion matrix for Random forest

* + 1. **Classification Report:**

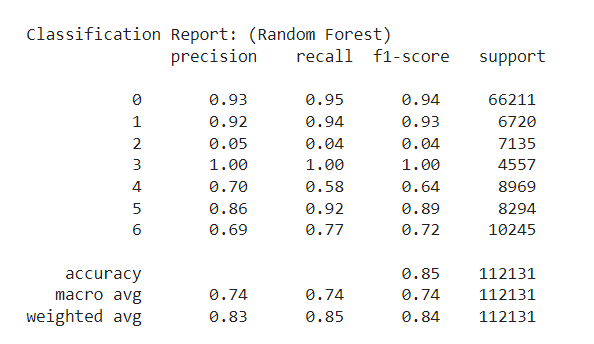


FIG-5.12 Classification report for Random Forest

## Comparison of all models

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST DATA** |  | **RANDOM FOREST** | **XGBOOST** | **KNN** | **RNN** | **CNN** | **LSTM** |
|  | **NORMAL** | 0.93 | 0.93 | 0.91 | 0.88 | 0.88 | 0.89 |
|  | **THEFT 1** | 0.69 | 0.54 | 0.58 | 0.46 | 0.48 | 0.44 |
|  | **THEFT 2** | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| **PRECISION** | **THEFT 3** | 0.70 | 0.52 | 0.51 | 0.43 | 0.46 | 0.38 |
|  | **THEFT 4** | 0.86 | 0.69 | 0.69 | 0.69 | 0.74 | 0.67 |
|  | **THEFT 5** | 0.92 | 0.88 | 0.85 | 0.82 | 0.83 | 0.81 |
|  | **THEFT 6** | 0.05 | 0.21 | 0.16 | 0.27 | 0.10 | 0.22 |
|  |  |  |  |  |  |  |  |
| **ACCURACY** | **THEFT** | 0.85 | 0.84 | 0.83 | 0.81 | 0.81 | 0.79 |

TABLE-5.1 An overview of Precision and Accuracy for all the models

# CHAPTER 6

# CONCLUSION & FUTURE SCOPE

## CONCLUSION

* + - All the models have a high accuracy of greater than or equal to 80%.
    - Random Forest has the highest accuracy (85%). However, it completely fails in identifying Theft6 as the precision score is only 5%.
    - The accuracy of CNN is 81% but it fails to identify Theft6 as the precision (8%) is very low.
    - XGBoost, k-NN and LSTM have accuracies 84%, 83% and 80% respectively, and all the 3 of them have the ability to classify Theft6 relatively better than CNN and Random Forest as their precision values of Theft6 label are 21%, 16% and 22%.
    - RNN has an accuracy of 80% and is the model that best identifies Theft6 compared to all other models that have been tested with as the precision of Theft6 identification is 27%. However, its ability to identify Theft1 and Theft3 is poor compared to other models.
    - Therefore, XGBoost is the best model as it not only has a high accuracy but also identifies Theft6 equally well at the same time compared to other models.

## FUTURE SCOPE

Smart meters are still not used by the majority of consumers. As smart meters penetrate the market further, more data can be mined. To handle this data, more complex neural networks have to be built. Big Data Analytics can be used.

Theft6 which reverses the readings of the smart meters is a deceptive theft as most of the existing models still fail to identify it. The best possible precision attained is 27% by Long Short Term Memory Deep Learning Model.

Theft3 which multiplies each electrical equipment consumption with a different random number between 0.1 and 0.8 is more complex (compared to Theft1 where all the electrical equipment consumption is multiplied by the same random number between 0.1 and 0.8) and is therefore not getting classified properly by most of the models other than Random Forest which heavily misclassified Theft6 (precision 5%).

# REFERENCES

1. Dataset: <https://data.mendeley.com/datasets/c3c7329tjj/1>
2. Y. Wang, Q. Chen, T. Hong, and C. Kang, “Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges,” in IEEE Transactions on Smart Grid, vol. 10, no. 3, pp. 3125-3148, May 2019, doi: 10.1109/TSG.2018.2818167.

<https://ieeexplore.ieee.org/abstract/document/8322199>

1. S. Zidi, A. Mihoub, S. M. Qaisar, M. Krichen, and Q. Abu Al-Haija, “Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment,” in Journal of King Saud University – Computer and Information Sciences, Received 21 February 2022, Revised 12 April 2022, Accepted 11 May 2022, Available online 17 May 2022.

<https://www.researchgate.net/publication/360554845_Theft_Detection_Dataset_for_Benchmarking_and_Machine_Learning_based_Classification_in_a_Smart_Grid_Environment>

1. R. Punmiya and S. Choe, "Energy Theft Detection Using Gradient Boosting Theft Detector With Feature Engineering-Based Preprocessing," in IEEE Transactions on Smart Grid, vol. 10, no. 2, pp. 2326-2329, March 2019, doi: 10.1109/TSG.2019.2892595.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8610248>

1. Z. A. Khan, M. Adil, N. Javaid, M. N. Saqib, M. Shafiq, and J. G. Choi, “Electricity Theft Detection Using Supervised Learning Techniques on Smart Meter Data,” in Sustainability, vol. 12, no. 19, pp. 8023, September 28, 2020, doi: 10.3390/su12198023.

<https://www.mdpi.com/2071-1050/12/19/8023>

1. J. Y. Kim, Y. M. Hwang, Y. G. Sun, I. Sim, D. I. Kim, and X. Wang, “Detection for Non-Technical Loss by Smart Energy Theft With Intermediate Monitor Meter in Smart Grid,” in IEEE Access, vol. 7, pp. 129043-129053, September 11, 2019, doi: 10.1109/ACCESS.2019.2940443.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8832141>

# APPENDIX

1. **APPENDIX - A**

List of python libraries used -

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear\_model import LogisticRegression

from xgboost import XGBClassifier

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

from sklearn.multiclass import OneVsRestClassifier

from sklearn.preprocessing import label\_binarize

from google.colab import drive

from keras.layers import SimpleRNN, Dense

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense

1. **APPENDIX - B**

LSTM Code -

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_encoded, test\_size=0.2, random\_state=42)

X\_train = np.reshape(X\_train, (X\_train.shape[0], X\_train.shape[1], 1))

X\_test = np.reshape(X\_test, (X\_test.shape[0], X\_test.shape[1], 1))

model = Sequential()

model.add(LSTM(32, activation='tanh', input\_shape=(X\_train.shape[1], 1)))

model.add(Dense(64, activation='tanh'))

model.add(Dense(7, activation='softmax'))

model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

history = model.fit(X\_train, y\_train, epochs=20, batch\_size=128, validation\_split=0.2)

y\_pred = np.argmax(model.predict(X\_test), axis=-1)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label\_encoder.classes\_, yticklabels=label\_encoder.classes\_)

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.title('Confusion Matrix (LSTM)')

plt.show()

print("\nClassification Report: (LSTM)\n", classification\_report(y\_test, y\_pred))

1. **APPENDIX - C**

CNN code -

X = df.iloc[:, 1:11]

y = df['theft']

label\_encoder = LabelEncoder()

y\_encoded = label\_encoder.fit\_transform(y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_encoded, test\_size=0.2, random\_state=42)

X\_train = np.expand\_dims(X\_train, axis=2)

X\_test = np.expand\_dims(X\_test, axis=2)

model = Sequential()

model.add(Conv1D(32, 3, activation='relu', input\_shape=(10, 1)))

model.add(MaxPooling1D(2))

model.add(Conv1D(64, 3, activation='relu'))

model.add(MaxPooling1D(2))

model.add(Flatten())

model.add(Dense(64, activation='relu'))

model.add(Dense(7, activation='softmax'))

model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

history = model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2)

y\_pred = np.argmax(model.predict(X\_test), axis=-1)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label\_encoder.classes\_, yticklabels=label\_encoder.classes\_)

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.title('Confusion Matrix (CNN)')

plt.show()

print("\nClassification Report: (CNN)\n", classification\_report(y\_test, y\_pred))

1. **APPENDIX - D**

XGB Code -

df.columns = ["".join (c if c.isalnum() else "\_" for c in str(x)) for x in df.columns]

X = df.iloc[:, 1:11]

y = df['theft']

label\_mapping = {'Normal': 0, 'Theft5': 1, 'Theft6': 2, 'Theft2': 3, 'Theft3': 4, 'Theft4': 5, 'Theft1': 6}

y = y.map(label\_mapping)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

xgb\_model = XGBClassifier(random\_state=42)

xgb\_model.fit(X\_train, y\_train)

y\_pred\_xgb = xgb\_model.predict(X\_test)

conf\_matrix\_xgb = confusion\_matrix(y\_test, y\_pred\_xgb)

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix\_xgb, annot=True, fmt='d', cmap='Blues', xticklabels=label\_mapping.keys(), yticklabels=label\_mapping.keys())

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.title('Confusion Matrix (XGBoost)')

plt.show()

print("Classification Report (XGBoost):\n", classification\_report(y\_test, y\_pred\_xgb))

1. **APPENDIX - E**

kNN code -

X = df.iloc[:, 1:11]

y = df['theft']

label\_encoder = LabelEncoder()

y\_encoded = label\_encoder.fit\_transform(y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_encoded, test\_size=0.2, random\_state=42)

knn\_model = KNeighborsClassifier(n\_neighbors=5) # You can adjust the number of neighbors (k) as needed

knn\_model.fit(X\_train, y\_train)

y\_pred\_knn = knn\_model.predict(X\_test)

conf\_matrix\_knn = confusion\_matrix(y\_test, y\_pred\_knn)

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix\_knn, annot=True, fmt='d', cmap='Blues', xticklabels=label\_mapping.keys(), yticklabels=label\_mapping.keys())

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.title('Confusion Matrix (kNN)')

plt.show()

print("\nClassification Report (kNN):\n", classification\_report(y\_test, y\_pred\_knn))

1. **APPENDIX - F**

Random Forest -

X = df.iloc[:, 1:11] # Features

y = df['theft'] # Target variable

label\_mapping = {'Normal': 0, 'Theft5': 1, 'Theft6': 2, 'Theft2': 3, 'Theft3': 4, 'Theft4': 5, 'Theft1': 6}

y = y.map(label\_mapping)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

rf\_model = RandomForestClassifier(random\_state=42)

rf\_model.fit(X\_train, y\_train)

y\_pred = rf\_model.predict(X\_test)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label\_mapping.keys(), yticklabels=label\_mapping.keys())

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.title('Confusion Matrix (Random Forest)')

plt.show()

print("\nClassification Report: (Random Forest)\n", classification\_report(y\_test, y\_pred))

1. **APPENDIX - G**

RNN code -

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_encoded, test\_size=0.25, random\_state=42)

model = Sequential()

model.add(SimpleRNN(64, input\_shape=(X\_train.shape[1], 1), activation='relu'))

model.add(Dense(32, activation='relu'))

model.add(Dense(7, activation='softmax'))

model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

history = model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2)

y\_pred\_probs = model.predict(X\_test)

y\_pred = np.argmax(y\_pred\_probs, axis=1)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=label\_encoder.classes\_, yticklabels=label\_encoder.classes\_)

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.title('Confusion Matrix (RNN)')

plt.show()

print("\nClassification Report: (RNN)\n", classification\_report(y\_test, y\_pred))

ENERGY THEFT DETECTION USING MACHINE LEARNING AND DEEP LEARNING METHODS

ORIGINALITY REPORT

27

%

SIMILARITY INDEX

21%

INTERNET SOURCES

17%

PUBLICATIONS

13%

STUDENT PAPERS

PRIMARY SOURCES

[www.geeksforgeeks.org](http://www.geeksforgeeks.org/)

Internet Source %

1

3

2

[www.mdpi.com](http://www.mdpi.com/)

Internet Source %

2

2%

3 Salah Zidi, Alaeddine Mihoub, Saeed Mian Qaisar, Moez Krichen, Qasem Abu Al-Haija. "Theft detection dataset for benchmarking and machine learning based classiﬁcation in a smart grid environment", Journal of King Saud University - Computer and Information Sciences, 2022

Publication

2

fastercapital.com

Internet Source %

4

1

[www.ieee-jas.net](http://www.ieee-jas.net/)

Internet Source %

5

1

Submitted to Nottingham Trent University

Student Paper %

6

Submitted to University of Sydney

7

8

9

10

11

12

13

14

15

16

Student Paper

eudl.eu

Internet Source

[www.coursehero.com](http://www.coursehero.com/)

Internet Source

Yi Wang, Qixin Chen, Tao Hong, Chongqing Kang. "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges", IEEE Transactions on Smart Grid, 2019

Publication

ijeecs.iaescore.com

Internet Source

jaes.qu.edu.sa

Internet Source

[www.researchgate.net](http://www.researchgate.net/)

Internet Source

Submitted to Liverpool John Moores University

Student Paper

Submitted to Fiji National University

Student Paper

[www.jcdronline.org](http://www.jcdronline.org/)

Internet Source

1%

1%

1%

1%

1%

1%

1%

1%

<1%

<1%

17

18

19

20

21

22

23

24

Submitted to Cornell University

Student Paper

Submitted to University of Salford

Student Paper

"Recent Advancements in Artiﬁcial Intelligence", Springer Science and Business Media LLC, 2024

Publication

Submitted to University of North Texas

Student Paper

Submitted to Polytechnic of Namibia

Student Paper

Ton Duc Thang University

Publication

etheses.whiterose.ac.uk

Internet Source

Karen Joselyne Avilez-Cuahquentzi, Antonio Flores-Tlacuahuac, Diana Ramírez-Gamboa, Roberto Parra-Saldivar. "Data-driven recurrent neural networks modeling of cyanobacteria growth in bubble columns reactors under sparging with CO2-enriched Air", Chemical Engineering and Processing - Process Intensiﬁcation, 2024

Publication

gcris.ieu.edu.tr

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

25

26

27

28

29

30

31

32

33

34

Internet Source

[www.frontiersin.org](http://www.frontiersin.org/)

Internet Source

[www.holisticseo.digital](http://www.holisticseo.digital/)

Internet Source

Nitesh A. Funde, Meera M. Dhabu, Aarthi Paramasivam, Parag S. Deshpande. "Motif- based Association Rule Mining and Clustering Technique for Determining Energy Usage Patterns for Smart Meter Data", Sustainable Cities and Society, 2019

Publication

medium.com

Internet Source

thesai.org

Internet Source

vdoc.pub

Internet Source

[www.e3s-conferences.org](http://www.e3s-conferences.org/)

Internet Source

[www.legalserviceindia.com](http://www.legalserviceindia.com/)

Internet Source

Submitted to Aston University

Student Paper

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

35

36

37

38

39

40

41

42

43

Submitted to The University of Texas at Arlington

Student Paper

Submitted to Arts, Sciences & Technology University In Lebanon

Student Paper

Submitted to CSU Northridge

Student Paper

Submitted to University of Wales, Lampeter

Student Paper

ebin.pub

Internet Source

[www.trendingus.com](http://www.trendingus.com/)

Internet Source

pdfcoﬀee.com

Internet Source

doi.org

Internet Source

Muhammad Hamza Zafar, Syed Muhammad Salman Bukhari, Mohamad Abou Houran, Syed Kumayl Raza Moosavi et al. "Step towards secure and reliable smart grids in Industry 5.0: A federated learning assisted hybrid deep learning model for electricity theft detection using smart meters", Energy Reports, 2023

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

44

45

46

47

48

49

50

51

Publication

Yi Wang, Qixin Chen, Chongqing Kang. "Smart Meter Data Analytics", Springer Science and Business Media LLC, 2020

Publication

link.springer.com

Internet Source

[www.groundai.com](http://www.groundai.com/)

Internet Source

[www.hindawi.com](http://www.hindawi.com/)

Internet Source

www.ir.juit.ac.in:8080

Internet Source

Ejaz Ul Haq, Can Pei, Ruihong Zhang, Huang Jianjun, Fiaz Ahmad. "Electricity-theft detection for smart grid security using smart meter data: A deep-CNN based approach", Energy Reports, 2023

Publication

Jay Lee. "Industrial AI", Springer Science and Business Media LLC, 2020

Publication

Jin Zhu, Yingjun Shen, Zhe Song, Dequn Zhou, Zijun Zhang, Andrew Kusiak. "Data-driven building load proﬁling and energy

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

52

53

54

55

56

57

management", Sustainable Cities and Society, 2019

Publication

Kharat Jayashree Prashant, Kharat Prashant Krishnrao. "Frame Shuﬄing Forgery Detection Method for MPEG-Coded Video", Journal of The Institution of Engineers (India): Series B, 2024

Publication

Zhongzong Yan, He Wen. "Performance Analysis of Electricity Theft Detection for the Smart Grid: An Overview", IEEE Transactions on Instrumentation and Measurement, 2021

Publication

dokumen.pub

Internet Source

mdpi-res.com

Internet Source

research.iimidr.ac.in

Internet Source

Amol Dhumane, Mubin Tamboli, Srinivas Ambala, Pravin Game, Vishal Meshram, Rahul Patil. "Machine Learning Approach for Predicting the Placement Status of Students", 2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA), 2023

Publication

<1%

<1%

<1%

<1%

<1%

<1%

58

59

Farah Mohammad, Kashif Saleem, Jalal Al- Muhtadi. "Ensemble-Learning-Based Decision Support System for Energy-Theft Detection in Smart-Grid Environment", Energies, 2023

Publication

Rajiv Punmiya, Sangho Choe. "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing", IEEE Transactions on Smart Grid, 2019

Publication

Exclude quotes Oﬀ Exclude bibliography Oﬀ

<1%

<1%

Exclude matches Oﬀ